

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ АЕРОНАВІГАЦІЇ, ЕЛЕКТРОНІКИ ТА ТЕЛЕКОМУНІКАЦІЙ
КАФЕДРА ЕЛЕКТРОНІКИ, РОБОТОТЕХНІКИ І ТЕХНОЛОГІЙ
МОНІТОРИНГУ ТА ІНТЕРНЕТУ РЕЧЕЙ

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач випускової кафедри
_____ Шутко В.М.
« ____ » _____ 2022 р.

КВАЛІФІКАЦІЙНА РОБОТА

ЗДОБУВАЧА ОСВІТНЬОГО СТУПЕНЯ МАГІСТРА
ЗІ СПЕЦІАЛЬНОСТІ 153 «МІКРО- ТА НАНОСИСТЕМНА ТЕХНІКА»
ОСВІТНЬО-ПРОФЕСІЙНОЇ ПРОГРАМИ
«ФІЗИЧНА ТА БІОМЕДИЧНА ЕЛЕКТРОНІКА»

Тема: «Цифровий датчик температури навколишнього середовища»

Виконавець
студент групи ФЕ-237М _____ Жекало Назар Тарасович

Керівник
д.ф.-м.н., професор _____ Азнакаєв Емір Ганєєвич

Консультант розділу
«Охорона праці» _____ Козлітін О.О.

Консультант розділу
«Охорона навколишнього
середовища» _____ Радомська М.М.

Нормоконтролер _____ Сініцин Р.Б.

КИЇВ 2022

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет Аеронавігації, електроніки та телекомунікацій

Кафедра електроніки, робототехніки і технологій моніторингу та інтернету речей

Напрямок (спеціальність) 153 «Фізична та біомедична електроніка»
(шифр, найменування)

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Шутко В.М.

« ____ » _____ 2022 р.

ЗАВДАННЯ

на виконання дипломного проекту (роботи) студента

Жекало Назар Тарасович

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи): Цифровий датчик температури навколишнього середовища
2. Термін виконання проекту (роботи):
3. Вихідні дані до проекту (роботи): дипломна робота складається зі вступу, п'яти розділів, висновків та списку використаних джерел. Список використаних джерел налічує 34 найменувань. Загальний обсяг роботи – 111 сторінок.
4. Зміст пояснювальної записки (перелік питань, що підлягають обробці): Вступ. Розділ 1 Теоретичні основи. Розділ 2 Теплопровідність та інструкції з проектування для вимірювання температури повітря. Розділ 3 Експериментальне дослідження цифрового датчика температури. Розділ 4 Охорона праці. Розділ 5 Охорона навколишнього середовища. Висновки. Список використаних джерел.
5. Перелік обов'язкового графічного матеріалу: схеми, рисунки, таблиці.

6. Календарний план-графік

№ пор.	Завдання	Термін виконання	Підпис керівника
1	Розробити деталізований зміст розділів диплому		
2	Опрацювання літературних матеріалів за обраною тематикою		
3	Написання теоретичної частини диплому		
4	Проведення дослідів, написання практичної частини диплому		
5	Написання розділів з Охорони праці та Охорони навколишнього середовища		
6	Написання висновку до дипломної роботи, оформлення пояснювальної записки згідно вимог		
7	Усунення недоліків в пояснювальній записці до дипломної роботи		
8	Підготовка роздавального матеріалу, презентації, тез для доповіді		

7. Консультанти з окремих розділів

Розділ	Консультант (посада, П.І.Б.)	Дата, підпис	
		Завдання видав	Завдання прийняв
Охорона праці			
Охорона навколишнього середовища			

8. Дата видачі завдання: «25» жовтня 2021 р.

Керівник дипломної роботи

(підпис керівника)

_____ (П.І.Б.)

Завдання прийняла до виконання

(підпис випускника)

_____ (П.І.Б.)

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ	10
1.1. Історичні передумови	11
1.1.1. Спостереження за розширенням: рідини та біметали	11
1.1.2. Термоелектричні ефекти	12
1.1.3. Теплове випромінювання	13
1.1.4. Температурні шкали	14
1.2. Типи датчиків температури	14
1.2.1. Термопара	15
1.2.2. Резистивний температурний детектор RTD	19
1.2.3. Термістори	21
1.2.4. Термостат	22
1.2.5. Напівпровідники	23
1.2.6. Цифрові датчики температури	25
1.3. Порівняння різних типів датчиків температури	25
РОЗДІЛ 2 ТЕПЛОПРОВІДНІСТЬ ТА ІНСТРУКЦІЇ З ПРОЕКТУВАННЯ ДЛЯ ВИМІРЮВАННЯ ТЕМПЕРАТУРИ ПОВІТРЯ	32
2.1. Теплопровідність	32
2.2. Визначення домінуючого шляху теплопровідності	34
2.2.1 Безсвинцеві корпуси CSP, DSBGA	35
2.2.2 Безвивідні корпуси QFN, DFN	36
2.2.3 Безвивідні корпуси без підкладки для кріплення матриці DFN ..	37
2.2.4 Корпуси MSOP, SOIC, VSSOP, SOT-563, SOT-23	37

2.3. Визначення теплопровідності через друковану плату	38
2.3.1 Загальне рівняння теплопровідності	39
2.3.2 Поздовжня теплопровідність	39
2.3.3 Приклад: визначення домінуючого поздовжнього шляху теплопровідності	40
2.3.4 Перпендикулярна теплопровідність	41
2.3.5 Теплопровідність через отвір	42
2.3.6 Приклад: визначення основного перпендикулярного шляху теплопровідності	43
2.4. Рекомендації з проектування для вимірювання температури повітря	44
2.4.1. Розгляд заземлення	45
2.4.2. Розбиття друкованої плати на розділи	47
2.4.3. Острів ізоляції	47
2.4.4. Перфорація	48
2.4.5. Крайовий з'єднувач	49
2.4.6. Контроль теплової маси друкованої плати	49
РОЗДІЛ 3 ДОСЛІДЖЕННЯ ЦИФРОВОГО ДАТЧИКА ТЕМПЕРАТУРИ	51
3.1. DS18B20 цифровий вимірювач температури	51
3.1.1. Особливості DS18B20	51
3.1.2. Режим – передача сигналу тривоги	55
3.1.3. Живлення термодатчика DS18B20	56
3.1.4. 64-розрядний серійний код пристрою	58
3.1.5. Пам'ять датчика	59
3.1.6. Регістр конфігурації термодатчика	60
3.1.7. Генерація циклічного коду (CRC)	61

3.1.8. Однопровідний інтерфейс 1-Wire	62
3.1.9. Гранично допустимі параметри DS18B20	71
3.2. Моделювання та дослідження датчика DS18B20	74
3.3. Програмне забезпечення	78
РОЗДІЛ 4 ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА	81
4.1. Директива щодо обмеження небезпечних речовин	81
4.2. Приклади компонентів продукту, що містять заборонені речовини	82
4.3. Свинець в електроніці	83
РОЗДІЛ 5 ОХОРОНА ПРАЦІ	85
5.1. Аналіз шкідливих та небезпечних факторів	85
5.2. Розробка заходів з охорони праці	88
5.3. Пожежна безпека приміщення	89
5.4. Інструкція з охорони праці інженера-програміста	91
ВИСНОВКИ	93
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	96
Додаток А	99

РЕФЕРАТ

Пояснювальна записка до дипломної роботи «Цифровий датчик температури навколишнього середовища» містить: 111 сторінок, 54 рисунка, 34 використаних джерел.

ДАТЧИК, СЕНСОР, ТЕМПЕРАТУРА.

Об'єкт дослідження – цифровий датчик температури.

Мета дипломної роботи – розробити та реалізувати алгоритм керування датчиком температури DS18B20.

ВСТУП

Температурні параметри є важливою проблемою в кожному аспекті нашого особистого життя, професійного життя та навіть пристроїв, які ми використовуємо для нашого повсякденного існування. Більше того, у галузях науки, електроніки та комп'ютерних технологій у цілому проекти будуються навколо тих самих температурних проблем. Більш конкретно, дві температурні класифікації вимагають особливої уваги при розробці електронних пристроїв і використанні електронних компонентів, це температура навколишнього середовища та температура з'єднання або робоча температура.

Температура навколишнього середовища означає температуру навколишнього повітря або середовища, в якому знаходиться пристрій, коли пристрій увімкнено. Крім того, визначення температури навколишнього середовища є вимірюванням компонентів або самого обладнання, а також його середовища. Це вимірювання є дуже важливим для роботи пристрою, продуктивності та життєвого циклу.

Енергоємні електронні компоненти, такі як мікросхеми процесорів, програмовані вентильні матриці (FPGA), спеціальні інтегральні схеми (ASIC), а також силові мікросхеми нагріваються під час роботи. Коли систему ввімкнено, тепло, що виділяється цими мікросхемами, передається об'єктам з нижчою температурою поблизу. Вимірювання температури навколишнього повітря за допомогою пристрою для поверхневого монтажу може бути складним, оскільки передача тепла від компонентів на тій самій друкованій платі може впливати на показання температури навколишнього повітря датчиком і перешкоджати цьому. Щоб підтримувати точність у програмах, які вимагають вимірювання температури навколишнього повітря, важливо дотримуватися хороших методів компонування, таких як розуміння домінуючого теплового шляху, ізоляція острова, оточеного упаковкою, і тримання пристрою якомога далі від джерел тепла, що заважають. Ця примітка щодо застосування зосереджуватиметься на стратегіях

компонування для подолання проблем вимірювання температури поза бортом (навколишнього повітря), що охоплює багато застосувань.

Існує три способи передачі тепла: теплопровідність через тверді тіла, конвекція тепла через рідини та тепло, що виділяється випромінюванням. У цьому звіті зосереджено увагу на теплопровідності, оскільки воно домінує в теплопередачі в друкованих платах і, отже, має найбільше значення для вимірювання температури. Тепло передається від сусідніх компонентів друкованої плати до матриці датчика через саму друковану плату та через корпус датчика.

Розробнику системи доступно кілька різних технологій вимірювання температури. Правильна технологія залежить від цільової температури, яку потрібно виміряти, а також від інших системних вимог, таких як вартість, розмір контуру та час проектування.

РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ

Температура – це величина навколишнього середовища, яка просто вимірюється. Це зазвичай, тому що на більшість хімічних, механічних, електричних, фізичних і біологічних пристроїв впливає температура. Спеціальні біологічні процедури, хімічні реакції та навіть електронні системи найкраще працюють у обмеженому діапазоні температур. Температура є однією з найбільш часто вимірюваних характеристик, тому існує багато методів її визначення. Вимірюванням температури можна керувати за допомогою прямого контакту з джерелом поблизу або без прямого контакту з джерелом, використовуючи випромінювану енергію.

