



Курсова работа

НА

Симеон Симеонов Фак.№ 614062

Здравко Иванов Фак.№ 614050

Петко Колев Фак.№ 614048

Специалност: Компютърни системи и технологии

Образователно-квалификационна степен: Магистър

Ръководител:

Доц. Д-р Николай Каканаков

Пловдив 2022



1. Дефиниране на задачата

Да се разработи разпределена вградена система за следене на осветление и температура в помещение, като подава команди за управление на база отчитанията. Управлението на осветлението да бъде реализирано чрез цифрови изходи през релета. Контролът на температурата трябва да бъде реализиран посредством изпращане на команди към климатик.



2 Проектиране на решение

1. 21 Избор на хардуерни компоненти

- Избрана е платка Arduino Mega 2560 поради достатъчния си размер памет, аналогови и цифрови портове, както и развойната среда Arduino.
- За реализирането на функционалността на управлението на светлината е избран сензорът за светлина TЕМТ6000 както и модул с две петволтови релета.
- За реализирането на функционалността на управлението на температурата е избран терморезистор NTC 100kOhm за отчитане на температурата и LED инфрачервен диод за изпращане на сигнали.
- За реализирането на веб интерфейса е избран мрежов модул ENC28J60 SPI с RJ45 конектор, както и мрежов рутер TP-LINK.

2 22 Избор на платформа за развой

Използвана е средата Arduino IDE. Тя, както и платката са продукт на фирмата Arduino, която разработва хардуер и софтуер с отворен код. Средата позволява използването на C и C++, използвайки стандартно API по-известно като езика Arduino.



3. Описание на реализацията

3.1 Описание на хардуерната реализация

Портовете на сензора за светлина ТЕМТ6000 са свързани както следва:

- „S” -> аналоговия порт A0 на основната платка.
- „G” -> порт GND на основната платка.
- „U” -> порт 5V на основната платка.

Портовете на модула с релета са свързани както следва:

- „G” -> порт GND на основната платка.
- „IN1” -> PWM порт 3 на основната платка.
- „IN2” -> PWM порт 2 на основната платка.
- „VCC” -> порт 5V на основната платка.

Връзките на терморезистора са реализирани както следва:

- „+” -> порт 5V на основната платка.
- „-” -> аналогов порт A2 на основната платка.

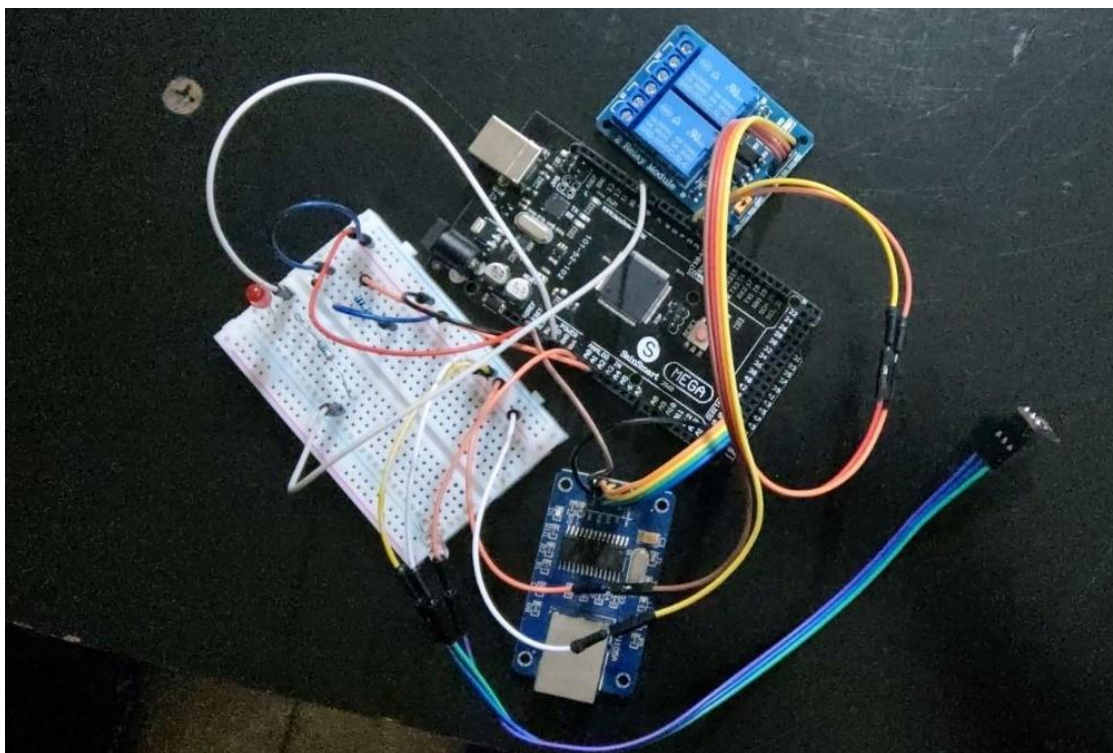
Връзките на инфрачервения диод са реализирани както следва:

- „+” -> PWM порт 9 на основната платка.
- „-” -> порт GND на основната платка.

Портовете на мрежов модул ENC28J60 SPI са свързани както следва:



- „VCC” -> порт 3.3V на основната платка.
- „GND” -> порт GND на основната платка.
- „CS” -> дигитален порт 53 на основната платка.
- „SCK” -> дигитален порт 52 на основната платка.
- „SI” -> дигитален порт 51 на основната платка.
- „S0” -> дигитален порт 50 на основната платка.





3.2 Описание на софтуерната реализация

Платформата Arduino предлага огромно разнообразие от различни библиотеки поради факта, че разчита на отворен код. Използвани са няколко библиотеки:

- IRemote – Тя предоставя функционалност за изпращане и получаване на инфрачервени сигнали.
- UPEthernet – Тя предоставя функционалност за превръщането на платката в сървър.

