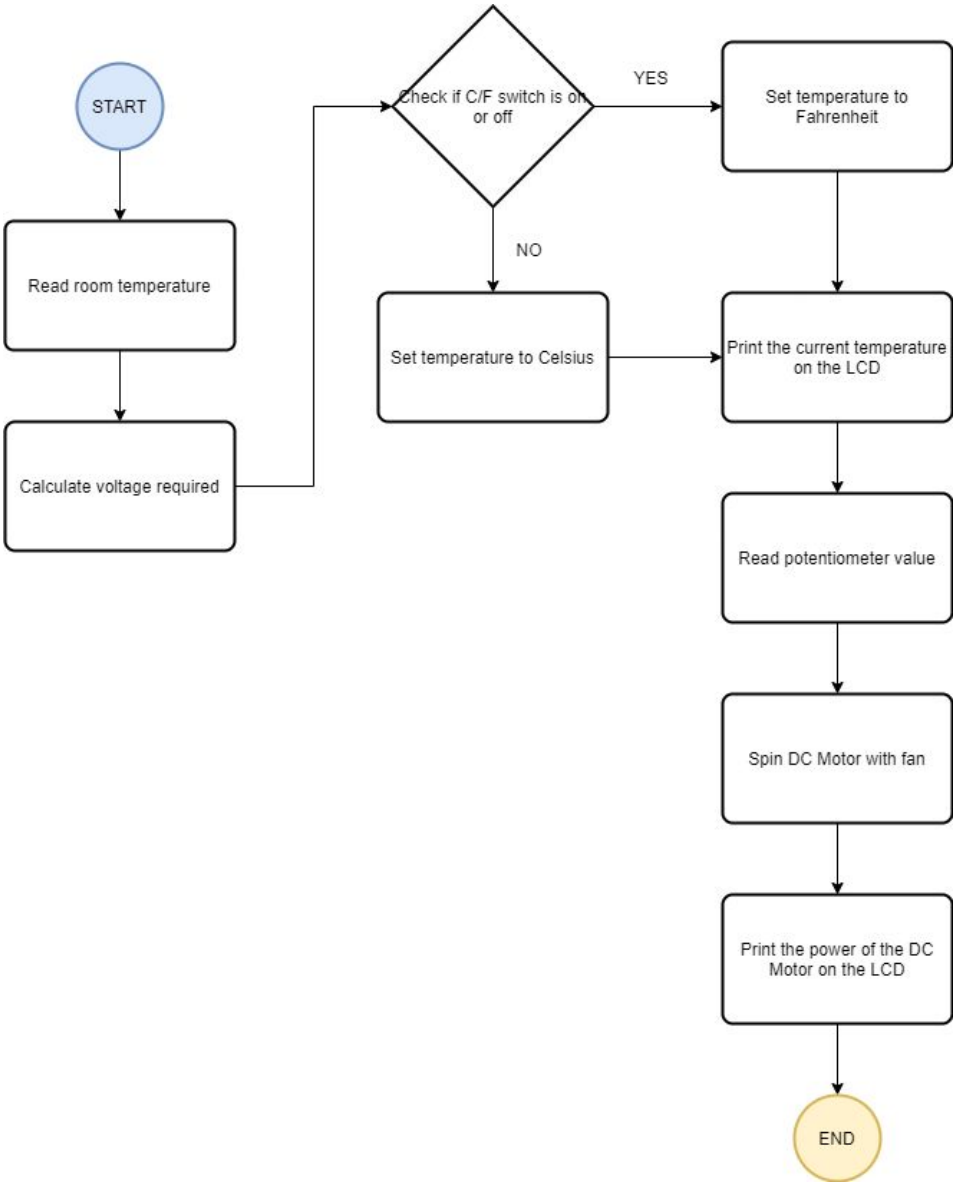
Документация

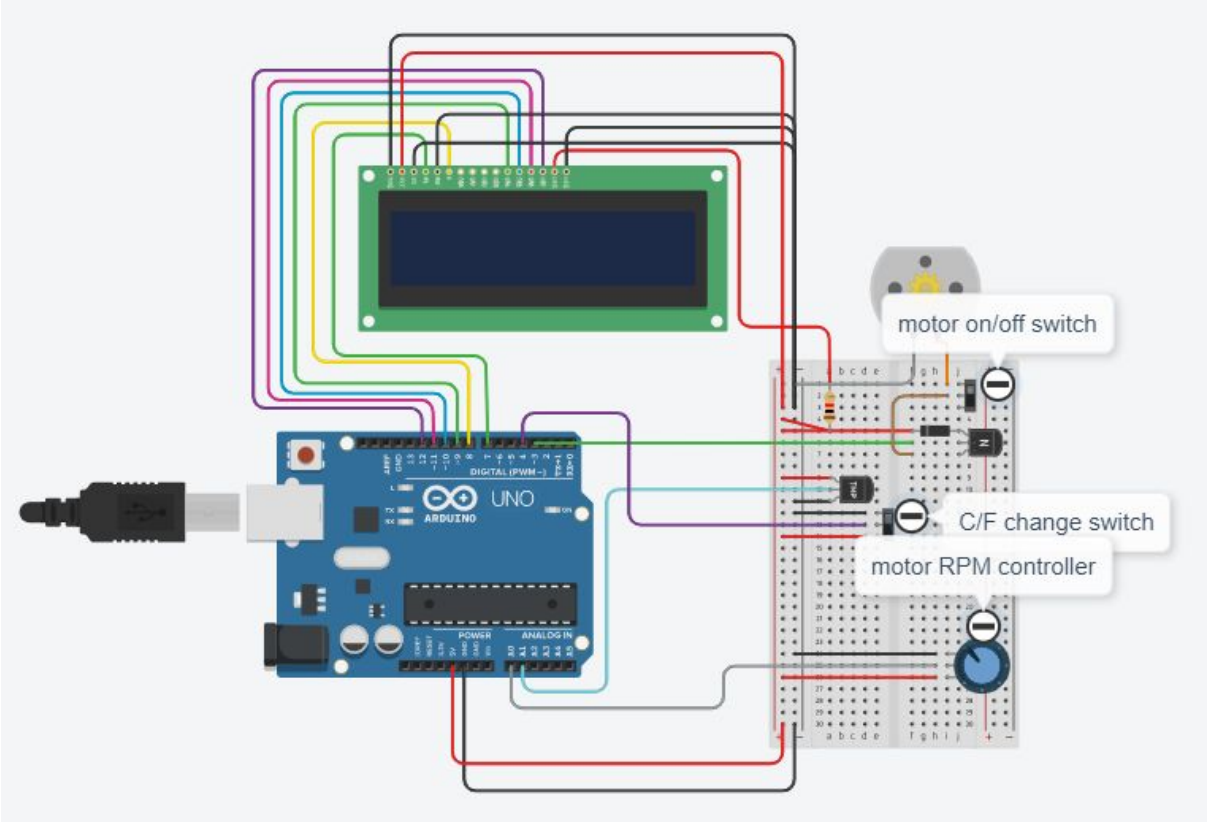
Участници в проекта:
Ивайло Христов и Божидар Атанасов

Описание на проекта:
Вентилатор контролиран от потенциометър с включен дисплей показващ мощност в проценти и текуща температура в градуси целзий или фаренхайт.