Датчик температури - це прилад, який використовується для вимірювання тепла або температури робочої частини машин. Існують різні типи датчиків температури. Датчик температури, такий як електричні термометри, включає в себе спеціальну броню і блок з одного або декількох елементів, який може містити, наприклад, шийну трубку, з'єднувальну головку, рукоятку і термогільзу. Частина датчика, яка встановлена в датчик температури, вимірює фактичну температуру та перетворює її на електричний сигнал. Вимірюванням температури керують різні прилади. Типовими прикладами датчиків температури є, як правило, термопара та RTD (датчик температури опору), які виконували вимірювання температури за допомогою методу електричного сигналу.

Сьогодні на ринку існує багато різних датчиків температури, таких як термопари, RTD, напівпровідникові датчики, термістори, інфрачервоні датчики тощо. Ці датчики температури відрізняються від простих термостатичних систем ON/OFF, які контролюють систему домашнього опалення, до надзвичайно чутливих типів, таких як напівпровідники. що може контролювати особливий порядок контролю пічних установок.

Рух молекул і атомів породжує кінетичну енергію у формі тепла. Чим більше рух, тим більше тепла виробляється. Датчик температури може вимірювати кількість теплової енергії, виробленої системою чи об'єктом,

даючи нам дозвіл ідентифікувати чи відчувати будь-які фізичні зміни генерації температури за допомогою аналогового чи цифрового виходу.

1.1. Історичні передумови

Відчуття гарячого та холодного є фундаментальними для людського досвіду, однак пошук способів вимірювання температури став викликом для багатьох великих умів. Незрозуміло, чи мали стародавні греки чи китайці способи вимірювання температури, тому, наскільки нам відомо, історія температурних датчиків почалася в епоху Відродження.

Тепло – це міра енергії в тілі або матеріалі – чим більше енергії, тим воно гарячіше. Але на відміну від фізичних властивостей маси та довжини, це було важко виміряти. Більшість методів були непрямими, спостерігаючи вплив тепла на щось і виводячи температуру з цього.

Створення шкали вимірювань також було складним завданням. У 1664 році Роберт Гук запропонував використовувати точку замерзання води як нульову точку, за якою вимірюють температуру. Приблизно в той самий час Оле Ремер побачив потребу у двох фіксованих точках, що дозволяло інтерполяцію між ними. Точками, які він вибрав, були температура замерзання Гука, а також температура кипіння води. Це, звичайно, залишає відкритим питання про те, наскільки гарячими чи холодними можуть бути речі.

На це відповіли Гей-Люссак та інші вчені, які працювали над газовими законами. У 19 столітті, досліджуючи вплив температури на газ при постійному тиску, вони помітили, що об'єм збільшується на частку $1/267$ на градус Цельсія (пізніше переглянуто до $1/273,15$). Це призвело до концепції абсолютного нуля при мінус $273,15^{\circ}\text{C}$.

1.1.1. Спостереження за розширенням: рідини та біметали

Повідомляється, що Галілей побудував пристрій, який показував зміни температури десь приблизно в 1592 році. Він використовував стиснення повітря в посудині, щоб набрати стовп води, висота стовпа вказує на ступінь

охолодження. Однак на це сильно вплинув атмосферний тиск і було не більше ніж новинкою.

Термометр, яким ми його знаємо, був винайдений у 1612 році на території сучасної Італії Санторіо Санторі. Він замкнув рідину всередині скляної трубки, спостерігаючи, як вона рухалася вгору по трубці, розширюючись. Шкала на трубці полегшувала перегляд змін, але системі бракувало точних одиниць.

З Ремером працював Даніель Габріель Фаренгейт. Він почав виготовляти термометри, використовуючи спирт і ртуть як рідину. Ідеальним є ртуть, оскільки вона має дуже лінійну реакцію на зміну температури в широкому діапазоні, але занепокоєння щодо токсичності призвело до скорочення використання. Зараз для його заміни розроблено інші рідини. Рідинні термометри все ще широко використовуються, хоча важливо контролювати глибину занурення колби. Використання термогілзи сприяє хорошему теплообміну.

Біметалічний датчик температури був винайдений наприкінці 19 століття. Це дає перевагу диференціального розширення двох скріплених разом металевих смуг. Зміни температури створюють вигин, який можна використовувати для активації термостата або датчика, подібного до тих, що використовуються в газових грилях.

1.1.2. Термоелектричні ефекти

На початку 19 століття електрика була захоплюючою сферою наукових досліджень, і незабаром вчені виявили, що метали відрізняються за своїм опором і провідністю. У 1821 році Томас Йоганн Зеєбек виявив, що напруга створюється, коли кінці різновидних металів з'єднують і поміщають при різних температурах. Пельтьє виявив, що цей ефект термопари оборотний і може використовуватися для охолодження.

У тому ж році Хамфрі Дейві продемонстрував, як питомий електричний опір металу пов'язаний з температурою. Через п'ять років Беккерель запропонував використовувати платино-платинову термопару для

вимірювання температури, але Леопольдо Нобілі створив цей прилад до 1829 року.

Платина також використовується в резистивному температурному детекторі, винайденому в 1932 році Ч. Мейерсом. Він вимірює електричний опір довжини платиного дроту і, як правило, вважається найточнішим типом датчика температури. RTD з використанням дроту за своєю природою крихкі та непридатні для промислового застосування. В останні роки були розроблені плівкові RTD, які менш точні, але більш надійні.

У 20 столітті також були винайдені напівпровідникові прилади для вимірювання температури. Вони реагують на зміни температури з хорошою точністю, але до недавнього часу їм бракувало лінійності.

1.1.3. Теплове випромінювання

Дуже гарячі та розплавлені метали світяться, виділяючи тепло та видиме світло. Вони також випромінюють тепло при нижчих температурах, але при більшій довжині хвилі. Англійський астроном Вільям Гершель був першим, хто визнав, близько 1800 року, що це «темне» або інфрачервоне світло викликає нагрівання. Працюючи зі своїм співвітчизником Меллоні, Нобілі знайшов спосіб виявити цю випромінювану енергію, з'єднавши термопари в серію, щоб зробити термобатарей.

У 1878 році з'явився болометр. Винайдений американцем Семюелем Ленглі, він використовував дві платинові смужки, одна з яких була чорною, у розташуванні мосту Вітстона. Нагрівання інфрачервоним випромінюванням викликало вимірну зміну опору.

Болометри чутливі до інфрачервоного випромінювання в широкому діапазоні довжин хвиль. Навпаки, пристрої типу детекторів фотонів, розроблені з 1940-х років, як правило, реагують лише на інфрачервоне випромінювання в обмеженому діапазоні хвиль. Детектори сульфїду свинцю чутливі до довжин хвиль до 3 мікрон, а відкриття потрійного сплаву HgCdTe у 1959 році відкрило двері для детекторів, адаптованих до певних довжин хвиль.

Сьогодні широко використовуються недорогі інфрачервоні пірометри, а теплові камери знаходять все більше застосувань у міру зниження ціни.

1.1.4. Температурні шкали

Коли Фаренгейт виготовляв термометри, він зрозумів, що йому потрібна температурна шкала. Він встановив температуру замерзання солоної води на 30 градусів, а її температуру кипіння на 180 градусів вище. Згодом було вирішено використовувати чисту воду, яка замерзає при трохи вищій температурі, тобто замерзає при 32°F і кипить при 212°F.

Через чверть століття Андерс Цельсій запропонував шкалу від 0 до 100, яка сьогодні носить його ім'я. Пізніше, побачивши переваги фіксованої точки на одному кінці шкали, Вільям Томсон, пізніше лорд Кельвін, запропонував використовувати абсолютний нуль як початкову точку системи Цельсія. Це призвело до шкали Кельвіна, яка сьогодні використовується в науковій сфері.

Сьогодні шкали вимірювання температури визначені в документі під назвою Міжнародна температурна система 90, або скорочено ITS-90.

1.2. Типи датчиків температури

Вимірювання температури має вирішальне значення для сучасних електронних пристроїв, особливо для дорогих портативних комп'ютерів та інших портативних пристроїв із щільною упаковкою ланцюгів, які розсіюють значну потужність у вигляді тепла. Знання температури системи також можна використовувати для контролю зарядки батареї, а також для запобігання пошкодженню дорогих мікропроцесорів.

Компактне портативне обладнання високої потужності часто має вентилятор для охолодження, щоб підтримувати температуру переходу на належному рівні. Щоб подовжити термін служби батареї, вентилятор повинен працювати лише за необхідності. Точне керування вентилятором вимагає знання критичних температур від відповідного датчика температури.

Попит на вимірювання температури в сучасних передових виробничих середовищах зріс і охоплює майже всі додатки в автомобільній, промисловій, персональній електроніці, комунікаціях і підприємствах. Враховуючи

різноманітність застосувань, які вимагають моніторингу температури, існує багато міркувань при виборі найефективнішого типу датчика, включаючи точність датчика, розмір, розміщення, дрейф і калібрування.

Далі порівнюються чотири найпопулярніші датчики температури:

- Датчики інтегральної схеми (IC). Виготовлені з кремнію та здатні використовувати передбачувані теплові характеристики кремнієвого р-n-переходу, датчики IC забезпечують високу точність, низьке споживання енергії, швидкий час відгуку та компактний форм-фактор.

- Термістори. Існує дві основні категорії термісторів: з негативними температурними коефіцієнтами (NTC) і позитивними температурними коефіцієнтами (PTC). З підвищенням температури опір NTC зменшується, а опір PTC зростає. NTC зазвичай вимагають калібрування для більшої точності, але кремнієві термістори PTC мають лінійну залежність між температурою та опором, тому не потребують калібрування, заощаджуючи ресурси та час у процесі проектування.

- Резистивні датчики температури (RTD). Оскільки RTD виготовлені з чистого металу, наприклад платини, вони можуть працювати з високою точністю в середовищах з надзвичайно високою температурою; недоліком є те, що вони є одними з найдорожчих типів датчиків температури. RTD працюють на відомій кореляції між збільшенням опору чистого металу та підвищенням температури.

- Термопари. Термопара з'єднує два різні метали, наприклад мідь і залізо, щоб утворити спай. Цей перехід виробляє невелику напругу при зміні температури, яку можна конвертувати у відповідне значення температури. Хоча термопари можуть вимірювати найширший діапазон температур серед чотирьох типів датчиків, вони часто більші за розміром і коштують дорожче, ніж IC-датчики або термістори.

1.2.1. Термопара

Термопара – це поширений тип датчиків температури, який виготовлений шляхом поєднання кінців двох різних металів. З'єднувальний кінець

називається «гарячий спай». Інший кінець вводиться як «холодний спай» або «холодний кінець». Холодний спай створюється в останній частині матеріалу термопар. Якщо температура холодного та гарячого спаїв змінюється, утворюється невелика напруга. Ця напруга є ЕРС (електрорушійною силою), її можна виміряти та використати для відображення зміни температури.

Термопара є найпоширенішим типом датчиків температури. Термопары поширені завдяки своїй зручності для користувача, простоті та швидкості реагування на зміни температури. Термопары також мають найвищий рівень температури з усіх датчиків температури, від нижчого ніж -200°C до більш ніж 2000°C .

