

Курсова работа

Дисциплина: Процесна информация и обработка

Студент: Юмер Местан

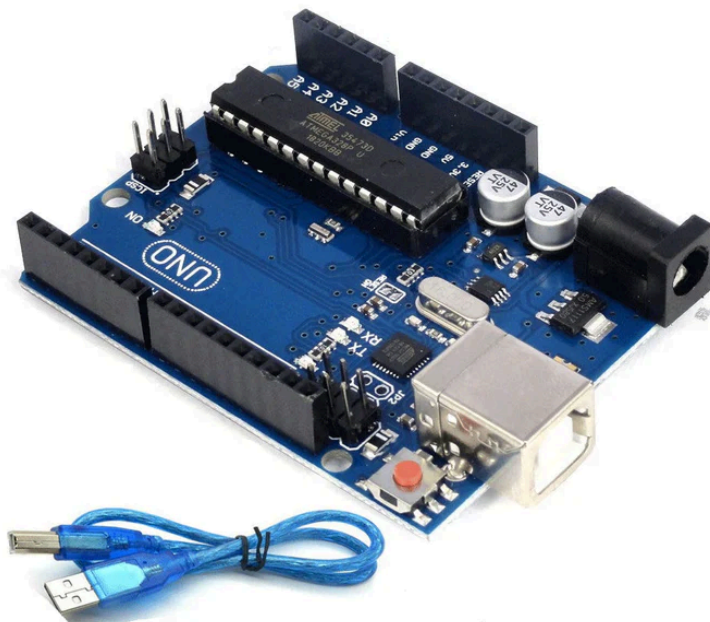
Група:

Факултетен номер: 501222071

1. Цел и постановка на задачата:

Целта на настоящата курсова работа е да се симулира работата на микроконтролер с отворен код Arduino Uno и устройство DHT-22. За целта се използва виртуална среда WOKWI, в която се демонстрира работата на датчик за температура и влажност, извеждане на показанията на LCD дисплей и звукова сигнализация при достигане на критични минимална и максимална температура.

2. Описание на използвания микроконтролер (Arduino Uno):



Фиг 1. - Arduino UNO R3 - платка със сменяем процесор

<https://kasabov.eu/cdn/shop/files/arduino-uno-r3-dip.jpg?v=1708362377&width=713>

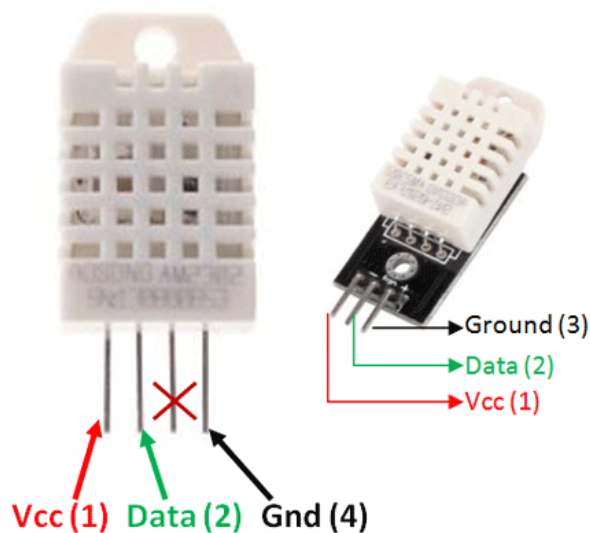
Arduino Uno е най-популярният микроконтролер в семейството на Arduino. В основата му е залегнал ATmega328P чип, който разполага с 32K байта флаш памет (където се съхраняват потребителски данни или код), 2k байта SRAM и 1k байт EEPROM. Разполага с

13 GPIO (General Purpose Input Output) пина, разположени в горната част на платката, 3 Ground пина, 6 аналогови пина (в долната част на платката).

- **Процесор:** ATmega328P с тактова честота от 16 MHz
- **Цифрови входно-изходни пинове:** 14, от които 6 поддържат PWM (Pulse Width Modulation) - разположени в горният край на платката
- **Аналогови входове:** 6 канала за аналогово четене - **A0 - A5** в долната част на платката
- **Памет:** 32 KB flash памет за съхранение на програма или данни дефинирани от потребителя, 2 KB SRAM и 1 KB EEPROM
- **Интерфейси:** Поддържа SPI, I2C (TWI) и UART (Serial) за комуникация с външни устройства
- **Свързаност:** USB-B порт за програмиране и сериен монитор, както и възможност за външно захранване
- **Разработка:** Програмирането се извършва чрез Arduino IDE, което опростява разработката на вградени системи и прототипи. В нашият случай се използва интегрираната среда за разработка на WOKWI

Тази платка е идеална за бързо прототипиране и образователни проекти, благодарение на своята простота, гъвкавост и богат набор от вградени модули и функции, които са използвани и в настоящата курсова работа.