Блокова схема:



Електрическа схема:

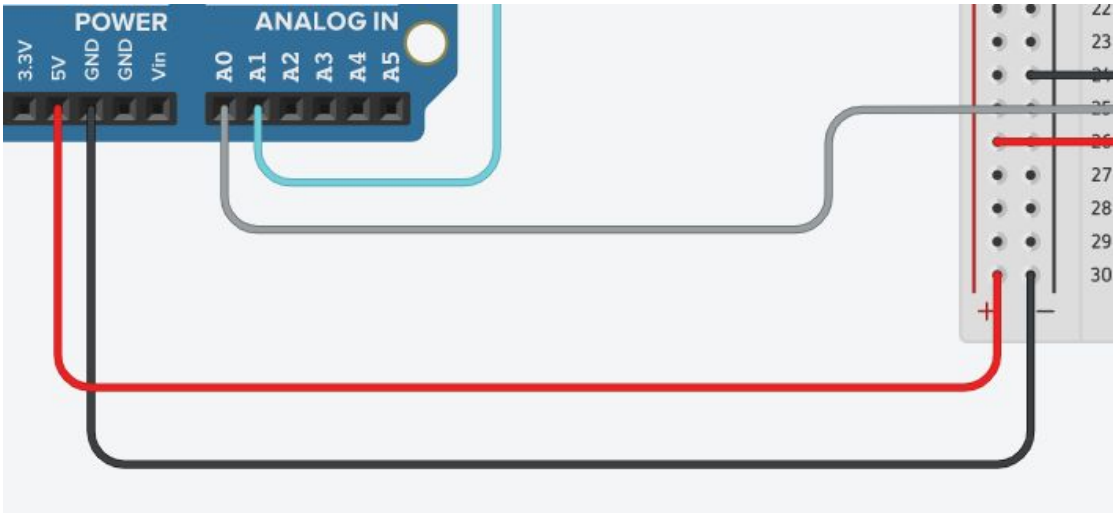


Списък съставни части:

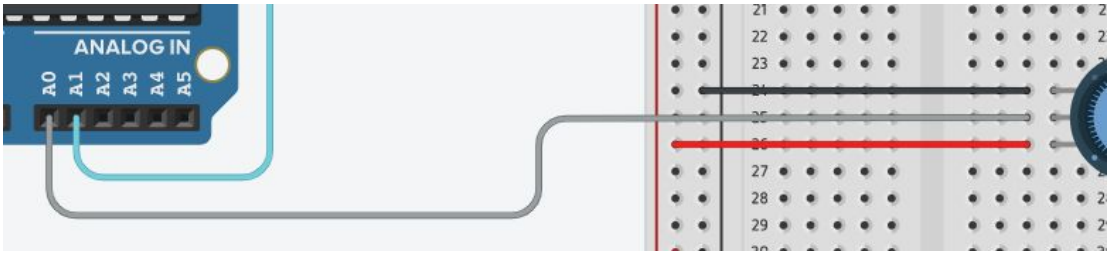
LCD 16x2
DC Motor
Arduino Uno R3
Breadboard
Potentiometer
Temperature Sensor
2x Slideswitch
NPN Transistor
Diod
Resistor

Инструкции:

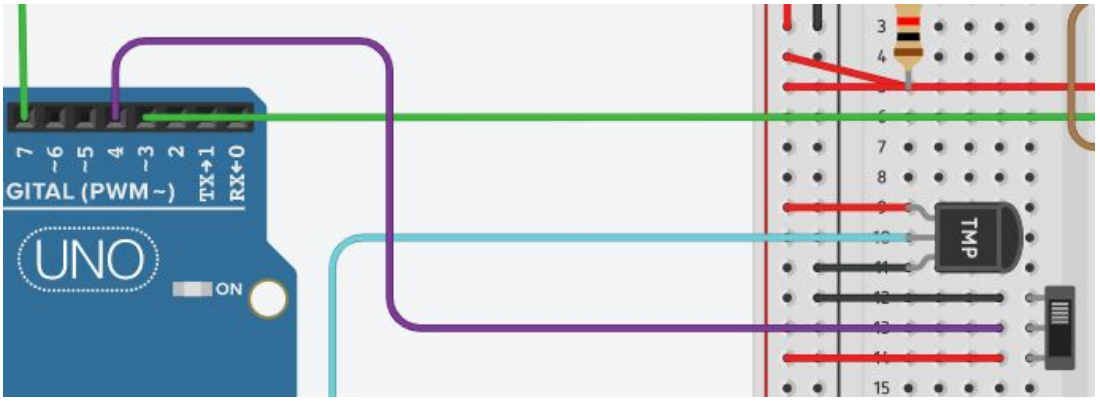
1. Връзваме GND и напрежението, breadboard и arduino



