|  |  |
| --- | --- |
| D:\UKTC\UKTC tempatets\Logo\UKTC-logo-color.png | **Национална професионална гимназия по компютърни технологии и системи**  **гр.Правец при технически университет – СОФИЯ** |

**Дипломна работа**

ДЪРЖАВЕН ИЗПИТ ЗА ПРИДОБИВАНЕ НА ТРЕТА СТЕПЕН НА ПРОФЕСИОНАЛНА КВАЛИФИКАЦИЯ – ЧАСТ ПО ТЕОРИЯ НА ПРОФЕСИЯТА

на

Ученик: Мирослав Станимир Котларов

ученик от 18122 (XII) клас

Тема: **Цифров терморегулатор с хистерезисно управление базиран на ESP32**

**ПРОФЕСИОНАЛНО НАПРАВЛЕНИЕ:** 481 КОМПЮТЪРНИ НАУКИ

**ПРОФЕСИЯ:** 481020 системен програмист

**СПЕЦИАЛНОСТ:** 4810201 системно програмиране

Ученик: ………………….. Ръководител: ………………………

*инж. Венцислав Начев*

Правец, 2023

Съдържание

[Увод 2](#_Toc133336730)

[Глава 1 Литературен обзор 3](#_Toc133336731)

[Глава 2 Използвани технологии 20](#_Toc133336732)

[Глава 3 Разработка и реализация 26](#_Toc133336733)

[Заключение 35](#_Toc133336734)

[Използвана литература 36](#_Toc133336735)

[Списък с използваните означения и съкращения 37](#_Toc133336736)

# Увод

Проектът има за цел да регулира въведена температура и да я поддържа за необходимото време на потребителя. Нужен е микроконтролер (ESP32), в който да бъде създаден хистерезисен алгоритъм за регулиране на температурата. Температурата ще бъде засичана с помощта на температурен сензор (Pt100). А изходът на контролерът ще бъде релеен, което ще изключва нагревателя(система за регулиране на захранването). Температурите, с които ще се работи са в диапазон от 50°C - 240°C . Въвеждането на желаната температура се изпълнява чрез генерираната на локална мрежа уеб страница от ESP32.

Този процес е напълно автоматичен и намалява необходимите усилия на потребителят, за следене на температурата. Начинът по който работи е при надвишаване на въведената температура, системата се изключва и при понижаване на минималната граница, която системата изчислява, системата се пуска.

# Глава 1 Литературен обзор

1. Какво е терморегулатор?

Терморегулатор или термостат е устройство за поддържане на постоянна температура. Най – простият терморегулатор се състои от: Нагряващ елемент, датчик за измерване на точна температура и регулиращ елемент, който да регулира температурата до желаната стойност.

1. Видове терморегулиране

Съществуват няколко вида за регулиране на температурата на нагряващ се елемент, това могат да бъдат: ON/OFF регулиране, регулиране чрез PID алгоритъм и регулиране чрез изкуствен интелект.

* 1. ON/OFF регулатори.

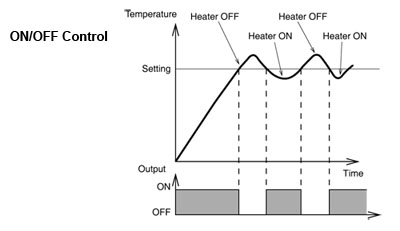
Регулирането на температура чрез включване и изключване е най-простият начин да регулираме температура. Такива терморегулатори срещаме в ежедневието и нормално са с фиксирани стойности. Регулирането на температурата става чрез изключване и включване на захранването. Трудно могат да бъдат подържани точни температури с такъв начин на регулиране на температурата.



*Фиг. 1.1 – нагревател с фиксирани стойности за регулиране*

* 1. ON/OFF регулатори управлявани от микроконтролер

Ако желаем потребителят да въвежда стойностите тогава ще бъдат необходими микроконтролер и термосензор свързан към него. Микроконтролерът ще приема две стойности въведени от потребителят (между двете стойности трябва да има разлика) и температурата ще бъде регулирана чрез хистерезис(фигура 1.2). Важно е между двете въведи температури да има разлика – това афектира броят на превключвания в системата - при малка разлика в температурите се стига до по-често превключване, което в следствие довежда до повреда (препоръчително е разликата между температурите да бъде около 15 градуса). С помоща на термосензорът можем да получаваме точни температурни стойности за нагревателят.



*Фиг. 1.2 – схема на хистерезис*

* 1. Регулиране с PID алгоритъм

С помоща на PID алгоритъма можем да достигнем висока прецизност при управлението на нагряващият се елемент. Но какво е PID алгоритъм и как работи?

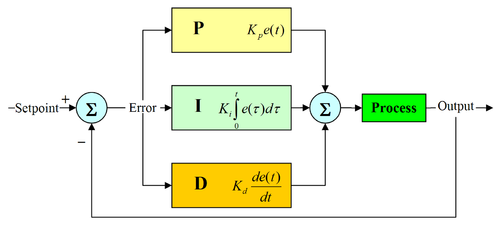
PID (Proportional Integral Derivative/Пропорционален Интегрален Деференциален) алгоритъм е метод за управление с обратна връзка. Неговото приложение се намира почти навсякъде където е нужно подържането на някаква величина, това може да бъде: температура, скорост, център на маса(баланс), подържане на налягане и много други. Целта на алгоритъма е автоматизацията на технологични процеси, в които човек трудно би успял да постигне. Той съществува от 20-те години на 20-ти век и към днешна дата е най-масово разпространеният регулатор.



*Фиг. 1.3 – термпературен контролер с PID управление*

Основният принцип на работа е поправяне на грешка, получена между предварително установена стойност и текуща стойност на регулиращата се величина. Той притежава един вход и един изход (SISO - Single Input Single Output/Един вход един изход). Входът е получената грешка(разликата между зададената и текуща измерена стойност на регулираната величина). А изходът е неговото управляващо въздействие, което може да е електрическо напрежение, честота на ел. ток, налягане или друго. Изходът се изчислява на базата на трите му съставни части:

* P – пропорционална – играе роля на моментална и директна реакция, например при рязко намаляне на скорост в двигател – подава се напрежение с цел увеличаване на скоростта.
* I – интегрална – смисълът ѝ е да премахва установена грешка при промяна на товар, например установяването на обороти при промяна на товар в превозно средство
* D – диференциална – реагира към бързите промени на грешката и по-бързо затихване на грешката, например когато температурата започне да се покачва, още преди да стигне желаното ниво контролерът спира нагревателя, за да не стигне до прегряване.



*Фиг. 1.4 – паралелна схема на ПИД регулатор*

* 1. Регулиране чрез изкуствен интелект

С напредването на технологиите в 21-ви век вече навлизат нови идеи за подобряване на текущите регулатори. Това са регулаторите управлявани от изкуствен интелект. Те са свързани към Wi-Fi мрежата в къщата и работата с тях е изключително лесна. Потребителят трябва да притежава смартфон или друго устройство, което да може да подържа приложението и да може да се свърже към интернет мрежата. Процесът по регулиране го извършва изцяло изкуственият интелект, намирайки най-ефективният метод за изпълняване спестявайки ток и усилията на потребителят. Те са напълно препрограмируеми и имат много и различни функции, които могат да бъдат прилагани за по-удобна работа на потребителят.



*Фиг. 1.5 – терморегулатор с изкуствен интелект*

Специално свойство на умните уреди е възможността им да бъдат интегрирани с други умни устройстсва, позволявайки още повече контрол и спестена енергия.

1. Какво е температурен сензор и какви са неговите разновидности?

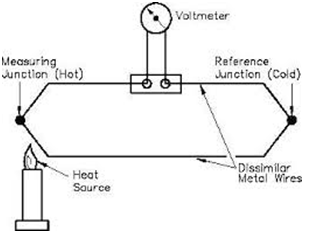
Температурата е една от най-често измерваните величини и следователно не е изненадващо, че има много начини за нейното усещане. Отчитане на температурата може да се осъществи чрез директен контакт с източника на отопление или дистанционно, без директен контакт с източника. Днес на пазара има голямо разнообразие от температурни сензори, включително термодвойки, съпротивляващи температурни детектори (RTD), термистори, полупроводникови сензори и други.

* 1. Термодвойка

Това е тип температурен сензор, който се прави чрез свързване на два различни метала в един край. Съединеният край се нарича HOT JUNCTION (топла връзка). Другият край на тези разнородни метали се нарича COLD JUNCTION (студена връзка). Ако има разлика в температурата между горещата и студената връзка, се създава малко напрежение. Съществуват два типа: тип K и тип J. Разликата между типовете са металите от които са направени двете връзки.