Термопары існують у різних композиціях металів або калібрувань. Найпоширенішими видами термопар є термопары з недорогоцінних металів; однак термопары з благородних металів використовуються у високотемпературних системах. Кожне калібрування має певний рівень температури та середовище, тоді як максимальна температура змінюється залежно від діаметра дроту, який використовується в термопарі.

Термопары в основному складаються з двох різних металів, таких як константан і мідь, які скріплені разом. Один спай підтримується при постійній температурі, введений як еталонний спай, тоді як інший є вимірювальним. Коли температура двох з'єднань змінюється, на з'єднанні виникає напруга, яка використовується для визначення температури.

Застосування термопар

Термопару можна використовувати в кількох сферах застосування, зокрема:

- Датчики полум'я в машинах безпеки для газових систем.
- Датчики температури в термостатах для побутових приладів.
- Вимірювання температури для промислових процесів, таких як вихлоп газових турбін і дизельних двигунів.

Конструкція термопары

Принцип роботи термопари дуже базовий і простий. Коли з'єднання двох різних металів, таких як мідь і константан, зливається, вони генерують «термоелектричний» результат, який забезпечує кілька мілівольт (мВ), постійну різницю потенціалів між ними. Різниця напруг між ними називається «ефектом Зеебека», оскільки різниця температур утворюється вздовж провідних проводів, створюючи ЕРС. Тоді зовнішня напруга від термопарі є операцією змін температури.

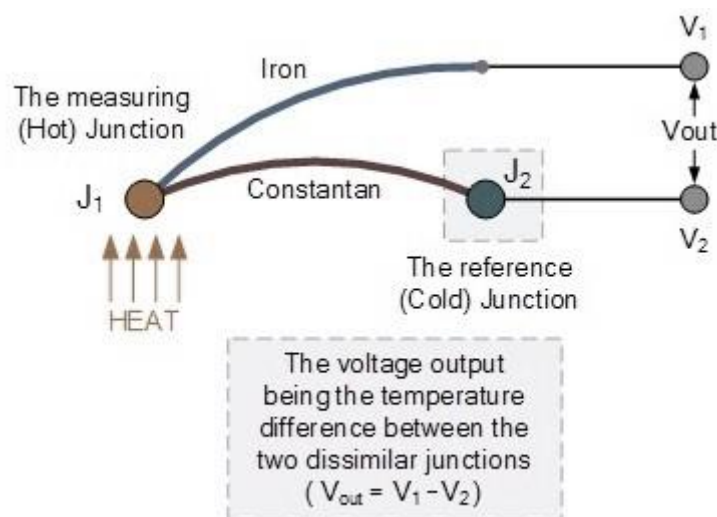


Рис. 1.1. Датчик температури: конструкція термопарі

Якщо обидві частини мають однакову температуру, то різниця потенціалів у переходах дорівнює нулю. Іншими словами, ми не маємо вихідної напруги і $V_1 = V_2$. Тим не менш, коли з'єднання з'єднані ланцюгом і мають різні температури, вихідна напруга буде визначена порівняно з різницею температур між ними, $V_1 - V_2$. Ця зміна напруги зростатиме залежно від температури, доки не буде досягнуто пікової напруги на з'єднаннях. Це визначається властивостями двох різних металів, що використовуються.

Три звичайні речовини термопарі, які використовуються для загальних датчиків температури, це залізо-константан, нікель-хром і мідь-константан. Вихідна напруга термопарі дуже низька, наприклад, лише кілька мілівольт (мВ) для коливання температури на 10°C . Через таку низьку вихідну напругу зазвичай потрібні деякі рішення, наприклад підсилення.

Посилення термопар

Необхідно уважно вибрати тип підсилювача, робочий або дискретний, оскільки необхідна відповідна стабільність, щоб припинити повторне калібрування термопар через повторювані інтервали. Таким чином, підсилювач типу чоппера та інструментального підсилювача є ідеальним варіантом для більшості датчиків температури в різних застосуваннях.

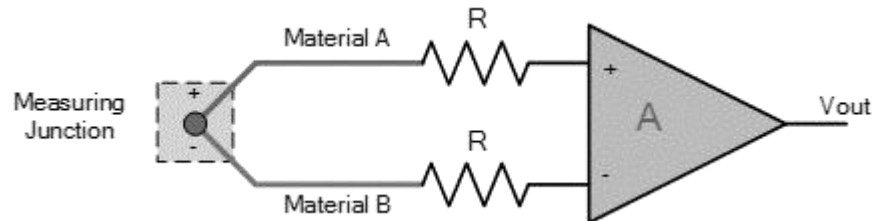


Рис. 1.2. Датчик температури: посилення термопар

Для визначення температури гарячого спаю необхідно знати температуру холодного спаю. Тут точність обмежена тим фактом, що існують дві системи, які мають окремі допуски та можливості, які взаємодіють одна з одною.

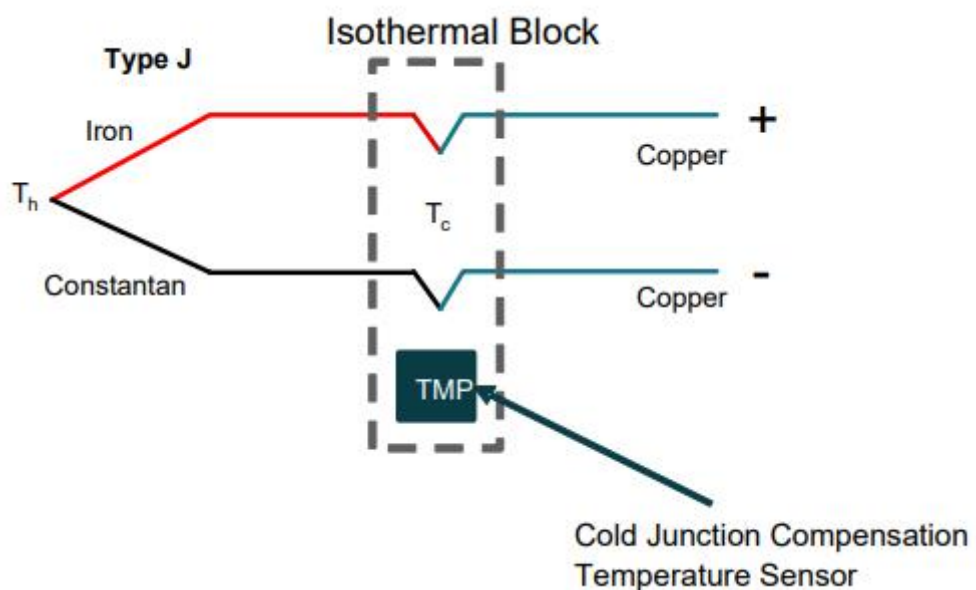


Рис. 1.3. Термопара з датчиком температури CJC

Термопары не потребують зовнішнього збудження, тому на них не впливають проблеми самонагрівання. Вони також можуть витримувати екстремальні температури ($>2000^{\circ}\text{C}$). Хоча вони міцні та недорогі, термопары потребують додаткового датчика температури для компенсації холодного

спаю (СЈС). Вони, як правило, нелінійні та дуже чутливі до паразитних спаїв, де термопара прикріплена до плати.

Нарешті, оцифрування термопарі буде чутливим до раніше обговорених помилок АЦП.

1.2.2. Резистивний температурний детектор RTD

Резистивний температурний детектор або RTD – це датчик температури, опір якого залежить від температури. Зазвичай він виготовляється з платини, тоді як ті, що виготовлені з нікелю або міді, не є чимось незвичайним, RTD можуть бути виготовлені в багатьох різних формах, наприклад, у формі тонкої плівки або дроту. Проблема в тому, що платина занадто дорога, і одним із практичних недоліків використання такого типу інструменту є її вартість. Застосування постійного струму, вимірювання вихідної напруги та визначення опору RTD є простими методами вимірювання температури в RTD.

RTD приблизно відображають лінійні діаграми опору температурі над своїми робочими частинами. Наприклад, на платі оцінки PT100 RTD для вимірювання температури використовується простий RTD (PT100 має стандартний діапазон опору 100 Ом при 0°C). Для вимірювання температури у віддалених ділянках також можна використовувати зовнішній 2 або більше провідний PT100. Величина струму завжди низька, щоб зменшити самонагрівання відповідно до розсіювання потужності. Схема, представлена

на малюнку, є умовою постійного струму, яка використовує підсилювач, опорну напругу та транзистор PNP.

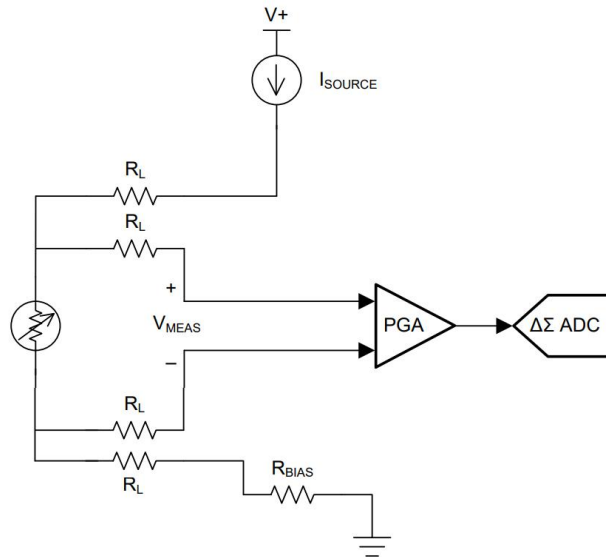


Рис. 1.4. Складна 4-провідна схема RTD

RTD - це точні датчики температури, виготовлені з високочистих провідних металів, подібних до датчиків термістора. Ці інструменти мають тонку плівку платини, нанесену на білу керамічну поверхню. RTD мають додатні температурні коефіцієнти (PTC), але, на відміну від термістора, їх зовнішня напруга надзвичайно лінійна, що забезпечує дуже точні вимірювання між датчиками температури. Тим не менш, вони мають дуже низьку термічну чутливість, тобто зміна температури генерує лише дуже низьку зміну вихідного сигналу, наприклад, $1\Omega/^\circ\text{C}$.

RTD - це пасивні резистивні системи, такі як термістори. Вони пропускають постійний струм у датчик температури, щоб отримати вихідну напругу, яка лінійно зростає з температурою. Звичайний RTD має опір приблизно 100 Ом при 0°C , який підвищується до 140 Ом при 100°C з діапазоном робочих температур від -200 до $+600^\circ\text{C}$.

RTD є резистивним приладом, тому нам потрібно пропускати в них постійний струм і контролювати вихідну напругу. Однак будь-яка зміна опору, відповідно до I^2R (закон Ома), викликає зміну показань. Щоб уникнути цієї проблеми, RTD зазвичай підключають до мережі моста Уїтстона за допомогою додаткових з'єднувальних проводів.

1.2.3. Термістори

Термістор — це датчик температури, опір якого змінюється залежно від температури, подібно до датчиків температури. Однак термістори виготовлені з напівпровідникових речовин. Їхня назва походить від терміну: ТЕРМОчутливий резІСТОР. Термістор має особливий тип резистора, який змінює його фізичний опір більше, ніж інші датчики температури.