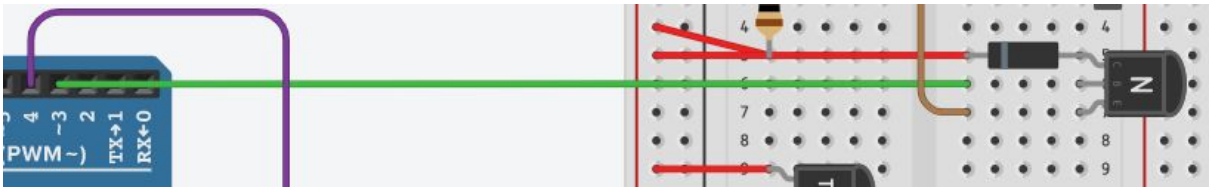
2. Поставяме потенциометъра на breadboard-а и го свързваме. Terminal 1 на -, Wiper на A0, Terminal 2 на +.



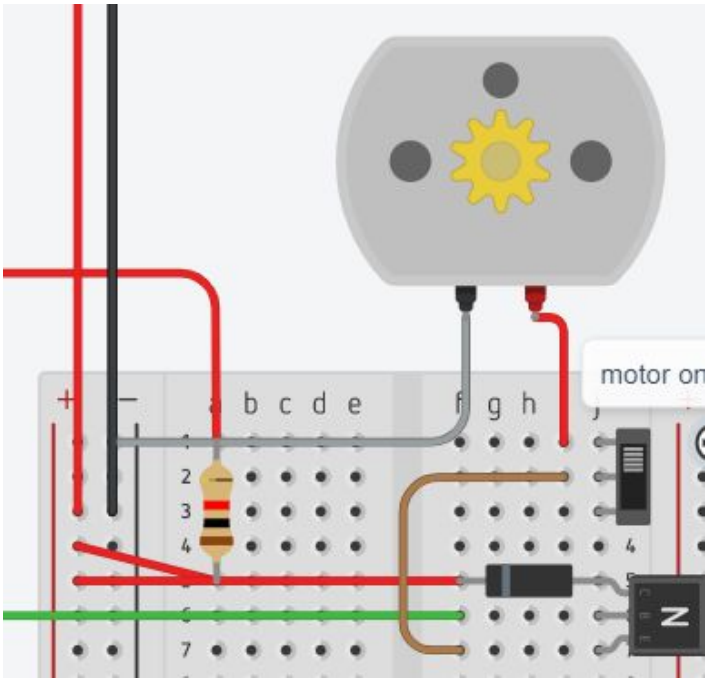
3. Поставяме температурния сензор и switch-а и ги свързваме. Power на +, GND на -, Vout на A1 (сензор). Terminal 1 на -, Common на D4, Terminal 2 на + (switch).