*Фиг. 1.6 – термодвойка*



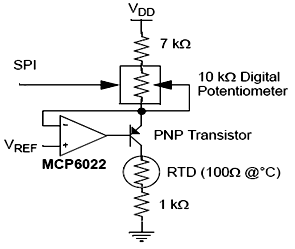
*Фиг. 1.7 – схема на термодвойка*

* 1. RTD (съпротивляващ температурен детектор)

RTD е устройство за измерване на температурата, чието съпротивление се променя с температурата. Обикновено са изработени от платина, въпреки че устройствата, изработени от никел или мед, не са необичайни, RTD могат да приемат много различни форми като намотана тел, тънък филм и други. За да се измери съпротивлението на температурния детектор, е нужно прилагането на постоянен ток, чрез измереното полученото напрежение може да се определи съпротивлението на детектора. Схемата, показана на фигура 9, е източник на постоянен ток, използващ еталонно напрежение, един усилвател и PNP транзистор.



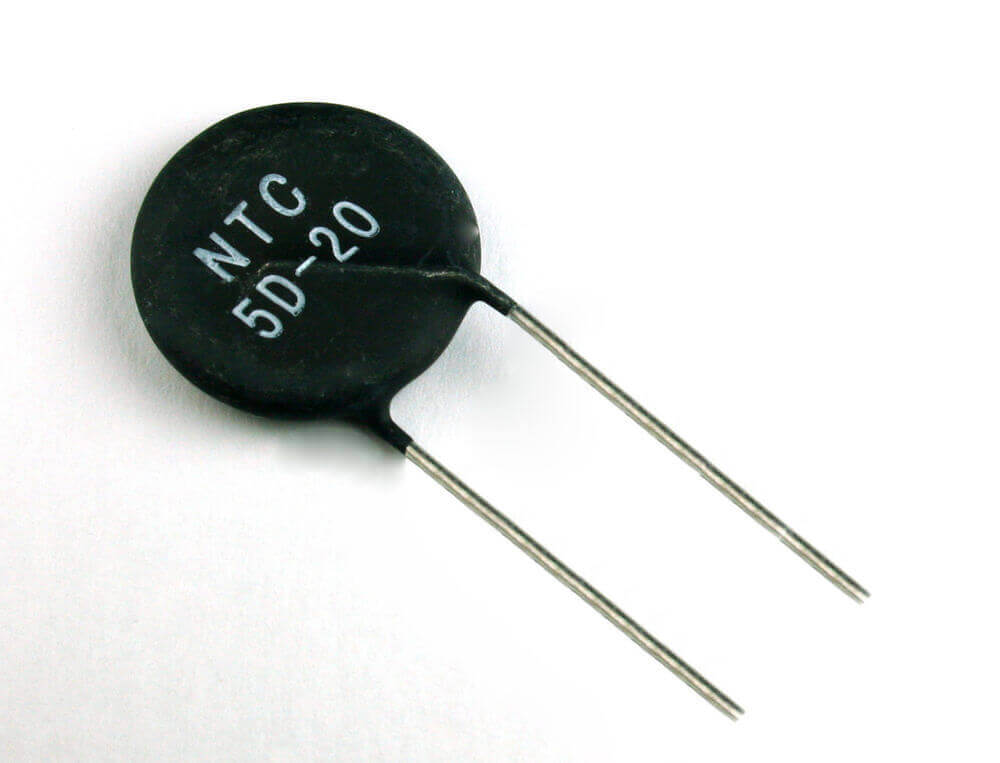
*Фиг. 1.8 – съпротивляващ температурен сензор (RTD)*



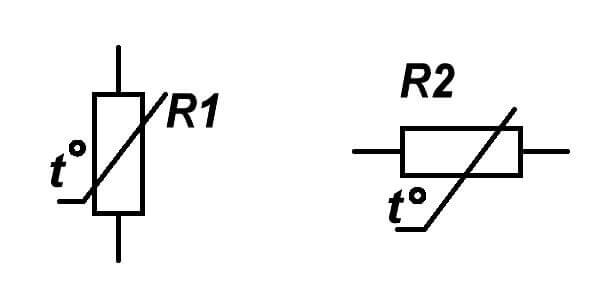
*Фиг. 1.9 – схема използваща RTD*

* 1. Термистори

Подобно на RTD, термисторът е сензор за температура, чието съпротивление се променя с температурата. Термисторите обаче са направени от полупроводникови материали. Съпротивлението се определя по същия начин като RTD, но термисторите показват силно нелинейна крива на съпротивление спрямо температура. По този начин, в работния диапазон на термисторите, можем да видим голяма промяна на съпротивлението за много малка промяна на температурата. Това прави термисторът идеален за високочувствителни устройства.



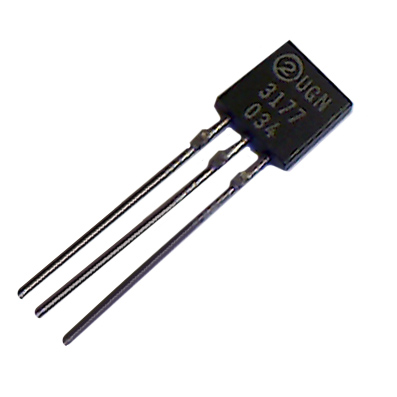
*Фиг. 1.10 – термистор*



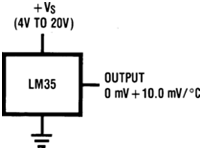
*Фиг. 1.11 – знак за обозначаване на термистор в схема*

* 1. Полупроводникови сензори

Полупроводниковият сензор е устройство, което използва полупроводникови елементи за измери температурата. Те притежават висока точност, ниска цена и малък размер, което води до високото им приложение в индустрията.



*Фиг. 1.12 – полупроводников сензор за температура*



*Фиг. 1.13 – схема използваща полупроводников сензор*

1. Какво е микроконтролер?

Микроконтролерът е компютърен чип, който съдържа всички основни компоненти на един компютър - централен процесор, памет за програми и данни, входно/изходни портове и други периферни устройства в един малък интегрален схемен пакет. Той е проектиран да изпълнява специални функции в една система или устройство, като управлява електронните сигнали, комуникациите, сензорите и други периферни устройства.

* 1. Платформа Arduino Uno

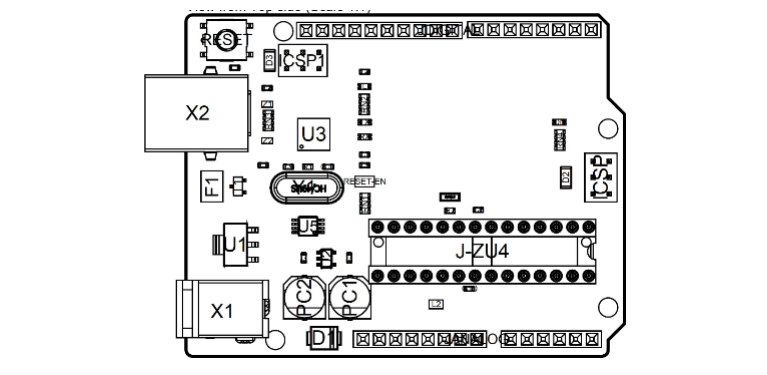
Платформата Arduino Uno позволява управлението на системи и устройства. Тя е най-продаваната платка за разработка поради изключително ниската ѝ цена, лесната работа с нея и универсалното ѝ приложение почти навсякъде.



*Фиг. 1.14 - Arduino Uno*

* 1. Основни характеристики на Arduino Uno

Процесорът, който управлява цялата платка е ATmega328p. Той е 8 битов със скорост от 16MHz. Flash памет от 32kb(памет за големи количества информация), SRAM от 2kb(оперативна памет), EEPROM от 1kb(памет за малки количества информация).

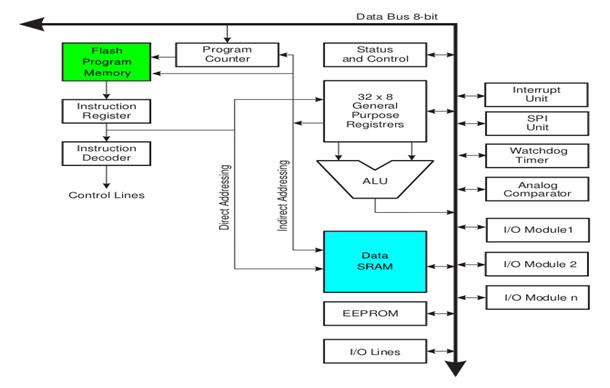


*Фиг. 1.15 - Топология на платката*

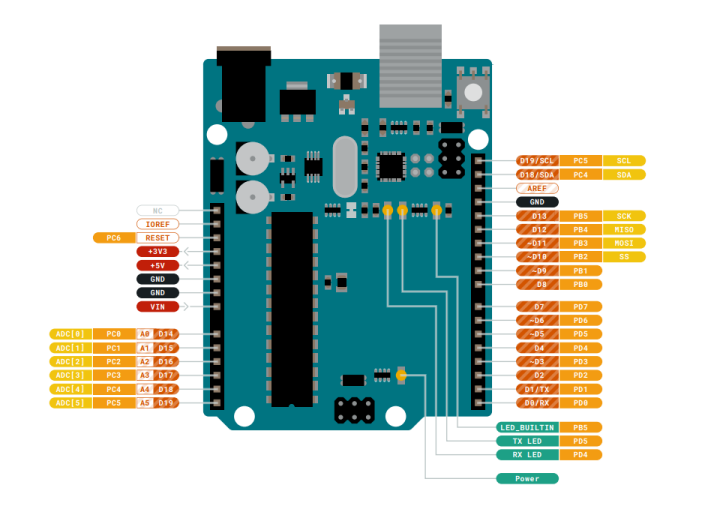
Съществуват два начина за захранване на платката: чрез VIN(обозначен на фигура 1.15 с X1) или чрез USB-B(обозначен на фигура 1.15 с X2) портовете. Максималното напрежение, което може платката да приеме от захранващ кабел през VIN порта е 20V, а максималното напрежение от USB-B порта е 5.5V.