Част от реализацията на уеб интерфейса се осъществява от следния фрагмент код:

```
byte mac[] = { 0xAE, 0xBD, 0xBE, 0xDF, 0xFE, 0xED };
IPAddress ip(192, 168, 0, 177);
EthernetServer server(80);

String html = "<!DOCTYPE html> <html lang=\"en\"><hea

void ether2 () {
    EthernetClient client = server.available();
    if (client) {
        Serial.println("new client");
        boolean currentLineIsBlank = true;
        while (client.connected()) {
            if (client.available()) {
                c = client.read();
                buffer += c;
                buffer2 += c;
                if (c == '\n' && currentLineIsBlank) {
                    client.println(html);
                    break;
                }
            }
            if (c == '\n') {
                currentLineIsBlank = true;
            }
        }
    }
}
```

Използвани са глобални променливи за задаване на MAC и IP адреси. Чрез „server” променливата се инициализира сървър, който реализира функционалността платката да се свързва с мрежа. В променливата html се съдържа уеб интерфейса, който сървър предоставя на клиента. След което във функцията ether2 на променливата „client” се присвоява свързаният вече клиент. С помощта на методите „client.connected”, „client.available”, „client.read”,



променливите „buffer”, „buffer2” и „с”, както и на серия от проверки се реализира „прочитането” на получените от клиента данни.

Дизайнът на потребителския интерфейс е следният:

Light Controls	Temperature Controls
Choose luminosity treshold: Low	Choose temperature treshold: 22
<input type="button" value="Submit"/>	

Реализирана е функция за измерване температура посредством термистор. Част от реализацията се осъществява от следния фрагмент код:

```
void temperature()
{
    Vo = analogRead(ThermistorPin);
    R2 = R1 * (1023.0 / (float)Vo - 1.0);
    logR2 = log(R2);
    T = (1.0 / (c1 + c2 * logR2 + c3 * logR2 * logR2 * logR2));
    T = T - 273.15;
    delay(1);
}
```

Измерват се входните и изходните напрежения по веригата с термистора и по дадена математическа формула се изчислява стойността на температурата.

Част от функционалността за предаване на инфрачервен сигнал към устройство е реализирана от следния фрагмент:

```
if ( T <= (tempStatus) && !on_off) {
    for (int i = 0; i < 3; i++) {
        mySender.sendNEC(0x20DF10EF, 32);
        delay(40);
    }
    on_off=true;
}
else if(T >= (tempStatus+3 && on_off)
{
    for (int i = 0; i < 3; i++) {
        mySender.sendNEC(0x20DF10EF, 32);
        delay(40);
    }
    on_off=false;
}
```



Посредством проверка на сигналите за температурата се изпращат три сигнала към инфрачервения приемник. Използва се метода „sendNEC”, като се подават като аргументи кода на командата и дължината на сигнала.

Основната част от програмата, която реализира управлението на светлината е следната:

```
value = analogRead(temt6000Pin);  
if (value <= lightStatus) {  
    digitalWrite(2, LOW); //turn on  
}  
if (value > lightStatus) {  
    digitalWrite(2, HIGH); //turn off  
}
```

Прочита се стойността, подадена от сензора за светлина и се преобразува в цели числа от 0 до 1023. При сравняване на получената стойност с тази, зададена в уеб интерфейса, съответно се изпраща сигнал за включване или изключване на релето.



4. Разпределение и тестване на задачите

4.1. Разпределение

Петко Колев – Свързване на светлинния сензор и модула с релетата. Реализиране на програмната част за управление на релетата.

Здравко Иванов – Свързване на термистора и инфрачервения диод. Реализиране на програмната част за управление на температурата.

Симеон Симеонов – Свързване на ethernet модула. Реализиране на програмната част за уеб интерфейса.

4.2 Тестване

- Светлина – частта тества независимо като използва изкуствен източник на светлина и го насочва към светлинния сензор. Подсигурена е стойност по подразбиране, която е зададена преди потребителя да избере гранична стойност от уеб интерфейса. Поради липсата на специализирани измервателни уреди за светлина, граничните стойности са подбрани така, че системата да включва осветлението и така да поддържа нормална обективна осветеност. При приближаване на източника на светлина и надвишаване на граничната стойност се очаква релетата да се изключат. Опитите показват, че при различно разстояние, в зависимост от избраната гранична стойност системата работи в очакваните граници.

- Температура – за тестване на тази част използваме телесна топлина като топлинен източник. Отново са подбрани гранични стойности, които отговарят на подходящи такива за отопление/охлаждане на помещение. В случай на отопление при пад на AMBIENTната температура под зададената гранична, системата изпраща сигнал към устройство (в този случай тествано с телевизор) за включване. При надвишаване на границата с повече от 3°C се изпраща сигнал за изключване. Опитите показват, че при започване на симулация с температура по-ниска от граничната и симулация на покачване на температурата системата включва и изключва уреда (телевизора) по очаквания начин.



- Цялостна система – При цялостно тестване на системата всички сензори работят в очакваните граници, както и отчетените стойности в очакваните граници. Изпращаните от системата команди работят коректно спрямо зададените от потребителя параметри.



5. Линкове

Библиотеки:

<https://docs.arduino.cc/hardware/mega-2560>

<https://github.com/Arduino-IRremote/Arduino-IRremote>

<https://github.com/UIPEthernet/UIPEthernet>

Тестове:

<https://www.youtube.com/watch?v=uCPvkV3I6w0>

<https://www.youtube.com/watch?v=4OkcmRxD4fc>

<https://www.youtube.com/watch?v=1ZjMiR8dTT4>