Їх опір вказано так само, як і терморезистивні датчики температури, але термістори мають нелінійну діаграму опір-температура. Як наслідок, він може забезпечувати значну варіацію опору для дуже невеликої зміни температури в робочому діапазоні. Це робить його дуже чутливим інструментом, ідеальним для високотехнологічних додатків і додатків із заданими значеннями.

Термістори зазвичай виготовляються з керамічних речовин, таких як оксиди марганцю, нікелю або кобальту, покриті спеціальною поверхнею скла. Їх особливими перевагами перед іншими типами є точність, повторюваність і швидка реакція на зміни температури.

Термістори — це пасивні компоненти, опір яких змінюється залежно від температури. Термістори поділяються на дві категорії: негативний температурний коефіцієнт (NTC) і позитивний температурний коефіцієнт (PTC).

У той час як термістори пропонують різноманітні варіанти упаковки для бортового та зовнішнього вимірювання температури, типова реалізація вимагає більшої кількості системних компонентів.

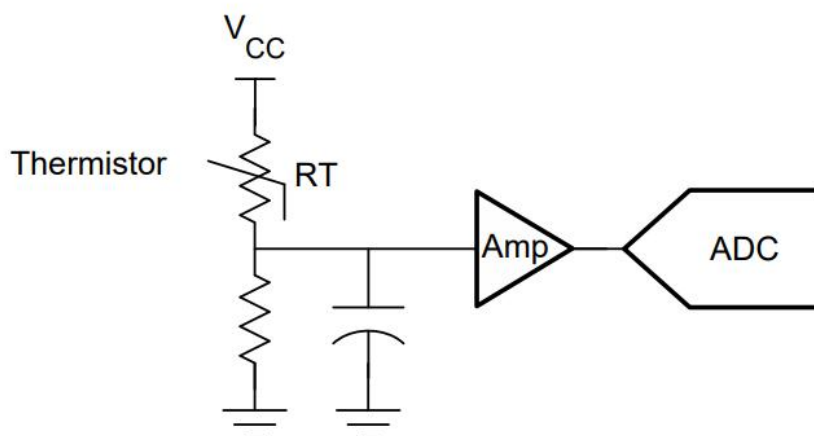


Рис. 1.5. Типова реалізація термістора

Термістор NTC є нелінійним і часто потребує збільшення вартості калібрування та програмного забезпечення. Винятком є кремнієві PTC термістори.

Справжню точність системи для термісторів часто важко визначити. До системних помилок NTC входять допуск NTC, резистор зміщення (допуск, температурний дрейф), АЦП (помилка квантування), помилка лінеаризації, опорна напруга (точність, температурний дрейф).

Термістори оцінюються за їхньою своєчасною реакцією на зміну температури, діапазоном резистивності при кімнатній температурі та номінальною потужністю відповідно до струму, що проходить через них. Доступні термістори з різними значеннями опору від кількох Ом до МОм, але для датчиків зазвичай використовуються ті типи з діапазоном кілоОм.

1.2.4. Термостат

Термостат — це контактний різновид електромеханічних датчиків температури, які в основному складаються з двох різних металів, таких як мідь, вольфрам, нікель або алюміній. Вони поєднуються, щоб створити біметалеву частину. Ступінь лінійного розширення двох різних металів змінюється та створює механічний вигин, коли деталь нагрівається.

Принцип біметалічного термостата

Біметалічний пристрій можна використовувати як електричний вимикач або як механічний спосіб роботи електричного вимикача в термостатичному моніторингу. Вони використовуються для керування нагрівальними елементами в печах, джерелах гарячої води, котлах і в системі охолодження радіаторів транспортних засобів.

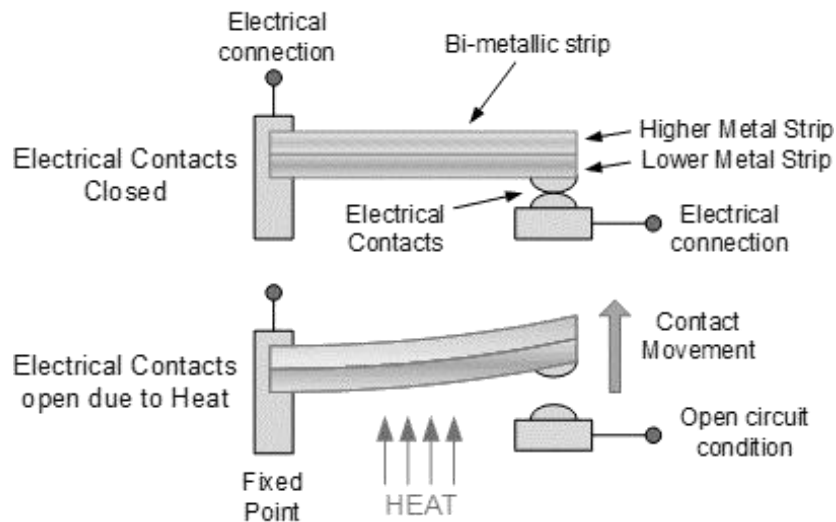


Рис. 1.6. Датчики температури: біметалічний термостат

Існує два спеціальні типи біметалічних термостатів, які класифікуються залежно від місця їх переміщення. По-перше, це «миттєва дія», яка генерує негайну реакцію «вкл/вимк» на електричних перемикачах у встановленому значенні. Термостати миттєвої дії зазвичай використовуються в будинках для контролю температури занурених джерел гарячої води, печей, прасок, їх також можна знайти на стінах для моніторингу системи опалення будинку.

Інший — повільніша «повзуча дія», яка повільно змінює свій стан із зміною температури. Це робить їх ідеальними для вимірювання температурних датчиків і циферблатів. Миттєві дії дуже дешеві та доступні на широкому робочому рівні, але одним основним недоліком є те, що вони мають великий діапазон гістерезису в своїх циклах «відкрито-закрито». Наприклад, вона може складати 20 о С, але насправді коливається від 18 до 22 о С.

1.2.5. Напівпровідники

Напівпровідники складаються з різних типів, таких як вихід струму, цифровий вихід, вихід напруги, кремнієвий вихід опору та діодні датчики температури. Нові напівпровідникові датчики температури забезпечують високу точність. Вони також пропонують високу лінійність у робочому діапазоні від 55 до 150°С. Використання внутрішніх підсилювачів може

масштабувати вихід до ідеальних значень, наприклад 10 мВ/°С. Вони також ефективні в схемах холодного спаю для високотемпературних станів термопар.

Датчики температури ІС покладаються на прогнозу температурну залежність забороненої зони кремнію. Прецизійний струм створює внутрішній р-n-перехід із прямим зміщенням із результуючим ΔV_{BE} , що відповідає температурі пристрою.

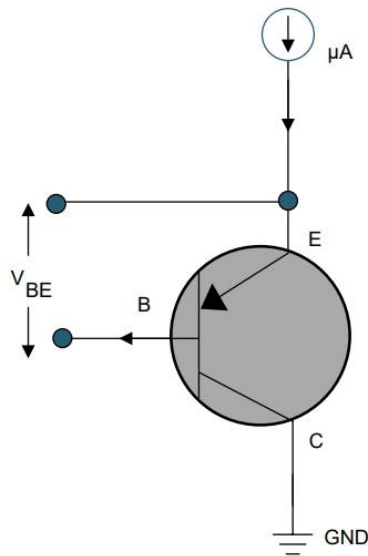


Рис. 1.7. Температурна залежність ширини забороненої зони кремнію

$$\Delta V_{BE} = \frac{KT}{q} \cdot \ln \left(\frac{I_{C1}}{I_{C2}} \right)$$

де K – постійна Больцмана, T – абсолютна температура, q – заряд електрона, I_{C2} – струм, пов'язаний з геометрією та температурою транзисторів. Рівняння передбачає напругу принаймні кілька сотень мВ на колекторі та ігнорує ранні ефекти.

Завдяки передбачуваній поведінці мікросхеми забезпечують високу лінійність і точність у широкому діапазоні температур (до $\pm 0,1^\circ C$). Датчики ІС можуть інтегрувати функціональність системи, яка забезпечує невелику площу та низьке енергоспоживання. Ці датчики дійсно працюють в обмеженому температурному діапазоні та пропонують менше варіантів

упаковки, які можуть вимірювати зовнішню температуру порівняно з термісторами.

Ці датчики, як правило, є повністю інтегрованими, а монолітні датчики та точність розроблені для всієї системи, а не для одного елемента.

1.2.6. Цифрові датчики температури

Цифрові датчики температури усувають вимоги до додаткових частин, таких як аналого-цифровий перетворювач, і немає необхідності калібрувати пристрої чи системи за певних температур, як це вимагається при використанні термісторів. Ці датчики використовуються з усім. Вони можуть збільшити основну температуру системи та контролювати функцію, щоб зробити її більш простою.

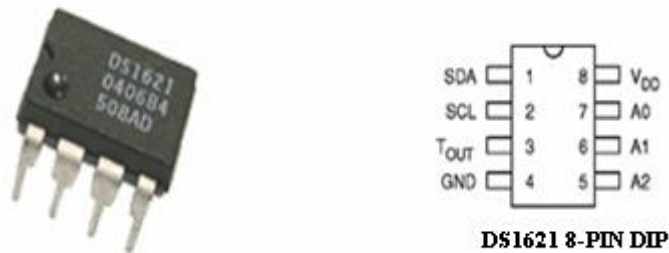


Рис. 1.8. Цифровий датчик температури

Основною перевагою цифрового датчика температури є його точний діапазон вихідної температури. Вихід датчика збалансований. Для цього не потрібні інші пристрої, наприклад аналого-цифровий перетворювач. Отже, його набагато легше використовувати, ніж термістор, який відображає нелінійну діаграму опір-температура. Відповідним прикладом таких датчиків є DS1621, який представляє 9-бітове зчитування температури.

1.3. Порівняння різних типів датчиків температури

У вбудованих системах існує постійна потреба у вищій продуктивності та більшій кількості функцій у меншому форм-факторі. Це вимагає від розробників системи моніторингу загальної температури, щоб забезпечити безпеку та захистити системи. Тенденція реєстрації даних датчиків ще більше спонукає до необхідності вимірювання температури не лише для

вимірювання системи чи умов навколишнього середовища, але й для компенсації чутливих до температури компонентів і підтримки точності всієї системи.

Тепловий дизайн Міркування щодо ефективного теплового моніторингу та захисту включають:

- Точність: точність датчика показує, наскільки близька температура до справжнього значення. Додатки повинні враховувати такі фактори, як лінійність і схеми збору даних у діапазоні робочих температур.
- Розмір: Хоча розмір датчика впливає на дизайн, аналіз загальної схеми може дати більш оптимізовану конструкцію.
- Розташування датчика: упаковка та розміщення можуть впливати на час відгуку та шлях провідності. Обидва є критичними для ефективного теплового проектування.