Един Arduino Uno микроконтролер притежа общо 32 пина, от които 13 служат като вход и изход с общо предназначение, 6 входа за аналоговият цифров преобразувател и останалите 10 служат за подържане на електрическа верига(това са пинове като +5V постоянно напрежение, пин за земя и други). Във фигура 2.3 се вижда схема с разположението на всеки един от пиновте върху микроконтролера.

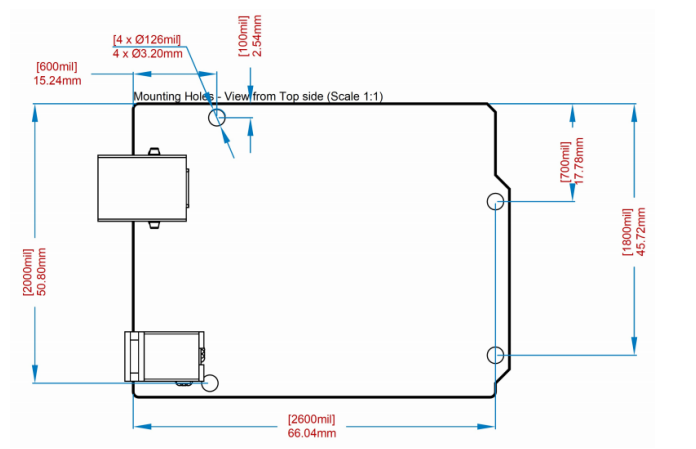
Процесорът, който управлява цялата платка е ATmega328p. Той е 8 битов със скорост от 16MHz. Flash памет от 32kb(памет за големи количества информация), SRAM от 2kb(оперативна памет), EEPROM от 1kb(памет за малки количества информация).



*Фиг. 1.16 – Блокова схема на архитектурата на Arduino*



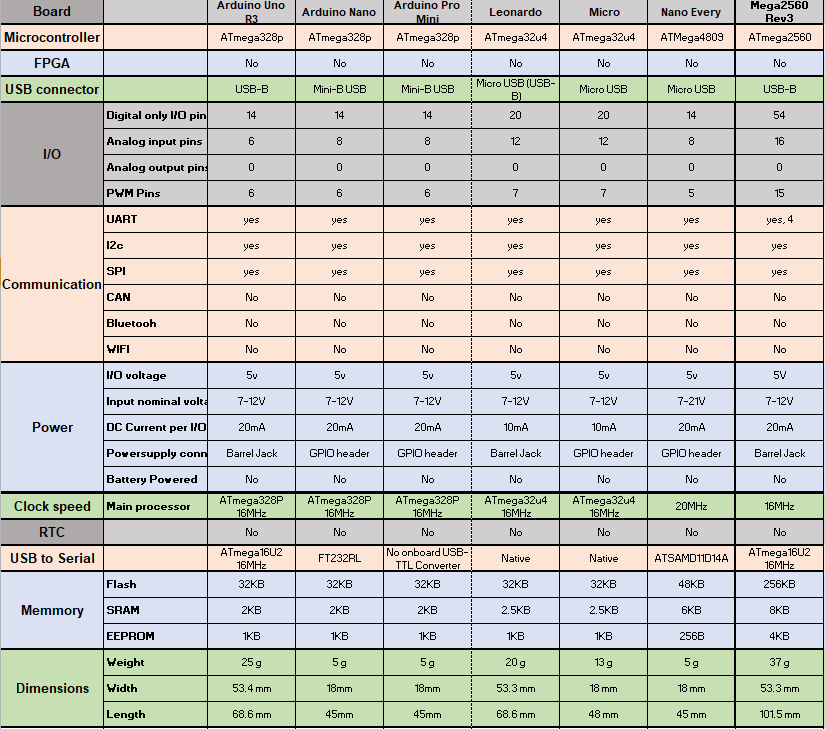
*Фиг. 1.17 – Схема на пиновете на Arduino Uno*



*Фиг. 1.18 – Размери на Arduino Uno*

* 1. Други разновидности на микроконтролерите на Arduino

Съществуват други разновидности, които са показани в таблицата на фигура 1.19. На таблицата са представени всички характеристики на някои от най-известните платки на Arduino.



*Фиг. 1.19 - Таблица с разновидности на други микроконтролери на Arduino*

1. Къде терморегулаторите намират приложение в практиката?

Очевидно е че необходимостта от температурни регулатори се изисква в почти всяка индустрия тъй като повечето физически, електронни, химични, механични и биологични системи са засегнати от температурата. Някои химически реакции, биологични процеси и дори електрически вериги се представят най-добре в ограничени температурни диапазони. Примери за това могат да бъдат в:

* Автоматизираната индустрия - използват се в промишлени процеси за контрол на температурата на материалите и оборудването.



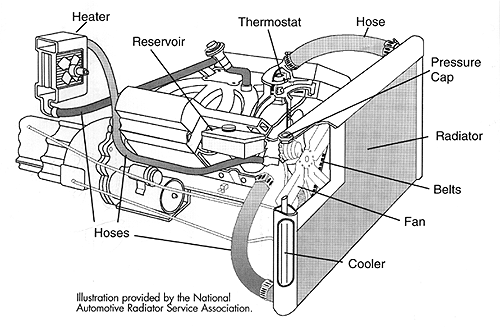
*Фиг. 1.14 – пример за автоматизирана индустрия*

* Медицинско оборудване - използват се за контрол на температурата в медицинското оборудване като инкубатори и хладилни камери.



*Фиг. 1.15 – апарат за съхранение на кръв*

* Автомобилна индустрия - използват се в автомобилната индустрия за контрол на температурата на охлаждащата течност на двигателя.



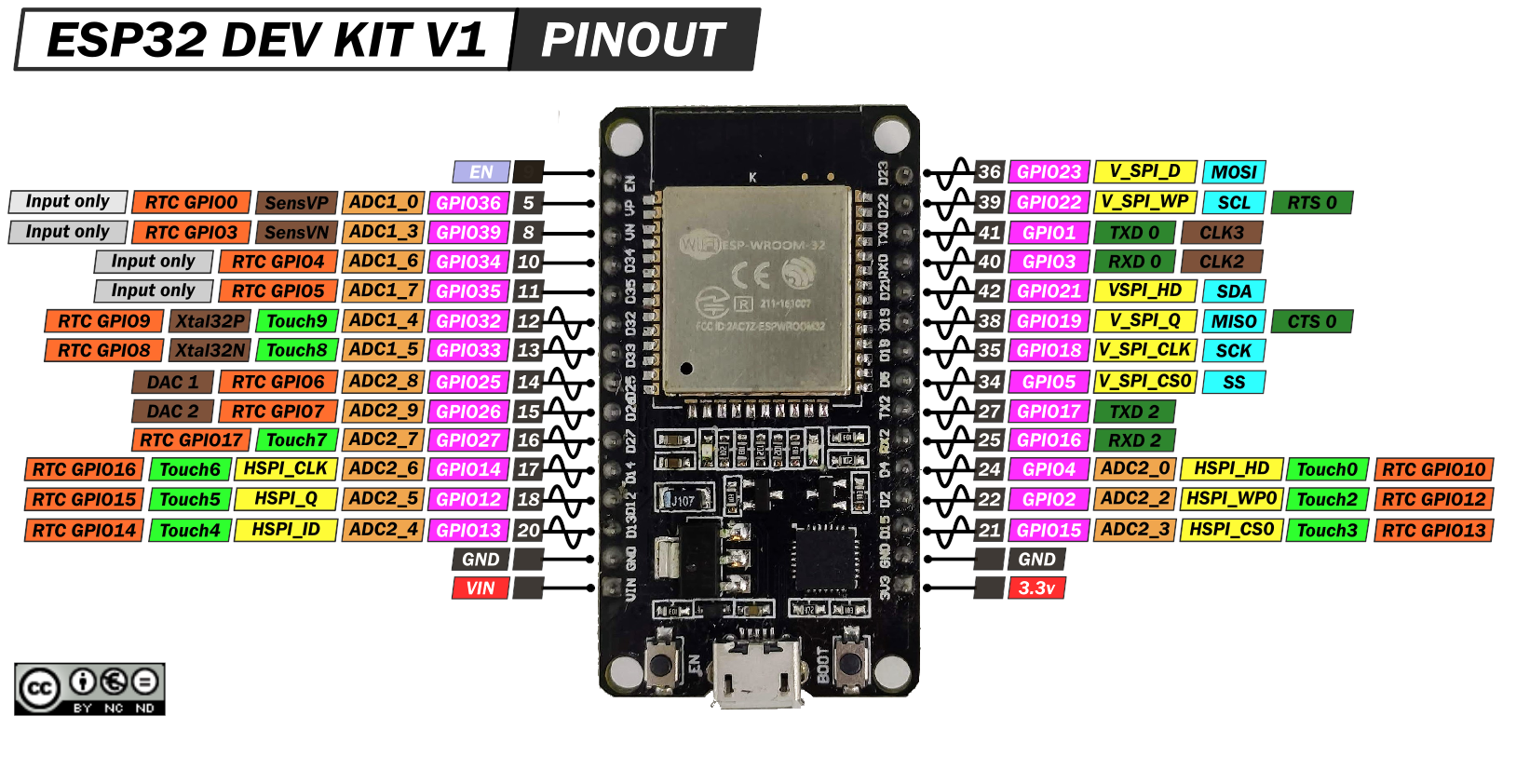
*Фиг. 1.16 – охладителна система на двигател*