3. Описание на устройство DHT22 - работа и цели на употреба в бита и индустрията



Фиг. 2 - Сензор за влажност и температура DHT22 на компанията AOSONG

https://components101.com/sites/default/files/component_pin/DHT22-Sensor-Pinout.png

DHT22 е базов, евтин дигитален сензор за отчитане на температура и влажност. Другото име с което може да бъде срещнат е AM2302.

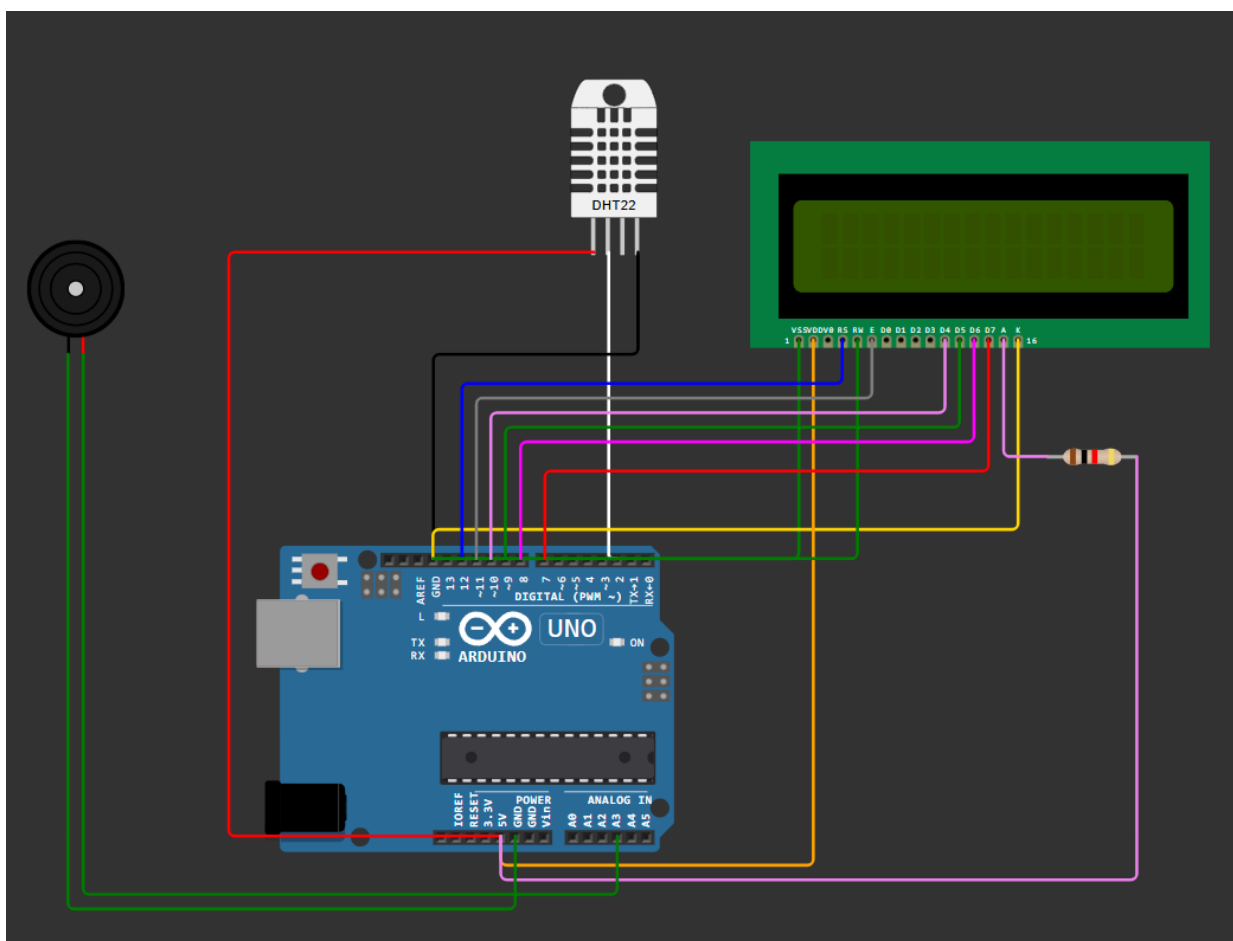
Характеризира се с малки размери, ниска консумация и сравнително дълго разстояние на предаване (до 20 метра), което го прави подходящ избор при нужда от отдалеченост на сензора от основното приемащо устройство, както и при нужда от работата му в неблагоприятна околна среда (температурният му диапазон е в широки граници) или монтирането му на малка площ и ограничено пространство.