	ІС датчики	Термістор	РТД	Термопара
Діапазон	-55°C - +200°C	-100°C - +500°C	-240°C - +600°C	-260°C - +2300°C
Точність	Добра	Залежить від калібрування	Краща	Найкраща
Розмір	Найменший	Менший	Помірний	Великий
Складність	Легка	Помірна	Складна	Складна
Лінійність	Краща	Низька	Краща	Найкраща
Топологія	Точка-точка, багатоточкова, шлейфове з'єднання	Точка-точка	Точка-точка	Точка-точка
Ціна	Низька-помірна	Низька-помірна	Висока	Висока

Лінійність

Ідеальний датчик мав би ідеально лінійну реакцію: одинична зміна температури призвела б до одиничної зміни вихідної напруги в усьому температурному діапазоні датчика. Однак насправді жоден датчик не є ідеально лінійним. На рис представлено уявлення про реакцію температури на напругу трьох датчиків.

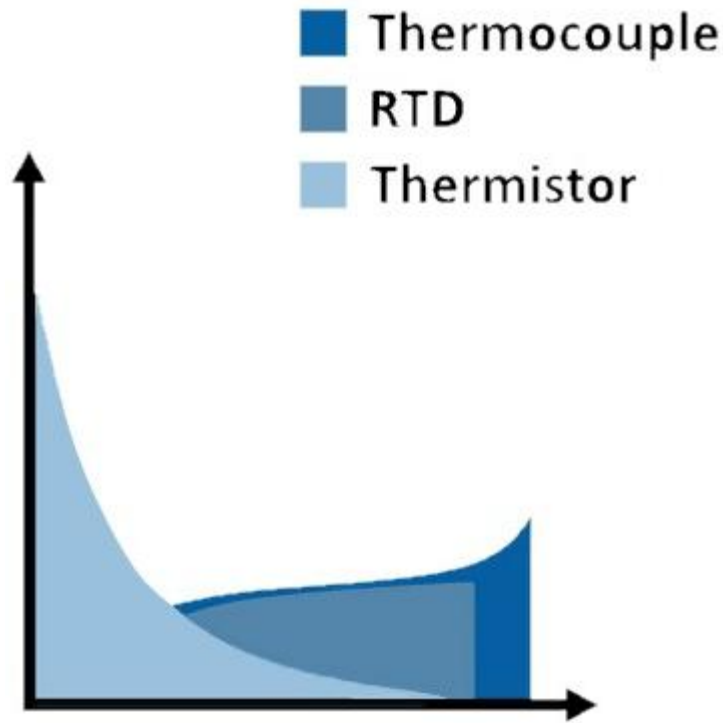


Рис. 1.9. Відповідь датчиків від температури до вихідного сигналу

Чутливість

Чутливість даного датчика вказує на відсоткову зміну вимірюваного виходу для даної зміни температури. Більш чутливий датчик, як-от термістор, може легше виявляти невеликі зміни температури, ніж менш чутливий датчик, як-от термопара. Ця чутливість, однак, відбувається за рахунок лінійності. Це може бути важливим фактором під час визначення ідеального вибору датчика для температур, які ви вимірюєте. Якщо ви маєте намір фіксувати зміни на частки градуса в невеликому діапазоні температур, термістор або RTD є більш ідеальним. Для фіксації більших змін температури в більш широкому діапазоні температур може бути достатньо термопари. Рис. дає відносне уявлення про напругу.

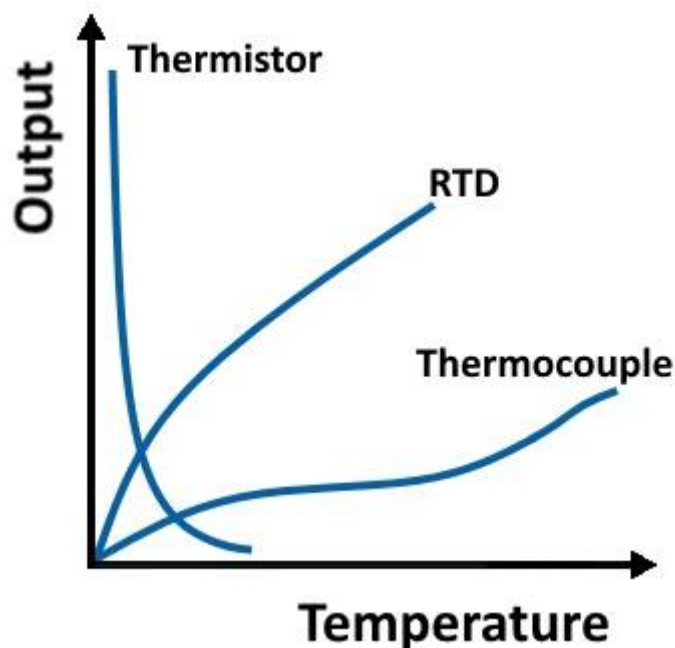


Рис. 1.10. Чутливість різних типів датчиків температури

Час реакції

Час відгуку – це міра часу, який потрібен датчику, щоб відреагувати на зміну температури. Багато факторів можуть призвести до збільшення або зменшення часу відповіді. Більший RTD або термістор, наприклад, має менший час відгуку, ніж менший. В обмін на цей недолік і гірше теплове шунтування, більший резистентний датчик температури або термістор менш чутливий до помилок самонагрівання. Подібним чином, незаземлені з'єднання термопар забезпечують менший час відгуку в обмін на електричну ізоляцію. На рис. показано відносну різницю в часі відгуку для незаземлених і заземлених термопар.

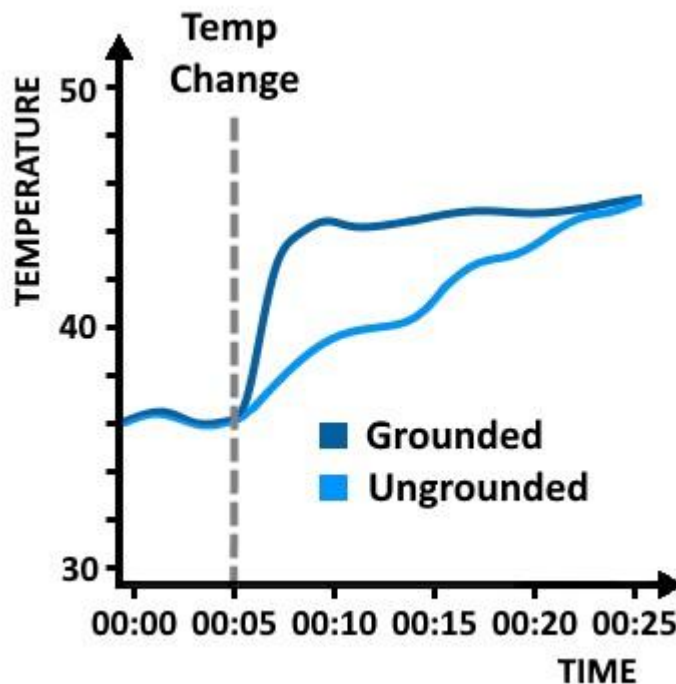


Рис. 1.11. Час відгуку заземлених і незаземлених термопар

Стабільність

Стабільність датчика температури є показником його здатності підтримувати постійний вихідний сигнал при заданій температурі. Матеріал відіграє ключову роль у стабільності певного датчика. З цієї причини, а також для забезпечення низької реактивності RTD часто виготовляють із платини. Однак підкладка, до якої прикріплена платина, може деформуватися під час тривалого впливу високих температур, що може спричинити додаткове та несподіване напруження, яке призведе до зміни виміряного опору.

Точність

Як і в будь-якій іншій програмі вимірювання, розуміння ваших потреб у точності має вирішальне значення для забезпечення надійних результатів. Вибір датчика та обладнання для вимірювання відіграє важливу роль в абсолютній точності вимірювань, але менші деталі, такі як кабельне підключення, відносна близькість до іншого обладнання, екранування, заземлення тощо, також можуть впливати на точність. Вибираючи датчик, зверніть увагу на вказані допуски та будь-які фактори, які можуть вплинути

на ці характеристики (наприклад, тривалий вплив високих температур). Також будьте обережні, вибираючи датчик і вимірювальний пристрій з однаковою точністю. RTD із жорстким допуском коштує дорожче, але ви можете не досягти додаткової точності, якщо використовуєте вимірювальний пристрій низької якості.

Довговічність

Щоб гарантувати, що датчики температури залишатимуться в робочому стані протягом усього періоду застосування, потрібно розуміти середовище, в якому їх розгортають. Деякі датчики (наприклад, термопари) за своєю конструкцією довговічніші. Однак метали, вибрані для конкретної термопари, мають різну стійкість до корозії. Крім того, датчик, укладений в ізоляційний мінерал і захисну металеву оболонку, є більш стійким до зношування та корозії з часом, але він коштує дорожче та забезпечує меншу чутливість. Слід також зауважити, що різні конфігурації датчиків можуть мати особливі вимоги до монтажу, щоб забезпечити міцне фізичне та теплове з'єднання.

Вартість

Як і в будь-якому аспекті проекту, вартість може бути ключовим обмежуючим фактором. У додатках із великою кількістю каналів, наприклад, переваги лінійності RTD можуть бути переважені відносним збільшенням вартості порівняно з термопарами. Розглядаючи загальну вартість системи, ви також повинні враховувати додаткову вартість проводки, монтажу та формування сигналу.

Вимоги до формування сигналу

Кожен тип датчика температури потребує певного рівня кондиціонування сигналу для належного отримання та оцифрування вимірюваного сигналу для обробки. Вибране вами вимірювальне обладнання може бути таким же важливим для забезпечення точних вимірювань, як і датчик, і може зменшити або посилити недоліки кожного типу датчика. Ці функції формування сигналу включають наступне:

- Ампліфікація

- Компенсація холодного спаю (лише термопари)
- Фільтрування
- Збудження (лише RTD і термістори)
- Коригування помилки зміщення
- Масштабування до одиниць вимірювання температури
- Корекція опору свинцю
- Межканальна ізоляція
- Виявлення відкритої термопари (тільки для термопар)

РОЗДІЛ 2 ТЕПЛОПРОВІДНІСТЬ ТА ІНСТРУКЦІЇ З ПРОЕКТУВАННЯ ДЛЯ ВИМІРЮВАННЯ ТЕМПЕРАТУРИ ПОВІТРЯ

Енергоємні електронні компоненти, такі як процесори, графічні процесори або FPGA, а також регулятори напруги нагріваються під час роботи. Деякі програми вимагають вимірювання температури навколишнього повітря, а інші потребують вимірювання температури сусіднього компонента на друкованій платі. Вимірювання температури навколишнього повітря за допомогою пристрою з технологією поверхневого монтажу (SMT) є складним завданням через тепловий вплив інших компонентів у системі. В інших системах, у яких необхідно виміряти температуру компонента, температура навколишнього повітря може впливати та погіршувати точність вимірювання.

Розробнику системи необхідно прийняти певні конструктивні рішення щодо типу упаковки та компонування друкованої плати під час інтеграції датчика температури. Цей розділ містить рекомендації для розробників системи та пояснює методи підвищення точності вимірюваної точки температури. Рекомендації надаються як для вимірювання температури повітря, так і для вимірювання температури компонентів. Детально описано методи компонування, орієнтацію пристрою та найкращі методи монтування.