# Глава 2 Използвани технологии

1. Микроконтролер ESP32

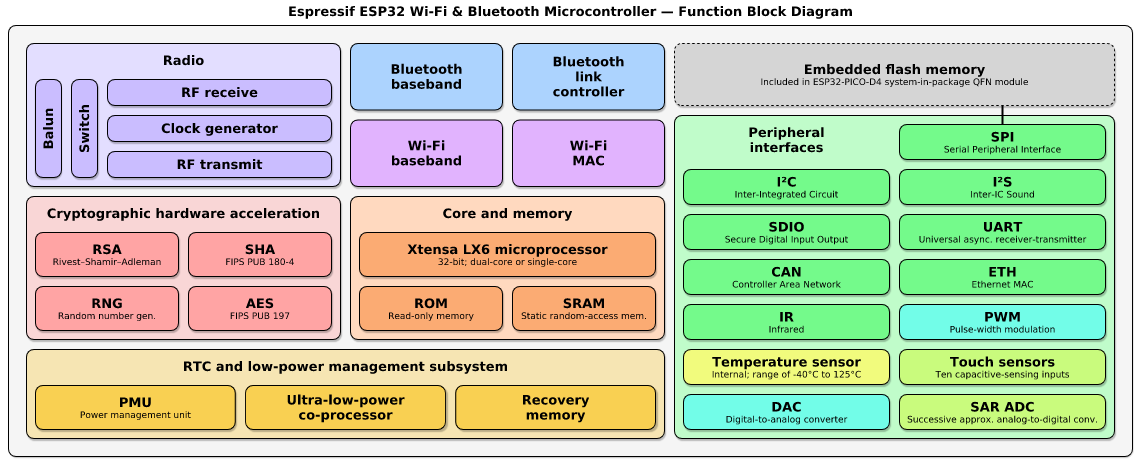
ESP32 е микроконтролер, който включва в себе си двуядрен процесор, WiFi и Bluetooth модули, както и много други периферни устройства. Той е създаден от китайската компания Espressif Systems и е един от най-популярните микроконтролери за IoT (Интернет на нещата). Може да се програмира с различни езици за програмиране, включително C и Python, и поддържа различни инструменти за разработка на софтуер, като Arduino IDE, ESP-IDF и други.

ESP32 има множество приложения, включително управление на домашни уреди, автоматизация на производствени процеси, IoT сензорни мрежи и др. Той също така може да бъде използван за разработване на умни устройства като умни часовници, фитнес тракери и други неща.



*Фиг. 2.6 - ESP32 и диаграма на пиновете му.*

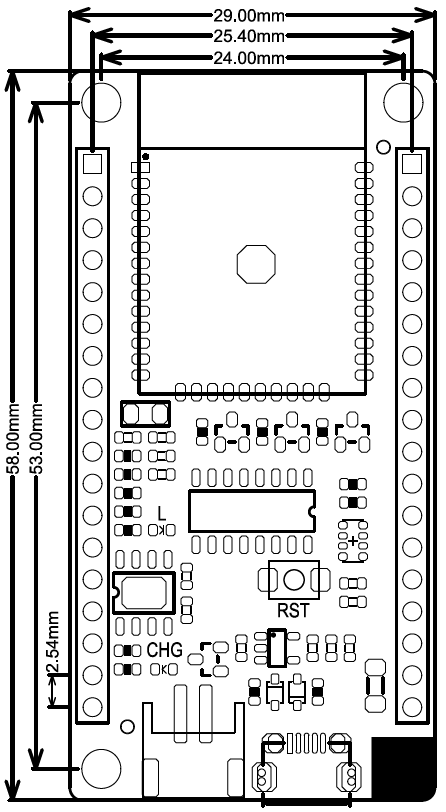
Основната роля на микроконтролерът в проектът е да поддържа сайт на локална мрежа, където потребителят да може да въведе желана температура и след това то да ги обработва и да връща резултат (изпращане на сигнал към Arduino микроконтролерът да включи или изключи системата).

* 1. Характеристики на ESP32

*Фиг. 2.7 – Блокова схема на функционалностите на ESP32*

ESP32 притежава двуядрен процесор с честота до 240MHz, вграден WiFi модул, който поддържа 802.11 b/g/n стандарт, вграден Bluetooth модул, който поддържа Bluetooth v4.2 и Bluetooth Low Energy (BLE) стандарти, вградена памет от 520KB SRAM и 4 MB Flash памет.

Wi-Fi модулът на ESP32 има вградена антена, която позволява безжична комуникация с други устройства, свързани към една и съща Wi-Fi мрежа. Други функции които модулът притежава са: Възможност за работа в различни честотни ленти, включително 2.4GHz и 5GHz, вградени фукции за проверка на качеството на сигнала и автоматична реконекция при загуба на връзката. Също така поддръжа различни протоколи за комуникация като: TCP, UDP, HTTP, HTTPS, MQTT и други.



*Фиг. 2.8 – Размери на ESP32*

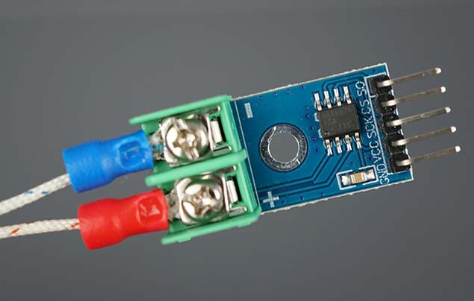
1. Термпературен сензор и усилвател

За да отчитаме точната температура на нагряващият се елемент е необходим термосензор, който да има пряк контакт с нагряващата се повърхност. Важно е да се преценят температурните диапазони, с които ще се работи. За проекта е необходим температурен сензор в диапазона 0 - 250°C +. След проучване избрах термодвойка направена от никелхром, тип К с работен диапазон от 0-800°C дължината на кабела е 100см и кабелите са с висока температурна изолация(фигура 2.6).



*Фиг. 2.7 – използвана термодвойка*

Освен термодвойката която сме избрали ще ни бъде необходима още една част. Това е усилвател за температурни сензори от тип К. Неговата роля е да чете изпратените сигнали от сензора и да ги направи четими за контролера. Усилвателят, който избрах е MAX6675 amplifier.



*Фиг. 2.8 – усилвател MAX6675*

1. Нагревател

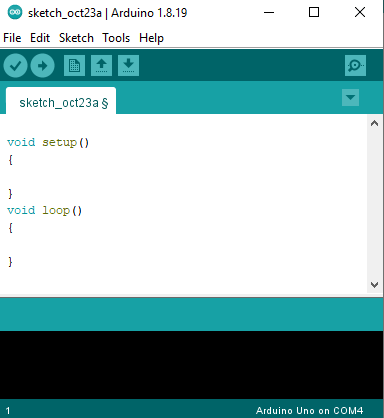
За нагревател може да се използва всякакъв нагряващ се елемент, но е важно термосензора да принадлежи на диапазона в противен случай може да се доведе до повреда на термосензора. За моят проект избрах електрическият котлон на Muhler MHP-155B(фигура 2.9)