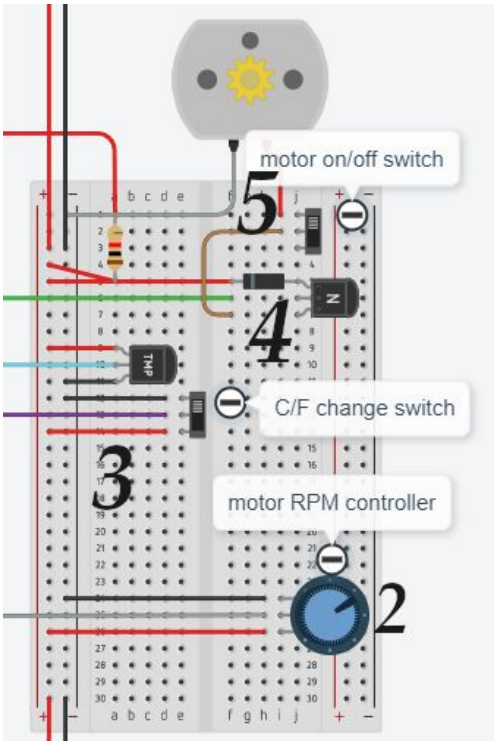
4. Поставяме NPN транзистора заедно с диода и ги връзваме. Collector на Anode, Base на D3(NPN). Cathode на + (диод).



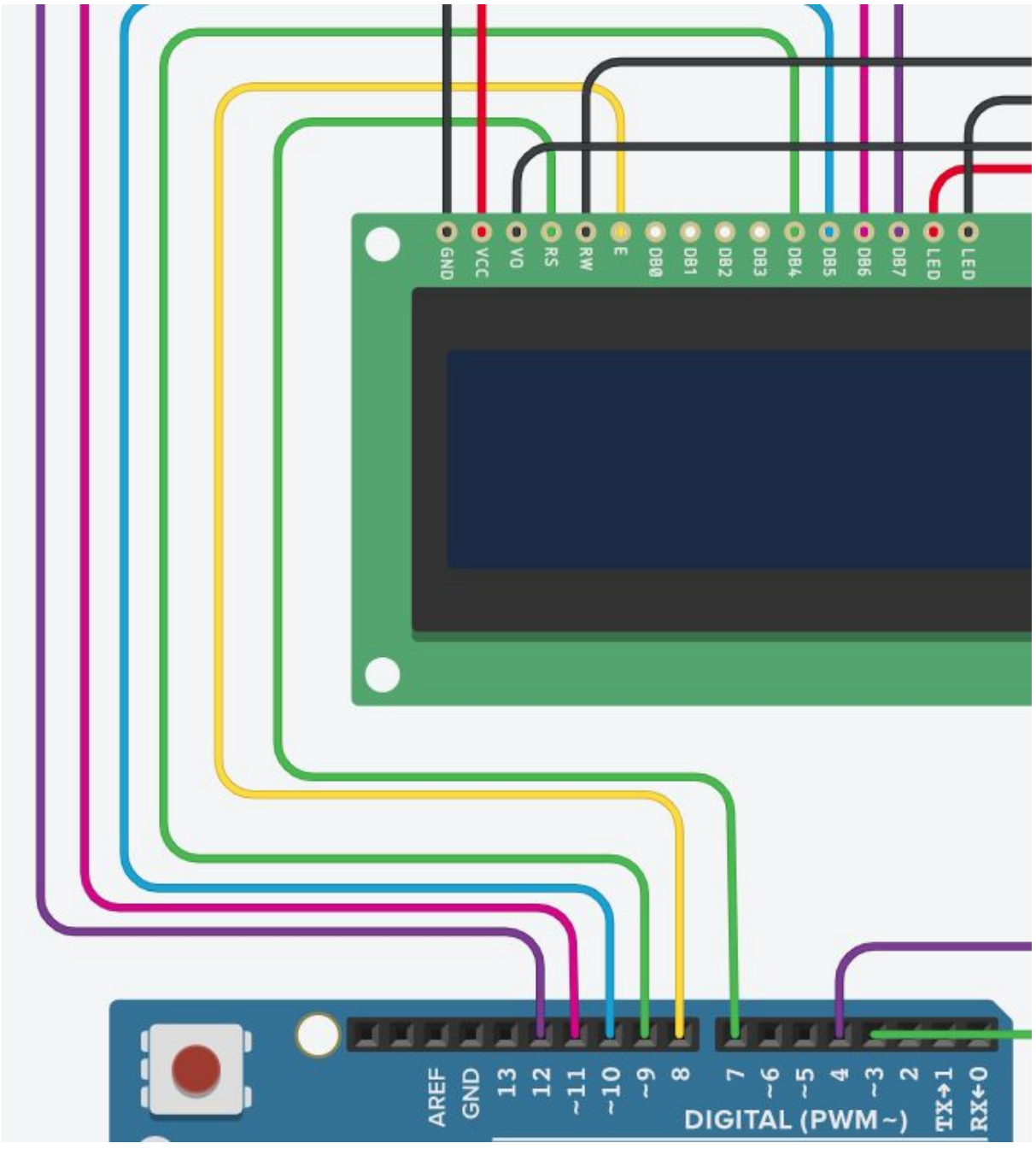
5. Поставяме DC мотора и switch-а и ги връзваме. Terminal 1 на -. Terminal 2 (мотор) на Terminal 1 (switch). Common (switch) на Emitter (NPN).