2.1. Теплопровідність

Існує три способи передачі тепла: теплопровідність через тверді тіла, конвекція тепла через рідини та гази та тепло, що виділяється випромінюванням. У цьому розділі зосереджено увагу на теплопровідності, оскільки воно домінує в теплопередачі в друкованих платах і, отже, має найбільше значення для вимірювання температури.

Теплопровідність визначається як передача тепла через об'єм або тіло. Тепло передається через мікроскопічні зіткнення частинок; чим більше зіткнень, тим гарячіший об'єкт. Теплопередача відбувається, коли існує різниця температур між двома об'єктами або між різними частинами об'єкта, і її швидкість залежить від геометрії, товщини та матеріалу об'єкта.

Відповідно до закону рівноваги тепло передається від гарячішого тіла до холоднішого, доки вся система не досягне остаточної рівноваги, як показано на рис. 2.1. Між двома об'єктами, які мають рівноважну температуру, немає загальної теплопередачі. Рівняння теплопередачі через провідність показано в рівнянні (1).

$$\frac{Q}{t} = kA \frac{(T_2 - T_1)}{d} \quad (1)$$

де

- Q/t : Швидкість теплопередачі [Дж/с]
- k : теплопровідність матеріалу [Вт/м × К]
- A : Поверхня контактної площі [м²]
- ΔT : різниця температур T_1 одного об'єкта та температури T_2 іншого [К]
- d : товщина матеріалу [м]

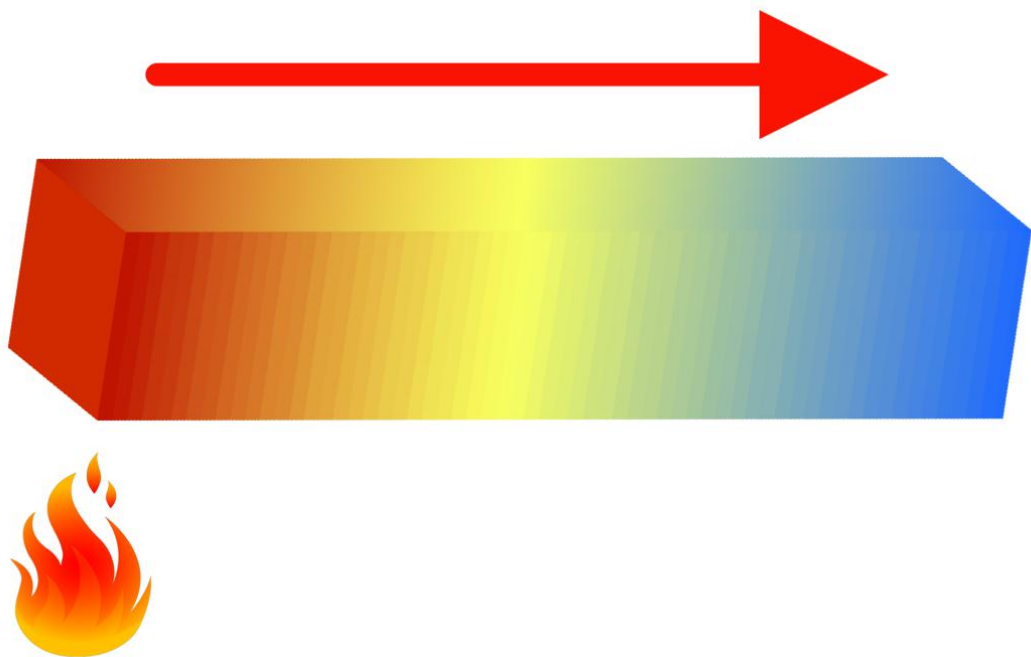


Рис. 2.1. Модель теплопровідності

Теплопровідність (k) є мірою здатності матеріалу проводити тепло. Вона використовується для опису того, як тепло проводить через матеріал. Метали мають високу теплопровідність, тоді як такі матеріали, як повітря, вовна, папір або пластик, погано проводять тепло. Матеріали з дуже низькою теплопровідністю, такі як пінополістирол, діють як теплоізолятор.

Матеріали, які найбільше стосуються термічного аналізу друкованих плат (ДП), це мідь, FR4 і паяльна маска. Мідь – чудовий провідник тепла; він проводить тепло значно швидше, ніж FR4. У таблиці 2.1 наведено теплопровідність ДП. Чим вище значення, тим ефективніший матеріал у передачі тепла, що призводить до меншого часу теплового відгуку. Для низьких значень k температурний градієнт між джерелом і датчиком може бути значно великим, і його слід ретельно враховувати під час компонування.

Таблиця 2.1. Коефіцієнти теплопровідності вибраних матеріалів

Матеріал	Теплопровідність k [W/(m×K)]
Повітря	0.0275
Паяльна маска	0.245
FR4	0.25
Золото	314
Мідь	385
Срібло	406

2.2. Визначення домінуючого шляху теплопровідності

Датчики температури для поверхневого монтажу мають кілька переваг перед датчиками з корпусами з наскрізними отворами. Переваги включають менший розмір упаковки з низьким профілем, зручне розміщення друкованої плати та простоту складання. Однак датчики температури SMT може бути важко ізолювати, оскільки вони мають тенденцію вимірювати температуру друкованої плати, а не температуру навколишнього повітря. Тому необхідно використовувати спеціальні методи компонування, якщо метою датчика температури є вимірювання температури навколишнього середовища, а не температури друкованої плати. Локальні аналогові або цифрові датчики температури визначають температуру, вимірюючи власну температуру матриці. Тому важливо розуміти основні шляхи передачі температури між

матрицею датчика температури та об'єктом або середовищем, температуру якого необхідно визначити.

Тепло передається в основному такими шляхами:

1. Плата для кріплення матриці (DAP), якщо вона є, забезпечує найбільш домінуючий тепловий шлях між друкованою платою та матрицею

2. Виводи забезпечують найбільший тепловий шлях, якщо тип корпусу не включає DAP

3. Формова суміш забезпечує додатковий тепловий шлях, але через її низьку теплопровідність будь-яка теплопередача через саму формувальну суміш відбувається повільніше, ніж теплопередача через проводи або DAP.

Вибір типу корпусу визначає, наскільки швидко датчик температури може реагувати на зміни температури. На рис.2.2. показано відносні швидкості теплового відгуку різних класів вибраних типів корпусів SMT, які використовуються для вимірювання температури.



Рис. 2.2. Швидкість відносного теплового відгуку (типова)

Типи корпусів без компаунду для форми (CSP, DSBGA) і корпуси з DAP (QFN, DFN) добре підходять, якщо потрібна швидка передача тепла від друкованої плати, тоді як типи корпусів без DAP кращі в додатках, у яких нижча швидкість реакції є бажаними. Швидка швидкість теплового відгуку дозволяє температурному датчику швидко реагувати на будь-які зміни температури і, отже, забезпечувати точні показання.

2.2.1 Безсвинцеві корпуса CSP, DSBGA

Виводи Wafer Chip Scale Package (WCSP) являють собою кульки Ball Grid Array (BGA), оброблені безпосередньо на матриці. Тепло від кульок BGA

передається напряму до матриці, а не через штифти або через накладку для кріплення матриці, як показано на рис.2.3. Як правило, це тип упаковки з найшвидшою температурною реакцією, оскільки немає форми для нагрівання .

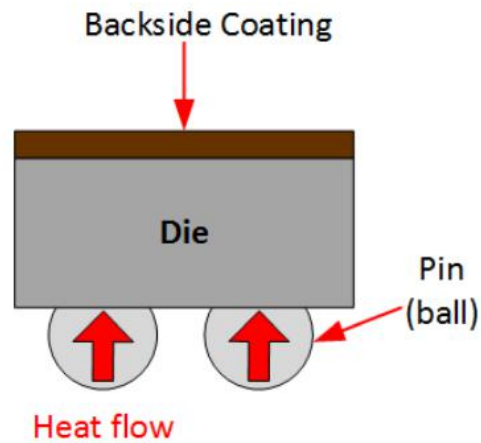


Рис.2.3. Теплопередача WCSP, переріз корпусу

2.2.2 Безвивідні корпуси QFN, DFN

Упаковки з DAP, такі як упаковки QFN і DFN, мають велику відкриту поверхню, через яку тепло може швидко передаватись. Ці типи корпусів швидко реагують на зміни температури мідної поверхні, до якої припаяно DAP. Оскільки матриця розташована безпосередньо на термічній прокладці, тепло може швидко передаватись від термопрокладки до матриці.

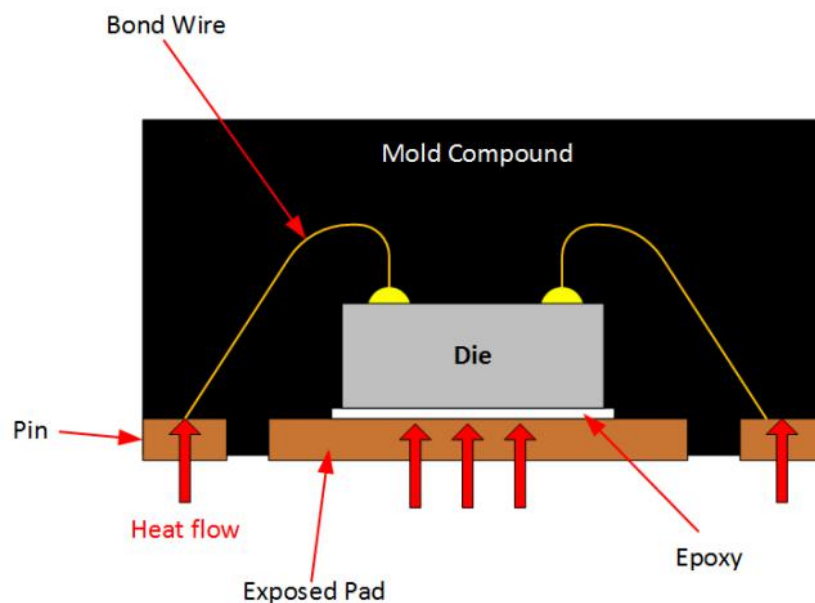


Рис.2.4. Поперечний переріз корпусу теплопередачі WSON

2.2.3 Безвивідні корпуси без підкладки для кріплення матриці DFN

Безвивідні корпуси без DAP, такі як 2-контактний корпус DFN, показаний на рис.2.5, передають найбільше тепла через самі штирі. Невелика упаковка цього типу все ще може швидко реагувати на зміну температури через невелику термічну масу суміші для форми.