*Фиг. 2.9 – електрически котлон Muhler MHP-155B*

1. Среда за разработка

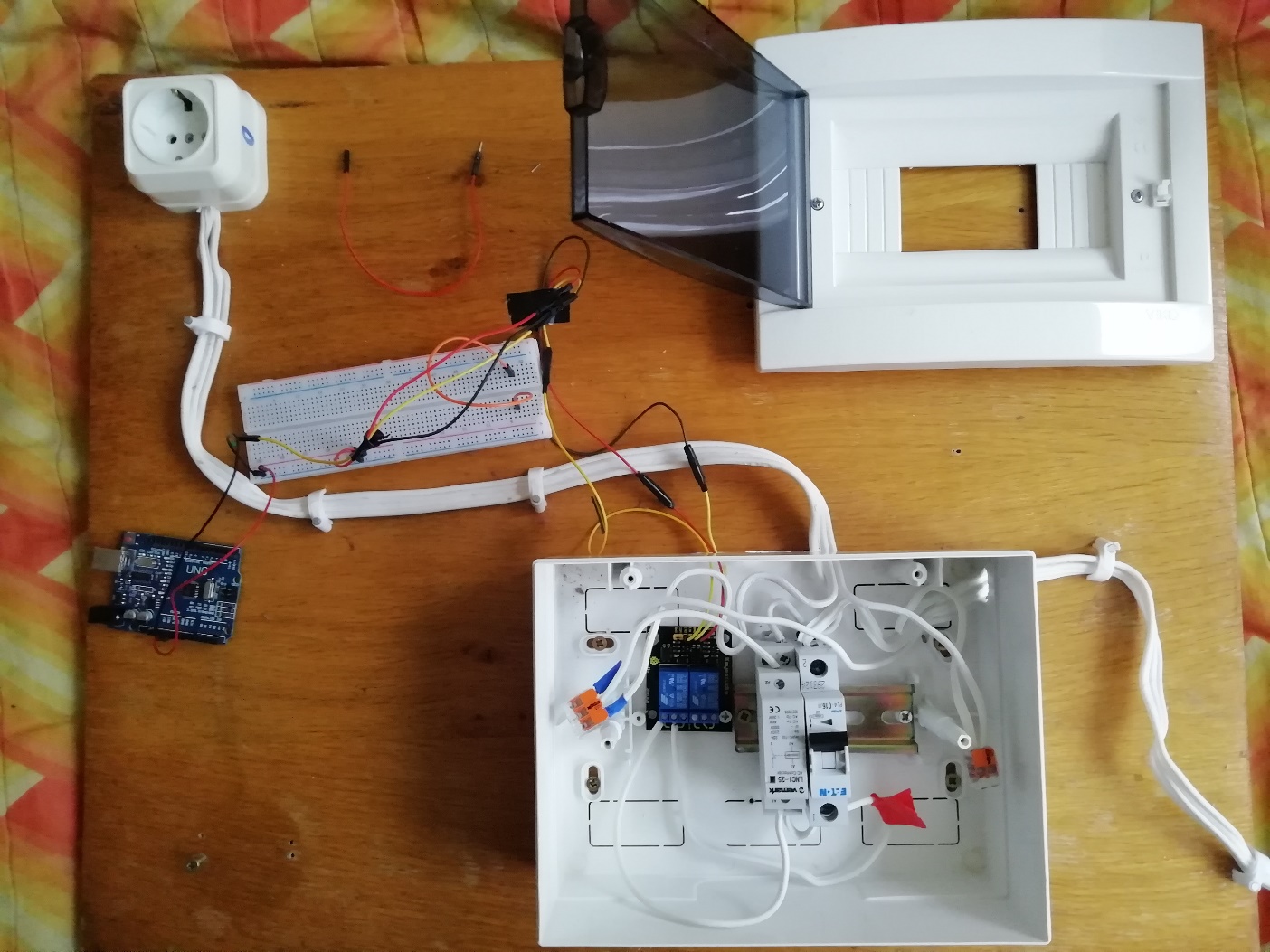
За реализирането на проекта използвам предложеният софтуер на Arduino – Arduino IDE, който ни позволява да създаваме изпълним код, който след това изпращаме в микроконтролерът. Това е текстов редактор, в който има конзола за отпечатване на съобщения и лента с инструменти за по-лесен достъп на функции, които софтуерът предлага. Приложението е написано на Java, но основен език за разработка е комбинация от C и C++ като разработчиците са добавили и няколко вградени метода за по-лесена работа с платката.



*Фиг. 2.10 Arduino IDE*

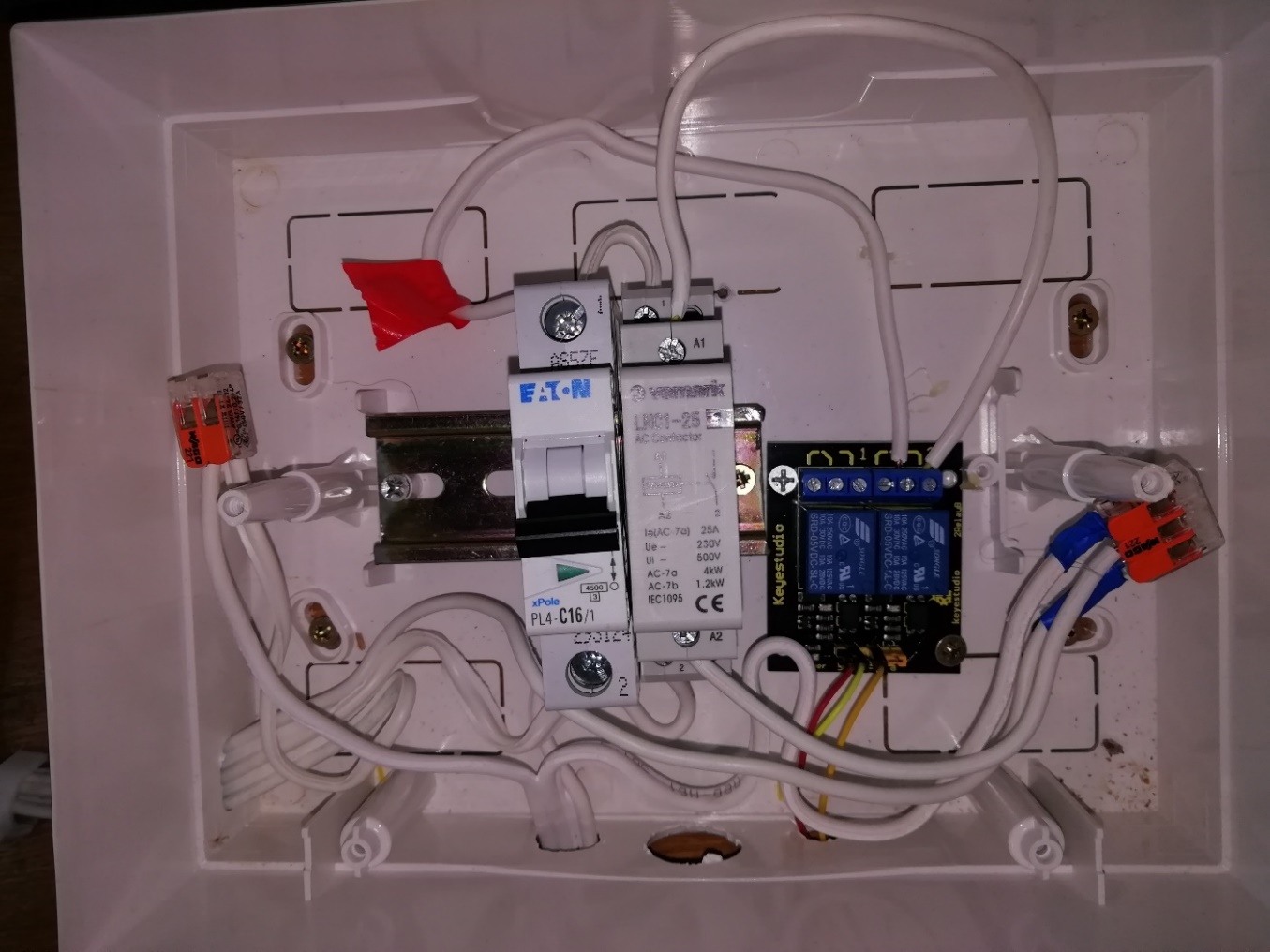
Използвам също така библиотека за по-лесна работа с усилвателят на термо сензора(max 6675.h). Библиотеката предлага функции, които директно изчисляват върнатите резултати от усилвателя и ги превръщат в градусната мярка, с която желаем да работим( целзии или фаренхайт).

# Глава 3 Разработка и реализация

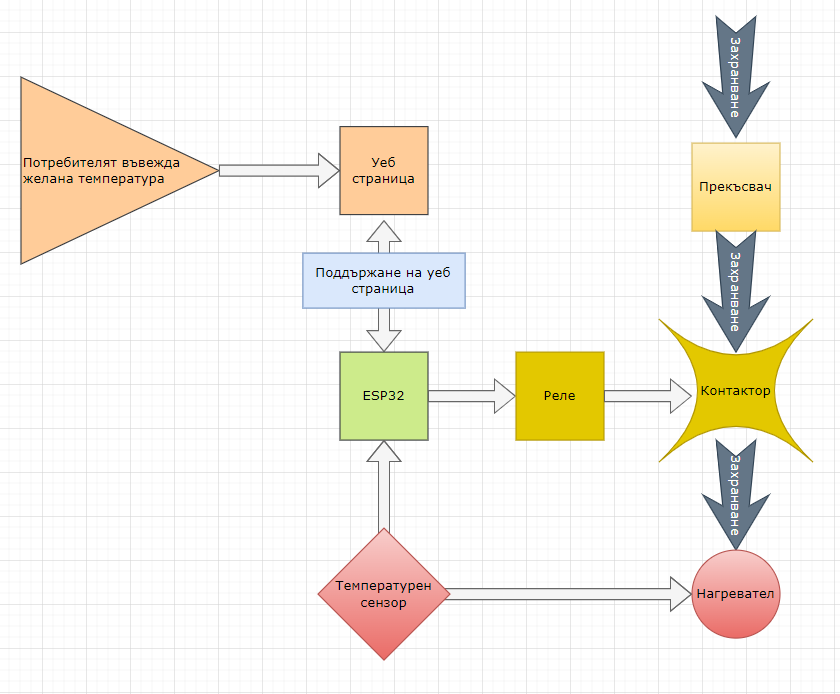
За реализирането на проекта е нужна система която да спира/пуска захранването на термонагревателя. За системата са използвани: автоматичен прекъсвач, електрически контактор, реле(за да задейства/изключва контактора). 