- Параметри / спецификация на устройството:

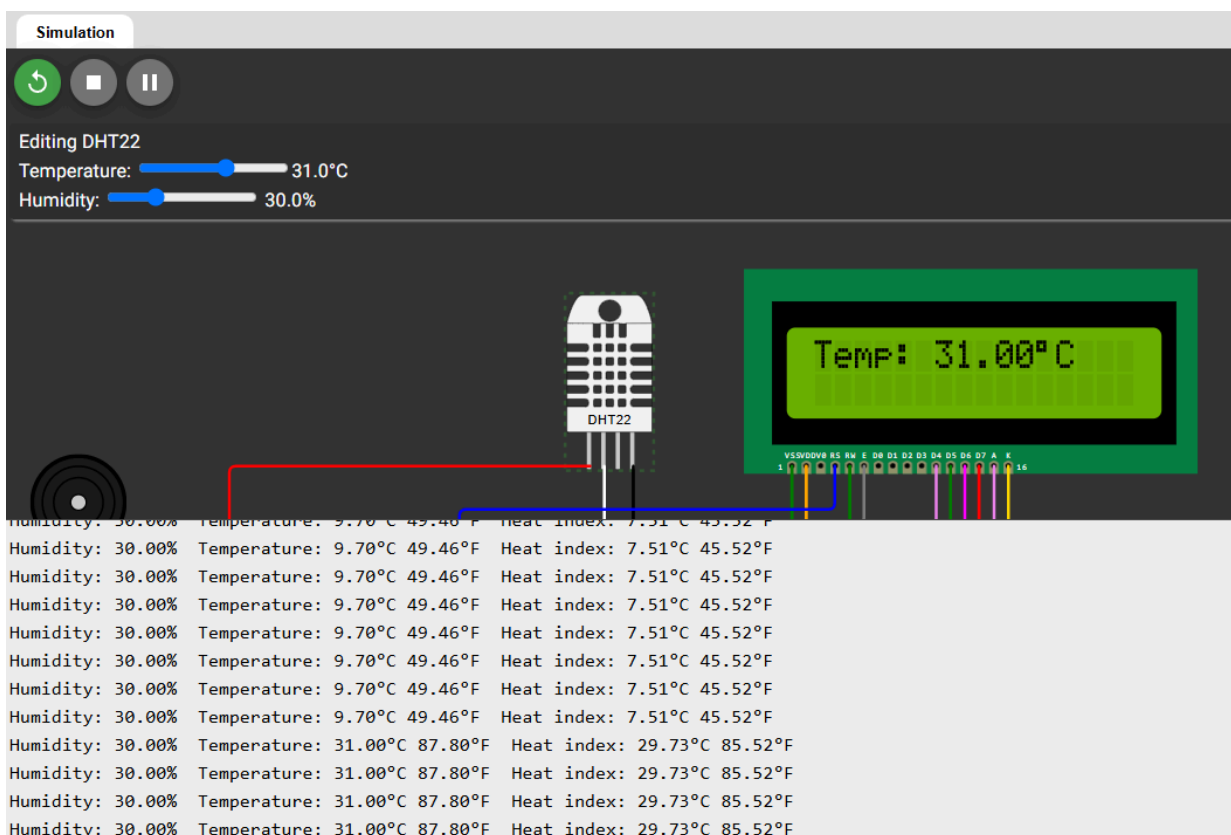
Параметър	Стойност
Захранване	3.3V - 6V DC
Изходен сигнал	Дигитален (еднолинейна шина)
Сензорен елемент	Полимерен кондензатор

Работен диапазон	Влажност: 0% - 100% RH Температура: -40°C до 80°C
Точност	Влажност: $\pm 2\%$ RH (макс. $\pm 5\%$) Температура: $\pm 0.5^\circ\text{C}$
Резолюция	Влажност: 0.1% RH Температура: 0.1°C
Повторяемост	Влажност: $\pm 1\%$ RH Температура: $\pm 0.2^\circ\text{C}$
Хистерезис (влажност)	$\pm 0.3\%$ RH
Дългосрочна стабилност	$\pm 0.5\%$ RH на година
Период на измерване	Средно: 2 секунди
Взаимозаменяемост	Напълно взаимозаменяем
Размери	Малък: 14 × 18 × 5.5 mm Голям: 22 × 28 × 5 mm