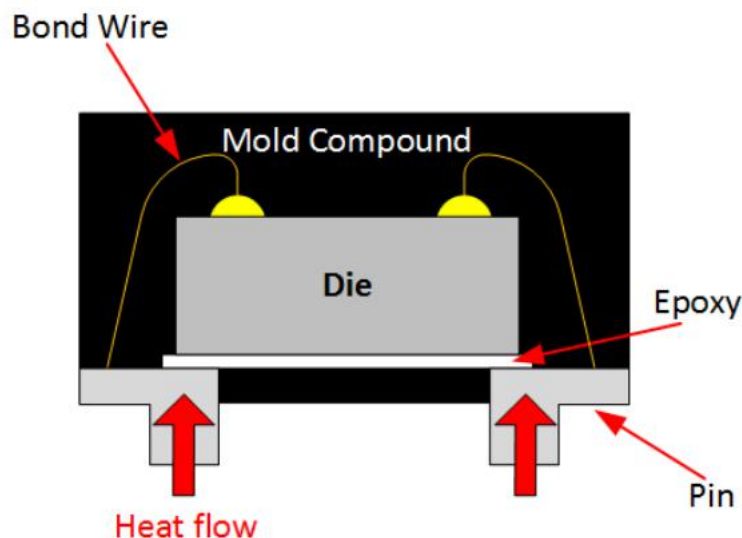


Рис.2.5. Теплопередача WSON, переріз корпусу

2.2.4 Корпуса MSOP, SOIC, VSSOP, SOT-563, SOT-23

Інші пакети, такі як SOIC-8, MSOP-8, SOT-563 і SOT-23, передають найбільше тепла через свої проводи (рис.2.6-2.8). Проводи передають від 60% до 70% тепла до термодатчика матриці.

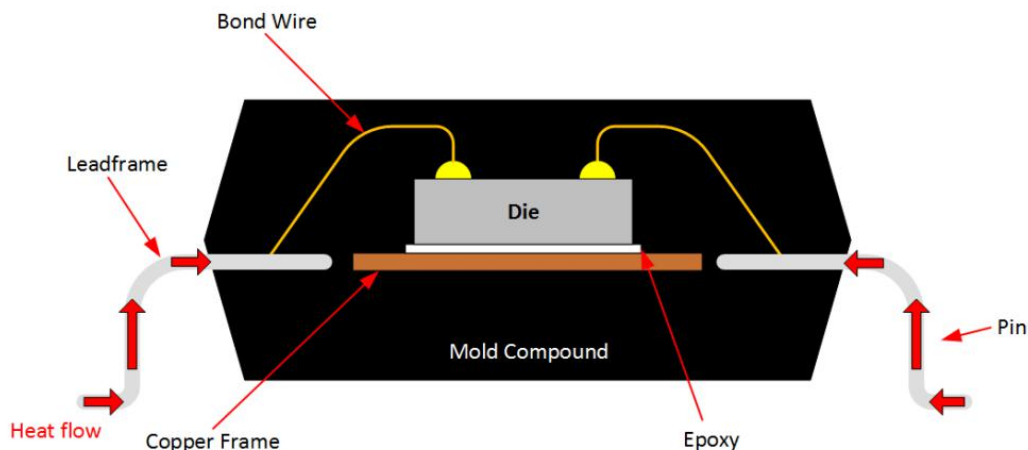


Рис.2.6. Теплопередача MSOP-8, SOIC-8, переріз корпусу

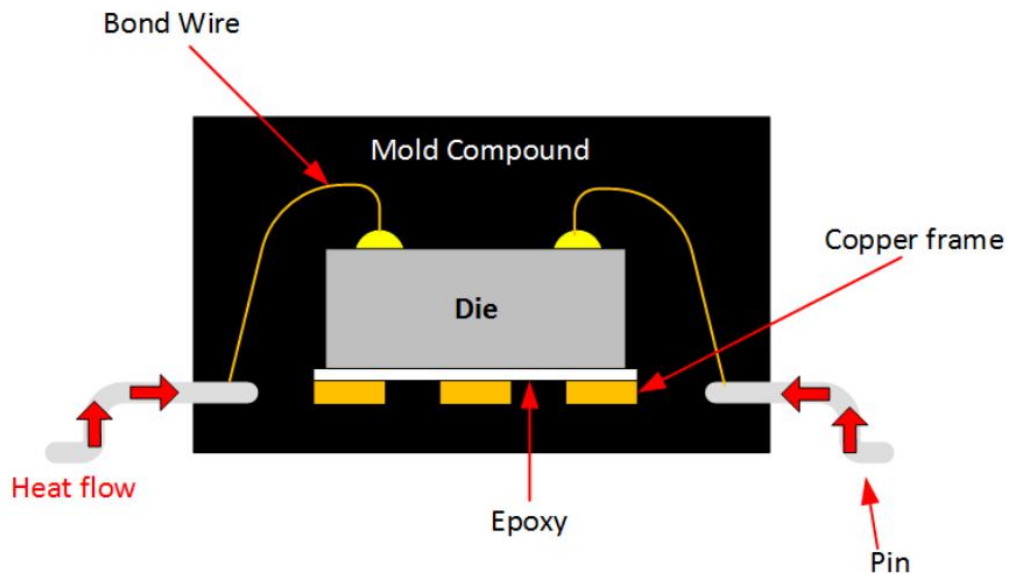


Рис.2.7. Теплопередача SOT-563, переріз корпусу

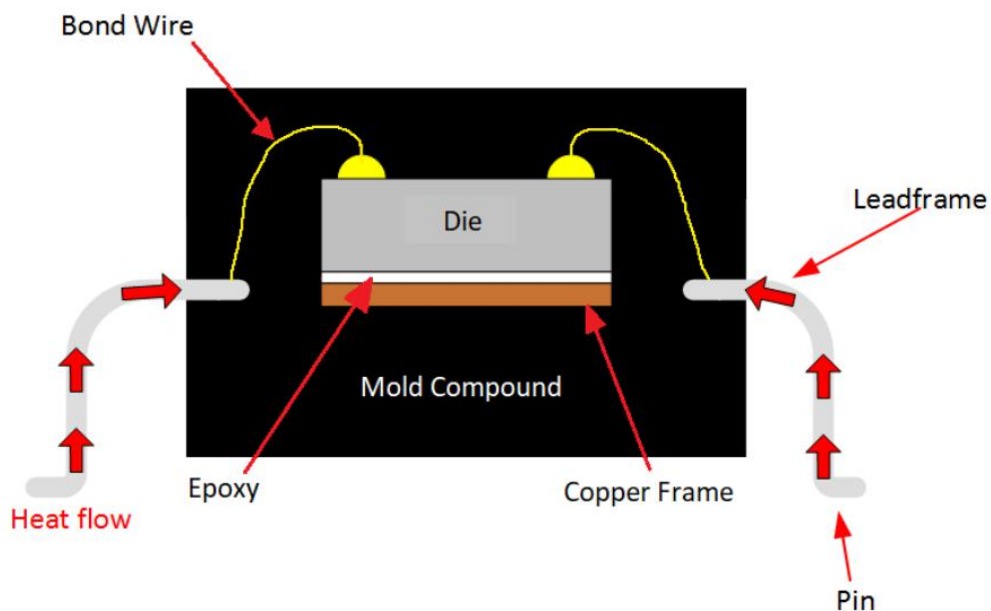


Рис.2.8. Теплопередача SOT-23, переріз корпусу

2.3. Визначення теплопровідності через друковану плату

Енергоємні компоненти можуть виділяти значну кількість тепла під час роботи, і розробник друкованої плати повинен розуміти, як друкована плата передає тепло. Компонування друкованої плати впливає на теплопровідність і, отже, на вимірювання температури. Розуміння загального термічного опору друкованої плати допоможе розробнику друкованої плати визначити, чи потрібно використовувати заповнені чи покриті переходами,

використовувати більш товсте мідне покриття чи додати додаткові шари міді для швидшого розсіювання тепла.

2.3.1 Загальне рівняння теплопровідності

Термічний опір можна виразити наступним рівнянням:

$$\theta = \frac{L}{k \cdot A_{CS}} \quad (2)$$

де

- θ – термічний опір [К/Вт]
- k – коефіцієнт теплопровідності [Вт/(м*К)]
- L – довжина теплового шляху [м]
- A_{CS} – це площа поперечного перерізу, в якій подається тепло [м²]

Щоб обчислити теплопровідність через друковану плату, окремі шляхи можна розбити та проаналізувати окремо. Основними компонентами є теплопровідність через друковану плату (див. розділ 2.3.4 і розділ 2.3.2) і теплопровідність через перехідний отвір (див. розділ 2.3.5). Найпоширенішими матеріалами для багатьох застосувань для друкованих плат є FR4, мідь і паяльні маски. Застосовуючи рівняння 2 до перпендикулярного шляху для відповідних матеріалів друкованої плати, поздовжнього шляху та теплового потоку через отвори окремо, можна точно змодельовати теплопровідність через друковану плату.

2.3.2 Поздовжня теплопровідність

На рис. 2.9. показано поздовжній шлях провідності друкованої плати з напрямком теплового потоку від джерела тепла вздовж FR4.

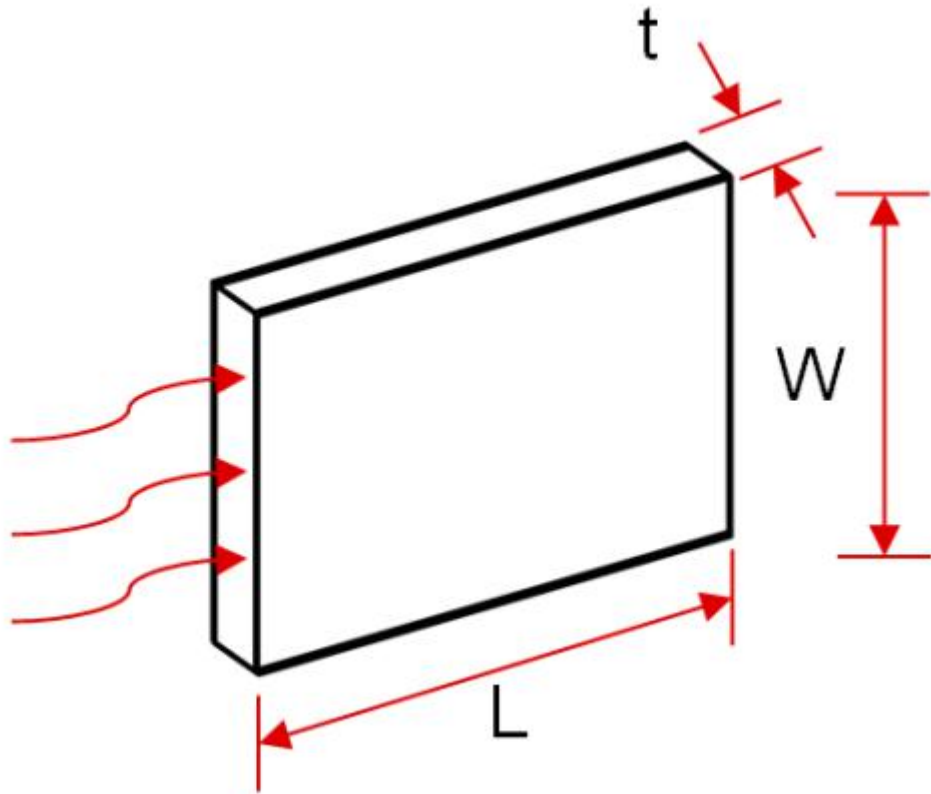


Рис. 2.9. Поздовжній теплопровідний потік

$$\theta = \frac{L}{k \cdot A_{CS}} = \frac{L}{k \cdot W \cdot t} \quad (3)$$

де

- L – довжина шляху теплового потоку
- W – ширина, t – товщина
- $W \cdot t = A_{CS}$ – площа поперечного перерізу, куди подається тепло