*Фиг. 3.1 – Система за управление на захранването*

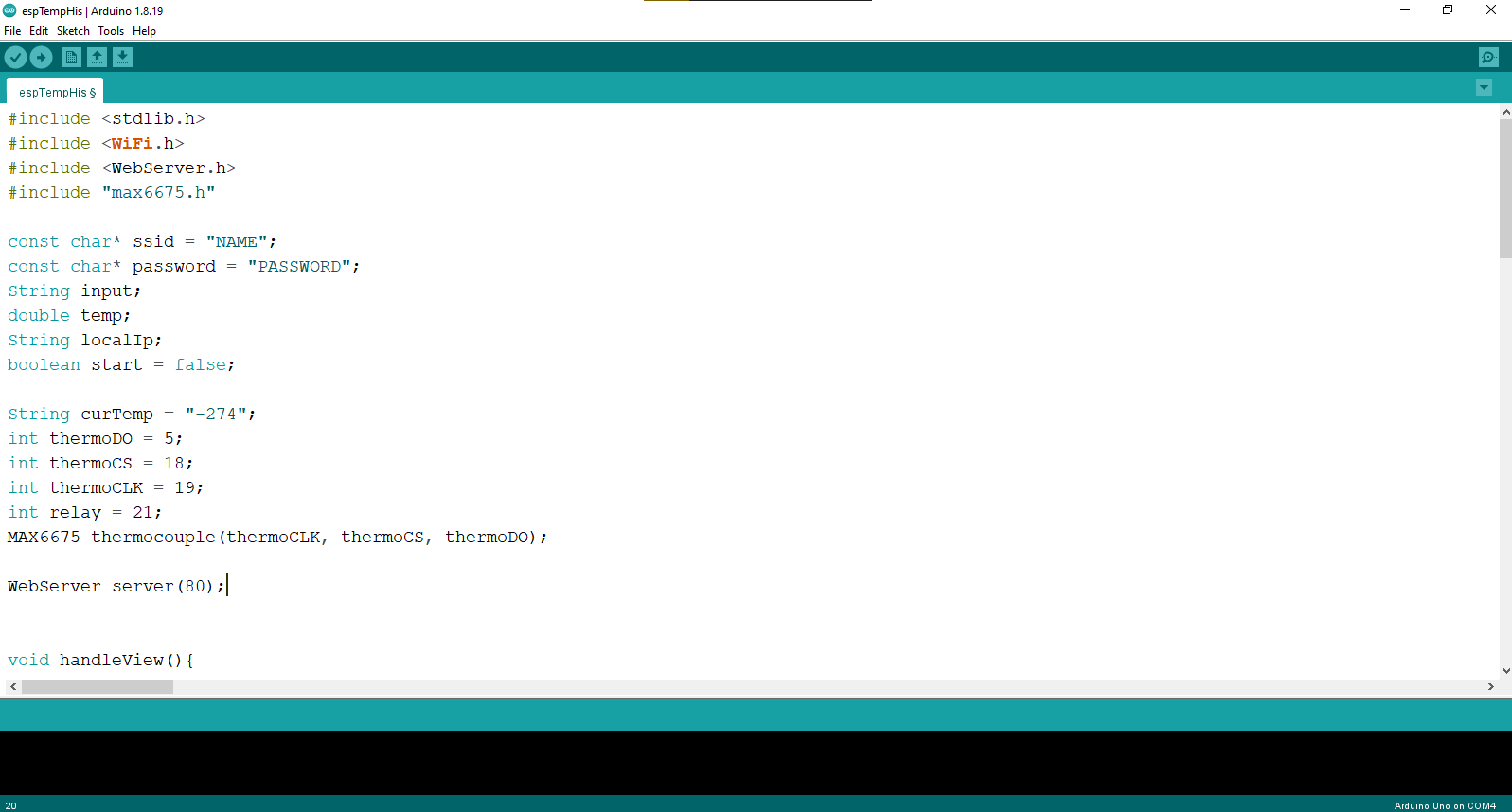
Захранването минава през автоматичният прекъсвач, след това преминава през контактора, който се управлява от реле, което чака сигнал от ESP32 микроконтролера. При получаване на сигнал релето се задейства, в следствие ,на което се пуска контакторът, включвайки системата.



*Фиг. 3.2 – Разпределение на кабелите*

  
*Фиг. 3.3 – Блокова схема на система за управление на захранването*

След реализирането на системата за управление е нужно да добавим ESP32 микроконтролера. Необходим ни е потребителски интерфейс създаден на уеб сървър, в който потребителят да въвежда желаната температура след това да обработи информацията, която е въведена и да провери дали температурният сензор е достигнал въведаната температура.



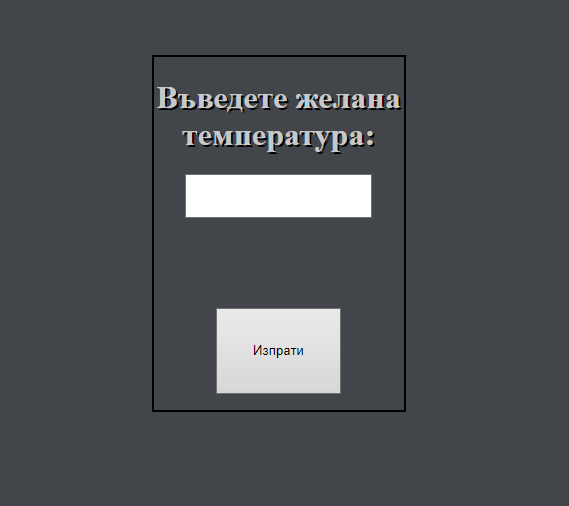
*Фиг. 3.3 – Включване на библиотеки и създаване на глобални променливи*

Кодът, реализиран на ESP32 микроконтролерът, започва с включването на библиотеките: stdlib.h, Wifi.h, WebServer.h и max6675.h . Тези библиотеки се използват съответно за стандартни входни и изходни операции, WiFi комуникация, комуникация с уеб сървър и комуникация със сензор за термодвойка.

След това се декларират глобалните променливи. Променливите “ssid” и “password” служат за запазването на името и паролата на Wi-Fi мрежата, в която ще създадем локалната мрежа по-късно. След това променливата “input”, която служи за запазване на въведената стойност от потребителя запазена под формата на текст, “temp” – желаната температура запазена под формата на десетично число, “localIp” – променлива, която да запазва адресът на локалната мрежа, булева променлива “start” за определяне на състоянието на системата (включена/изключена), “curTemp” – променлива за запазване на отчетената температура на сензора, следващите три променливи (“thermoDO”, “thermoCS”, “thermoCLK”) пазят номерата на пиновете които отговарят на тях в ESP32 (те ще извършват комуникацията с термосензорът) и “relay” , която също отговаря на номер на дигитален пин в микроконтролерът (ще изпраща сигнал до релето за пускане или спиране на системата). 