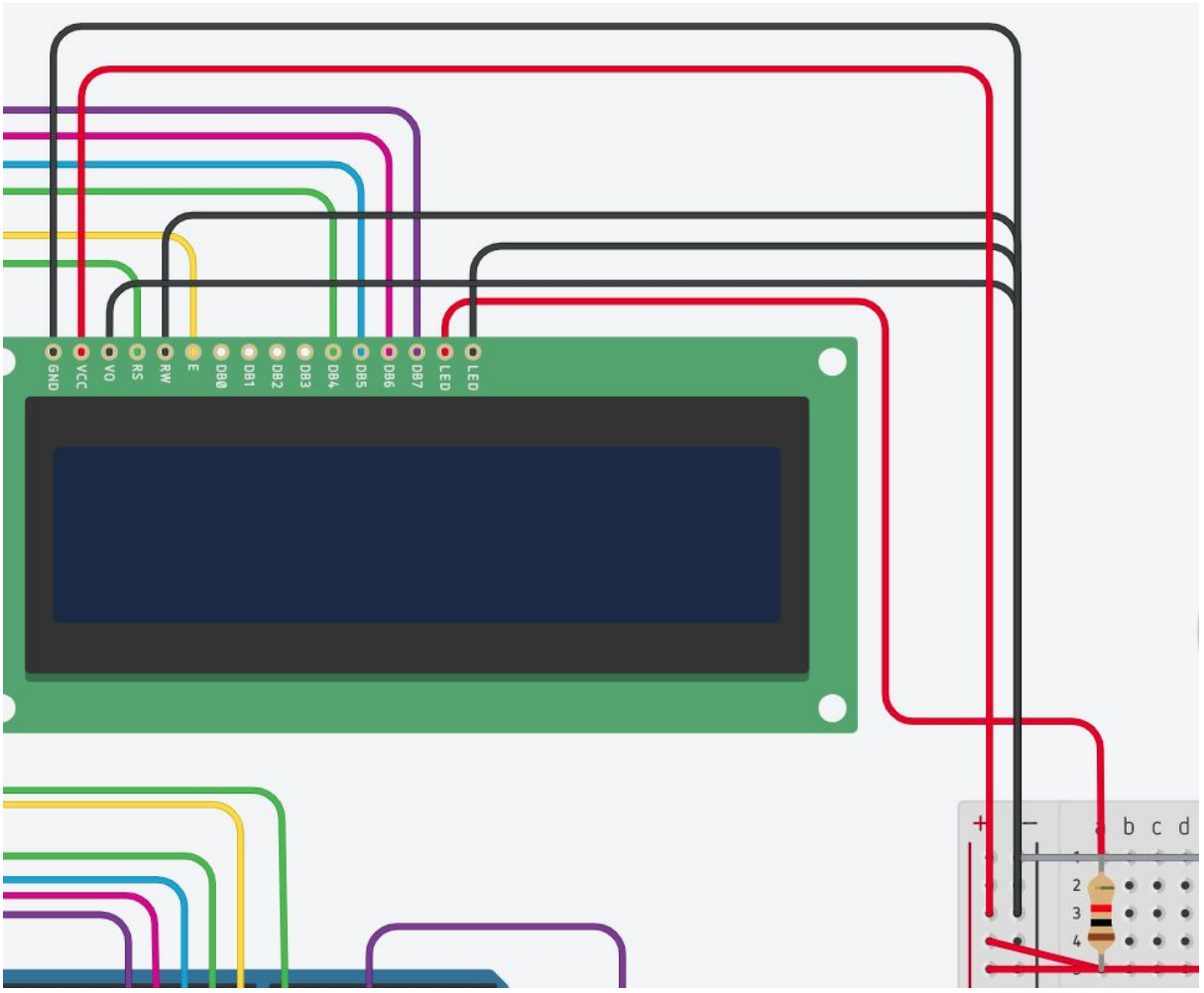
Как трябва да изглежда всичко накрая:



6. LCD екрана претежава пинове които се свързват към захранване и ground или към arduino за разменн на данни. Свързваме RS - arduino D7, E - arduino D8, DB4 - arduino D9, DB5 - arduino D10, DB6 - arduino D11, DB5 - arduino D10 и DB7 - arduino D12



7. Свързваме GDN - breadboard GND, VCC - breadboard 5V, VD - breadboard GND, RW - breadboard GND, LED Anode - през 1kΩ резистор към breadboard 5V, LED Cathode - breadboard GND



Сорс код:

```
//the transistor which controls the motor will be attached to
digital pin 9
const int motorControl = 3;
//The pin for the potenciometer
const int pinPot = A0;
//current temperature taken from sensor
const int tempPin = A1;
//the pin for button changing between C and F
const int CFPin = 4;
//bool used to keep track of current metric used
bool IsC = true;

#include <LiquidCrystal.h>
LiquidCrystal lcd(7, 8, 9, 10, 11, 12);
// the setup routine runs once when you press reset:
```

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

  // make the potenciometurs's pin an input:
  pinMode(pinPot, INPUT);

  // make the transistor's pin an output:
  pinMode(motorControl, OUTPUT);

  // make the tempetur sensor's pin an output:
  pinMode(tempPin, INPUT);

  //enables the serial monitor
  Serial.begin(9600);

  pinMode(CFPin, INPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  int reading = analogRead(tempPin);
  //Serial.println(reading);
  float voltage = reading * 5.0;
  voltage /= 1024.0;
  lcd.setCursor(0, 0);
  String messegeTemp = "Temp: ";
  if(digitalRead(CFPin) != HIGH)
  {
    messegeTemp += calcC(voltage);
    messegeTemp += "C    ";
  }
  else
  {
     messegeTemp += (calcC(voltage) * 9.0 / 5.0) + 32.0;
     messegeTemp += "F    ";
  }
  lcd.print(messegeTemp);
  //gets and displays temp

  //gets pinpot val and turns into readable num
  int analogVal = analogRead(pinPot);
```

```
  //gets potenciometer value from 1023 to 0
  analogWrite(motorControl, map(analogVal, 0, 1023, 0, 255));
  //turns it from 1023-0 to 255-0
  //gives it to the npn that runs the motor
  String messegeRPM = "Fan: ";
  messegeRPM += map(analogVal, 0, 1023, 0, 100);
  //map to get value for lcd
  messegeRPM += "%       ";
  //can't clear line or chars this fixes ghosting
  lcd.setCursor(0, 1);
  //sets cursor to second line
  lcd.print(messegeRPM);


  delay(1);        // delay for stability
}

float calcC(float voltage)
{
  return (voltage - 0.5) * 100;
}
```