5. Схема на свързване на устройството и микроконтролера



Фиг 3. - Схема на свързване



Фиг 4. - Работа на устройството в симулацията на WOKWI

7. Програмен код за базово отчитане и принтиране на стойности в серийния монитор:

```
#include "DHT.h"
#define DHTPIN 2
#define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321
DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600);
  Serial.println("DHT22 test!");
  dht.begin();
}

void loop() {
  // Изчакване 2 секунди преди ново измерване
  delay(2000);
  // Отчитане влажността
```

```

float h = dht.readHumidity();
// Отчитаме температурата в градуси по Целзий (default)
float t = dht.readTemperature();
// Отчитаме температурата в градуси по Фаренхайт (isFahrenheit = true)
float f = dht.readTemperature(true);

// Проверка дали някоя от стойностите е NaN - Not a Number,
// което е индикатор за грешка при отчитането.
if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println(F("Failed to read from DHT sensor!"));
    return;
}

// Изчисляване на температурния индекс по Фаренхайт
float hif = dht.computeHeatIndex(f, h);
// Изчисляване на температурния индекс по Целзий (isFahreheit = false)
float hic = dht.computeHeatIndex(t, h, false);
// Извеждане на резултатите в серийния монитор
Serial.print(F("Humidity: "));
Serial.print(h);
Serial.print(F("%   Temperature: "));
Serial.print(t);
Serial.print(F("°C "));
Serial.print(f);
Serial.print(F("°F   Heat index: "));
Serial.print(hic);
Serial.print(F("°C "));
Serial.print(hif);
Serial.println(F("°F"));
}

```

8. Програмен код, за реализация на устройство за мониторинг на температура, извеждането и на LCD дисплей и възпроизвеждане на звуков сигнал под и над дадени критични нива. Основната разлика с предходния код е в извеждането на отчетените данни, като тук вместо към серийния монитор използваме LCD дисплей (1-ви от двата реда) и говорител за възпроизвеждане на сигнал

```

#include "DHT.h"
#include <LiquidCrystal.h>
#define DHTPIN 2

```

```

#define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321
DHT dht(DHTPIN, DHTTYPE);

const int SPEAKER_PIN = A3;
LiquidCrystal lcd(12, 11, 10, 9, 8, 7);

void setup() {
    Serial.begin(9600);
    Serial.println(F("DHTxx test!"));
    lcd.begin(16, 2);
    dht.begin();
    lcd.print("Initialized!");
}

void loop() {
    // Изчакване 2 секунди преди ново измерване
    delay(2000);

    // Отчитаме влажността
    float h = dht.readHumidity();
    // Отчитаме температурата в градуси по Целзий (default)
    float t = dht.readTemperature();
    // Отчитаме температурата в градуси по Фаренхайт (isFahrenheit = true)
    float f = dht.readTemperature(true);

    // Проверка дали някоя от стойностите е NaN - Not a Number,
    // което е индикатор за грешка при отчитането.
    if (isnan(h) || isnan(t) || isnan(f)) {
        Serial.println(F("Failed to read from DHT sensor!"));
        return;
    }

    // Изчисляване на температурния индекс по Фаренхайт
    float hif = dht.computeHeatIndex(f, h);
    // Изчисляване на температурния индекс по Целзий (isFahreheit = false)
    float hic = dht.computeHeatIndex(t, h, false);

    // Това е кода необходим за надграждане до устройство с реално приложение:
    // Датчик за температура с LCD дисплей за извеждане на стойността, както и
    // говорител за възпроизвеждане на звуков сигнал при достигане на

```



```
// критични стойности

// задаваме курсора на първа позиция на първия ред на LCD екрана
lcd.setCursor(0,0);
lcd.print("Temp: ");
lcd.print(t);
// така принтираме специалния символ за "градус"
lcd.print((char)223);
lcd.print("C");
// проверка за критични нива (под -20 и над 45 градуса)
if(t>45 || t<-20){
    // Възпроизвеждане на тон
    tone(SPEAKER_PIN, 89);
}else{
    // Спиране възпроизвеждането на тон
    noTone(SPEAKER_PIN);
}
}
```

ЛИНК КЪМ ПРОЕКТА В WOKWI: <https://wokwi.com/projects/422499604817154049>