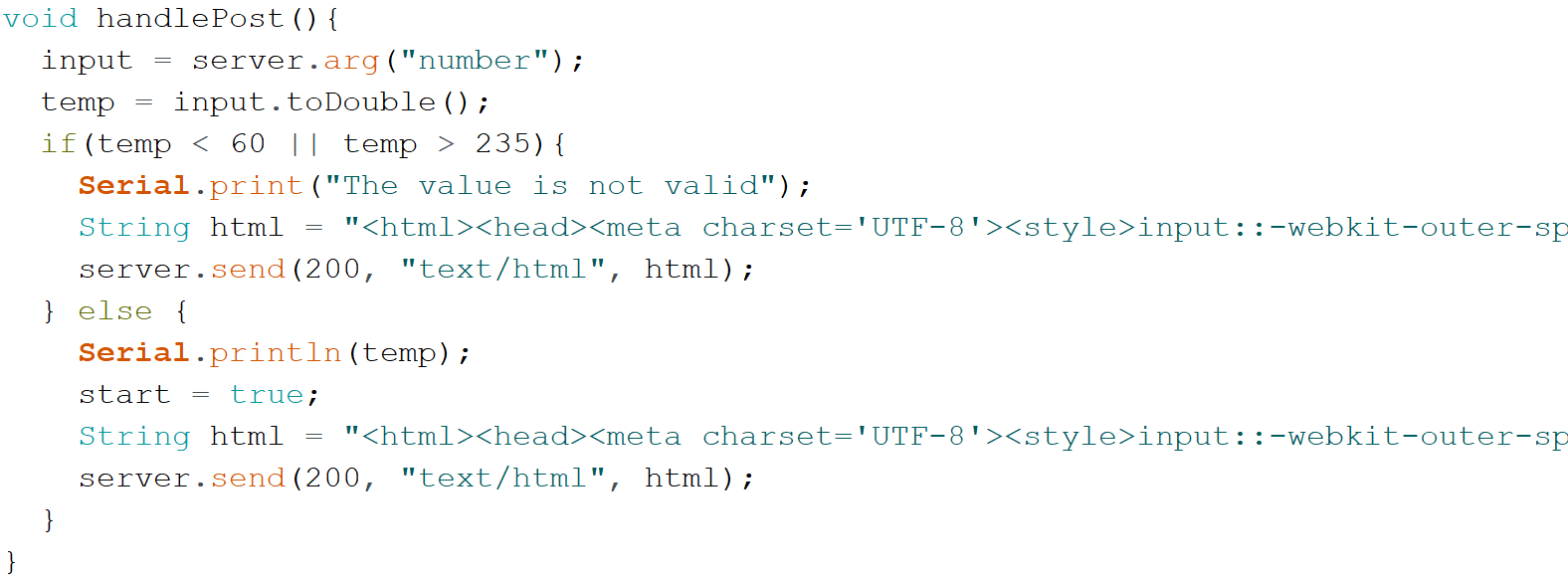
*Фиг. 3.4 – Функция “handleView”*

Функцията “handleView” е първата се изпълняваща фунция при зареждането на уеб страницата. В нея имаме променлива, в която сме запазили кода на уеб страницата. След това чрез “server.send” функцията, изпращаме под формата на HTML код променливата. Комуникацията между устройствата се извършва благодарение на HTTP мрежови протокол, затова въвеждаме в “server.send” функцията: 200(изходният код на HTTP протоколът), “text/html”(текст, който да бъде прочетен като HTML код от клиентският браузър) и променливата “html”.



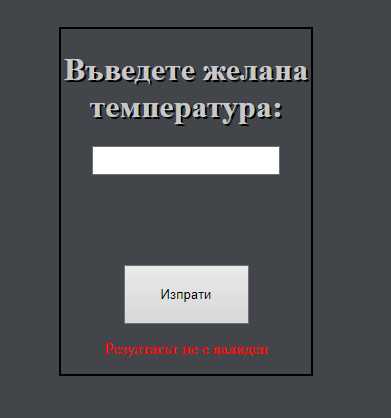
*Фиг. 3.5 – Генерираната HTML страница от функцията “handleView”*

В получената уеб страница имаме поле за попълване, в което можем да въвеждаме само числа в диапазон от 60 до 235 включително в противен случай излиза съобщението: “Моля въведете число между 60 до 235 включително”.



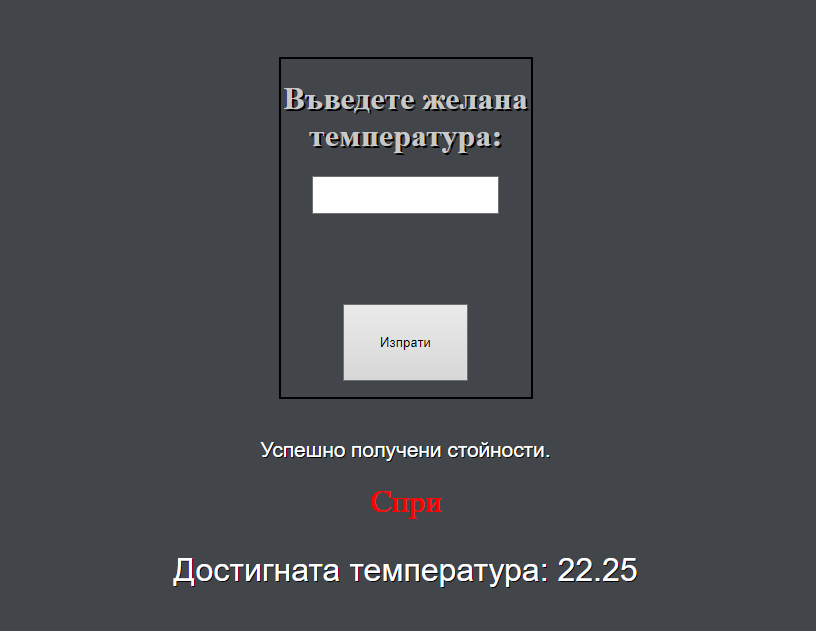
*Фиг. 3.6 – Функция “handlePost”*

След успешно изпращане на желаната температура, информацията попълнена в полето се запазва в глобалната променливата “input”. След това я превръщаме в десетично число и я запазваме в “temp” където след това правим повторна проверка, за да се усигури сигурност.



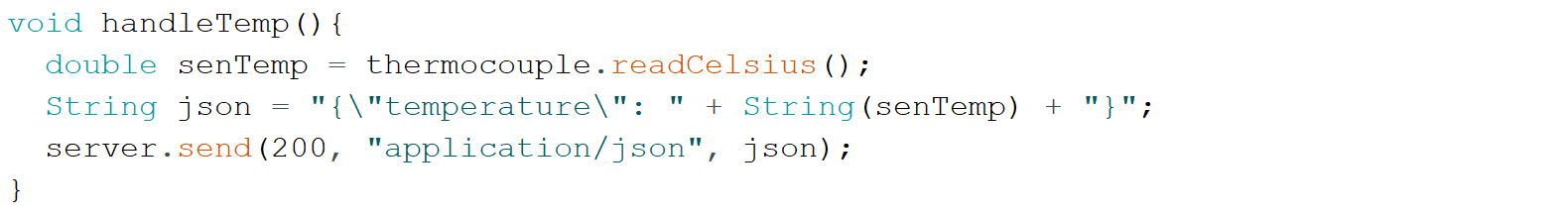
*Фиг. 3.7 – Генерирана страница в случай на грешка*

Ако резултатът се окаже невалиден се генерира страница, съобщавайки ни за грешката. В случай че нямаме грешки във въведената информация булевата променлива “start” променя стойноста си на “true” и се генерира страницата в която се показва температурата на нагревателят в реално време.



*Фиг. 3.8 – Генерирана страница при успешно въведени стойности*

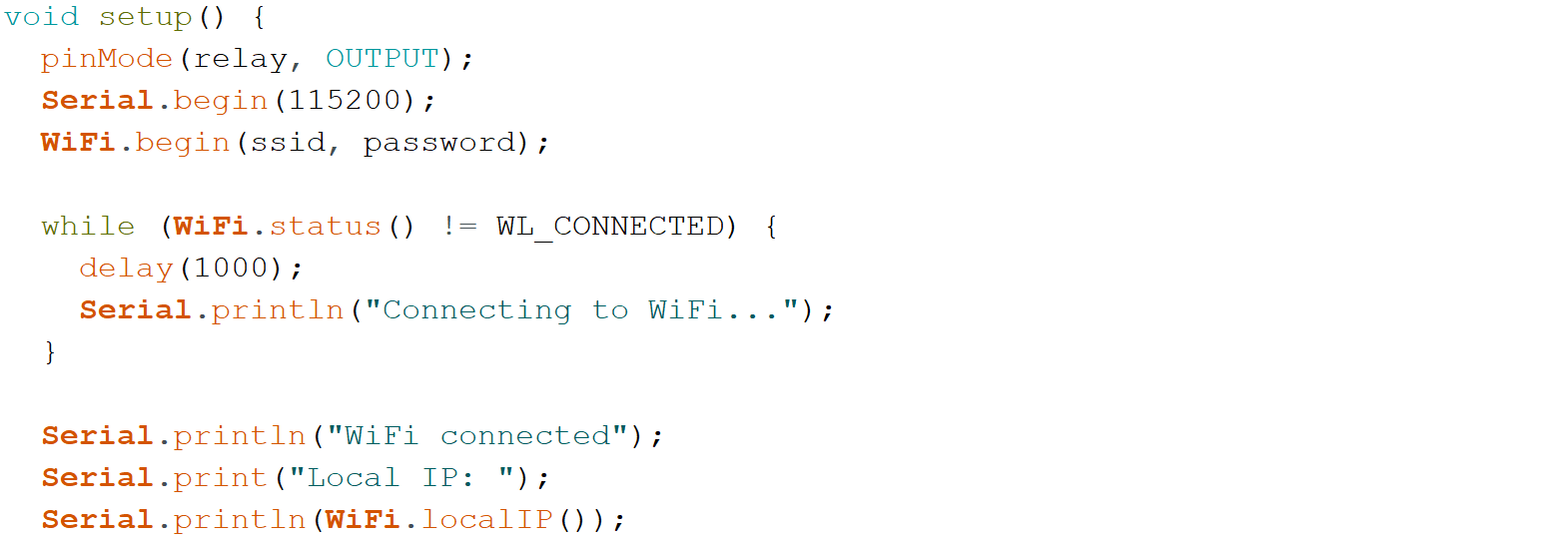
В получената нова страница забелязваме съобщение за получени стойности след това линк, на когото пише “Спри” (при натискане на линка се връщаме в първоначалната страница) и температурата, която се променя на всеки 750 милисекунди. Отчитането на температурата в реално време е реализирано чрез API връзка и комуникация с JSON обект, но ще разгледаме повече в следващата функция.



*Фиг. 3.9 – Функция “handleTemp”*

За да можем да променяме стойността в страницата на всеки 750 милисекунди ще ни е необходим JavaScript поради неговите предимства спрямо HTML, понеже при промяна на стойност в страницата, JavaScript не изисква презареждане на генерираната страница.

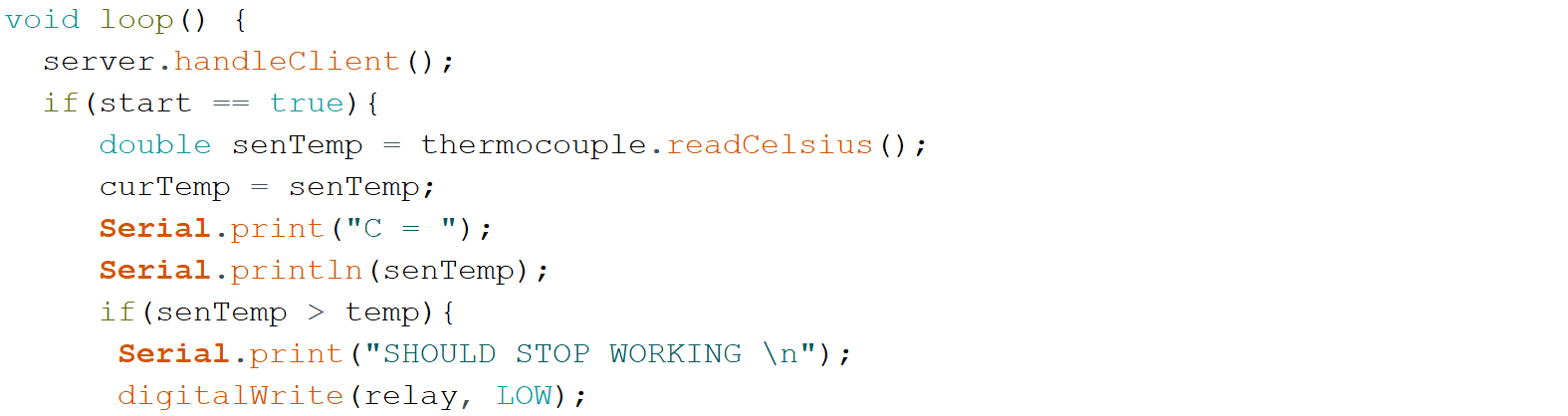
Във функцията “handleTemp” създаваме променлива в която да съхраним настоящата отчетена температура от температурният сензор. След това конкатенираме променливата в текстовият формат на JSON обект, полученият резултат запазваме в променливата “json”. Изпращаме чрез HTTP протоколът информацията, под формата на JSON обект, когато получи заявка (на всеки 750 милисекунди както е заявено в “html” променливата във функцията “handlePost”).

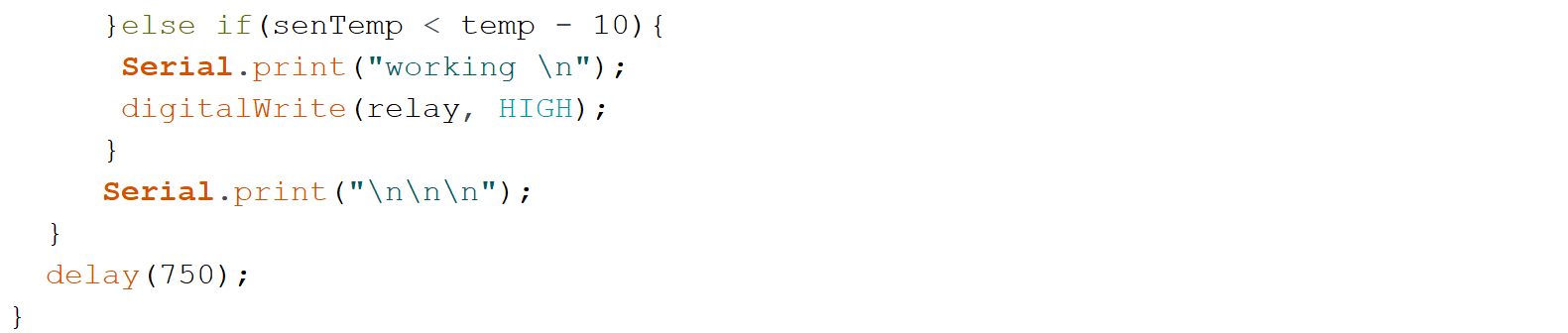




*Фиг. 3.10 – Функция “setup”*

Функцията “setup” е първата се изпълняваща функция при стартирането на ESP32 микроконтролерът. В нея се конфигурират пиновете на които ще работи релето, връзката към Wi-Fi мрежата и подготвянето на уеб страницата. При успешно свързване към Wi-Fi мрежата се генерира, локален за мрежата, адрес, на който могат да се свържат всички устройства свързани към Wi-Fi мрежата. Достъпването до адреса стартира уеб страницата, която се поддържа от HTTP протоколите - GET и POST заявките, които са насочени към съответните им функции.





*Фиг. 3.11 – Функция “loop”*

Функцията “loop” е функция, която се изпълнява безкрайно. В нея се стартират и поддържат комуникацията с уеб страниците. След това се прави проверка на глобалната булева променлива “start”. Ако тя е “true” ESP32 изпраща сигнал на релето да се активира, което в следствие задейства захранващата система, нагрявайки нашият нагревател. Когато температурата надвиши желаните стойности, контролерът изпраща сигнал на релето да спре електрическата веригата. Когато температурата спадне под границата (минималната стойност на границата е разликата на желаната темпертура с десет) релето се пуска отново, което започва нагряването на нагревателят отново.

# Заключение

Системата работи отлично, но има доста място за подобрение. Следващата стъпка е системата да премине на управление чрез симистори (TRIAC управление). Това може да ни спести много пространство и също така да намали общата цена необходима за реализацията на проекта. Това ни позволява да променим хистерезисното управление на PID управление. PID алгоритъмът може да ни позволи прецизно спазване на температурата в рамките на желаните стойности. Уеб страницата също може да бъде подобрена визуално. Също така смятам да добавя настройки за промяна на максималната температура в случай че потребителят използва нагревател, който може да достигне повече от 235 градуса.

## Използвана литература

(1 страница)

Книга, статия, документация и линк.

<https://bg.jf-parede.pt/temperature-sensors-types> - типове температурни сензори.

<https://roboticsbackend.com/the-arduino-language-in-10-points/#:~:text=The%20Arduino%20language%20is%20a,the%20microcontroller%20inside%20the%20board> - десет важни точки за ардуино езикът.

<https://www.digikey.com/en/maker/blogs/2020/the-basics-of-c-on-an-arduino-part-1-variables> - помощен материал за средата за разработка.

<https://encyclopedia.pub/entry/30906#:~:text=The%20Arduino%20project%20began%20in,environment%20using%20sensors%20and%20actuators> - история на ардуино.

<https://circuitdigest.com/article/different-types-of-arduino-boards#:~:text=Arduino%20Nano%20is%20a%20small,the%20USB-TTL%20converter%20chip> - различните видове платки на Ардуино

## Списък с използваните означения и съкращения

(1 страница)

PID controller – Proportional Integral Derivative controller / Пропорционален Интегрално Деференциален конторолер

SISO – Single Input Single Output/Един вход един изход

RTD – Resistance Temperature Detector / Резистен температурен сензор

SRAM – Static Random Access Memory / Статична памет с произволен достъп

EEPROM – Electrically Erasable Programmable Read-only Memory / Електрически изтриваема програмируема памет само за четене

IoT – Internet of Things / Интернет на нещата

TCP – Trasmission Control Protocol / Протокол за управление на предаването

UDP – User Data Protocol / Протокол за потребителски данни

HTTP – Hypertext Transfer Protocol / Протокол за трансфер на хипертекст

HTTPS – Hypertext Transfer Protocol Secure / Сигурен протокол за трансфер на хипертекст

MQTT – Message Queuing Telemetry Transport / Телементричен транспорт на опашка за съобщения

IDE – Integrated Development Environment / Интегрирана среда за разработка

API – Aplication Programming Interface / Интерфейс за програмиране на приложения

JSON – JavaScript Object Notation / JavaScript обектна нотация