2.3.3 Приклад: визначення домінуючого поздовжнього шляху теплопровідності

Застосування рівняння 3 до шару FR4 товщиною 1,6 мм квадратних розмірів (1 м × 1 м) призводить до термічного опору 2500 °C/Вт, як показано в рівнянні 4:

$$\theta_{FR4} = \frac{L}{k \cdot W \cdot t} = \frac{1m}{0.25 \frac{W}{m \cdot ^\circ C} \cdot 1m \cdot 2 \cdot 1.6 \cdot 10^{-3}m} = 2500 \text{ } ^\circ C/W \quad (4)$$

Дві мідні пластини товщиною 1 унція (35 мкм) однієї друкованої плати матимуть термічний опір 3710 °C/Вт, як розраховано в рівнянні 5.

$$\theta_{Cu} = \frac{L}{k \cdot W \cdot t} = \frac{1m}{385 \frac{W}{m \cdot ^\circ C} \cdot 1m \cdot 2 \cdot 35 \cdot 10^{-6}m} = 3710 \text{ } ^\circ C/W \quad (5)$$

Хоча товщина мідної пластини значно менша за товщину шару FR4, здатність передавати тепло є такого ж порядку. Це тому, що теплопровідність міді приблизно в 1500 разів більша, ніж теплопровідність FR4. Можна побачити, що мідні площини наведеного вище прикладу друкованої плати передають тепло трохи повільніше, ніж значно товщий шар FR4.

У поздовжньому напрямку кілька шарів друкованої плати потрібно розглядати паралельно. Порівняно з друкованою платою без мідних потоків, можна майже подвоїти швидкість теплопередачі вздовж площини, додавши два шари мідних потоків товщиною 1 унція.

Оскільки різні шари передають тепло паралельно, ефективний загальний термічний опір поперечного перерізу становить $1494^\circ C/W$, як розраховано за рівнянням 6:

$$\theta_{total} = \theta_{Cu} || \theta_{FR4} = \frac{\theta_{Cu} \cdot \theta_{FR4}}{\theta_{Cu} + \theta_{FR4}} = \frac{3710 \cdot 2500}{3710 + 2500} = 1494 \text{ } ^\circ C/W \quad (6)$$

2.3.4 Перпендикулярна теплопровідність

На рис. 2.10 показано перпендикулярний теплопровідний потік друкованої плати.

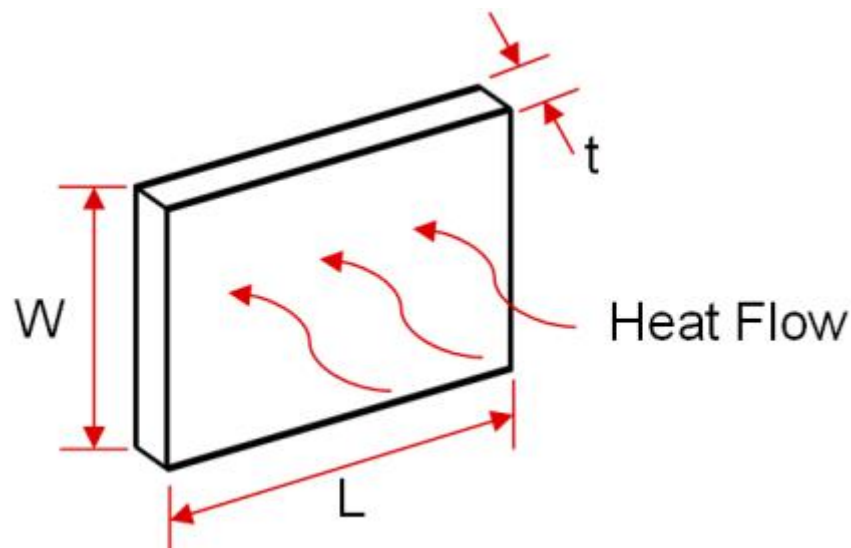


Рис. 2.10. Перпендикулярний теплопровідний потік

Рівняння 7 є рівнянням перпендикулярного теплопровідного потоку для визначення теплового опору матеріалу:

$$\theta = \frac{t}{k \cdot A_{CS}} = \frac{t}{k \cdot W \cdot L} \quad (7)$$

де

- t – шлях теплового потоку (тепло протікає через товщину матеріалу) [м]
- $W \cdot L = A_{CS}$ – площа поперечного перерізу, куди подається тепло [м²].

2.3.5 Теплопровідність через отвір

Розрахунок шляху теплопровідності отвору (як показано на рис. 2.11) може бути корисним, щоб визначити, чи достатньо звичайного отвору, чи потрібен більший розмір або кількість отворів, чи потрібно заповнити отвори струмопровідним заповненням. Провідне заповнення швидше передає тепло на протилежну сторону плати, але також збільшує вартість виробництва друкованої плати.

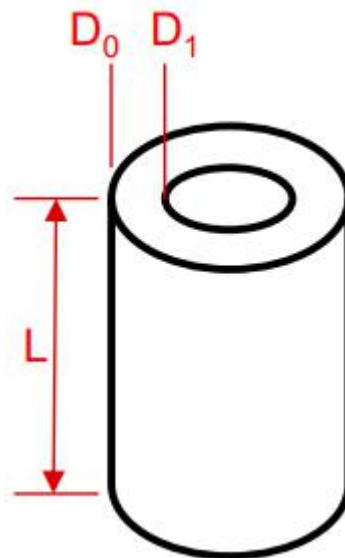


Рис. 2.11. Провідність через Via

Рівняння 8 застосовує загальне рівняння теплопровідності до отвору:

$$\theta = \frac{L}{k \cdot A_{CS}} = \frac{L}{k \cdot \pi \cdot \left(\left(\frac{D_0}{2} \right)^2 - \left(\frac{D_1}{2} \right)^2 \right)} \quad (8)$$

де

- L – довжина отвору (тепло протікає через довжину циліндра) [м]

- D_0 – зовнішній діаметр отвору [м]
- D_1 – внутрішній діаметр отвору [м].

2.3.6 Приклад: визначення основного перпендикулярного шляху теплопровідності

У перпендикулярному напрямку розробник друкованої плати може захотіти порівняти термічний опір отвору з еквівалентним тепловим опором FR4, щоб визначити, чи є розміщення додаткових отворів корисною технікою швидкої передачі тепла з одного боку друкованої плати на іншу.

Бічні стінки отвору без намету, незаповненого отвору з розміром отвору 0,5 мм, товщиною міді бічної стінки 35 мкм і довжиною 1,6 мм мають термічний опір 81 °C/W, як показано в рівнянні 9. Примітка що товщина бічної стінки отвору часто відрізняється від товщини мідного покриття та залежить від розмірів отвору та процесу виробництва виробника друкованої плати.

$$\theta_{Cu} = \frac{L}{k \cdot A_{CS}} = \frac{L}{k \cdot \pi \cdot \left(\left(\frac{D_0}{2} \right)^2 - \left(\frac{D_1}{2} \right)^2 \right)} = \frac{1.6 \cdot 10^{-3}}{385 \frac{W}{m \cdot ^\circ C} \cdot \pi \cdot \left(\left(\frac{0.5 \cdot 10^{-3}}{2} \right)^2 - \left(\frac{0.43 \cdot 10^{-3}}{2} \right)^2 \right)} = 81 \text{ } ^\circ C/W \quad (9)$$

Щоб отримати точний результат, термічний опір повітряного циліндра всередині отвору також необхідно розрахувати та враховувати паралельно з тепловим опором бічних стінок отвору.

$$\theta_{Cu} = \frac{L}{k \cdot A_{CS}} = \frac{L}{k \cdot \pi \cdot \left(\left(\frac{D_0}{2} \right)^2 - \left(\frac{D_1}{2} \right)^2 \right)} = \frac{1.6 \cdot 10^{-3}}{385 \frac{W}{m \cdot ^\circ C} \cdot \pi \cdot \left(\left(\frac{0.43 \cdot 10^{-3}}{2} \right)^2 - \left(\frac{0}{2} \right)^2 \right)} = 400646 \text{ } ^\circ C/W \quad (10)$$

Рівняння 10 показує, що термічний опір повітряного циліндра перевищує 400 000 °C/W. Оскільки він приблизно в 5000 разів більший за тепловий опір бічних стінок проходу, теплопровідність повітря має незначний вплив і ним можна знехтувати, як це доведено рівнянням 11.

$$\theta_{via} = \theta_{Cu} || \theta_{air} = \frac{\theta_{Cu} \cdot \theta_{air}}{\theta_{Cu} + \theta_{air}} = \frac{81 \cdot 400646}{81 + 400646} = 81 \text{ } ^\circ C/W \quad (11)$$

Заповнений повітрям отвір перехідного отвору не впливає на швидкість теплопередачі, тому майже весь теплообмін стандартного отвору відбувається через його бічні стінки. Однак отвори, в яких отвір заповнено

іншим матеріалом, можуть виграти від внеску теплопередачі цього матеріалу. Деякі конструкції вимагають заповнення отворів для передачі тепла навіть швидше, ніж звичайні отвори. Заповнені отвори слід розглядати, якщо навіть декілька паралельних стандартних отворів не забезпечують достатньо високу швидкість теплопередачі, щоб відповідати специфікаціям системи.

Термічний опір 81 °C/Вт для незаповненого отвору з цього прикладу можна порівняти з термічним опором суцільного циліндра FR4 такого ж зовнішнього діаметра, щоб визначити, наскільки ефективнішим є мідний отвір для передачі тепла з одного боку PCB до іншого. Рівняння 12 показує, що термічний опір циліндра FR4 такого ж розміру становить 32 595 °C/Вт, що приблизно в 400 разів більше, ніж термічний опір отвору.

$$\theta_{FR4} = \frac{L}{k \cdot A_{CS}} = \frac{L}{k \cdot \pi \cdot \left(\left(\frac{D_0}{2} \right)^2 - \left(\frac{D_1}{2} \right)^2 \right)} = \frac{1.6 \cdot 10^{-3}}{0.25 \frac{W}{m \cdot ^\circ C} \cdot \pi \cdot \left(\left(\frac{0.5 \cdot 10^{-3}}{2} \right)^2 - \left(\frac{0}{2} \right)^2 \right)} = 32595 \text{ } ^\circ C/W \quad (12)$$

Заповнений повітрям отвір сприяє незначній теплопередачі. Однак, оскільки теплопровідність міді приблизно на 1500 вище, ніж у FR4, перехідний отвір із зазначеними вище розмірами здатний передавати тепло на протилежну сторону друкованої плати через бічні стінки отвору приблизно в 400 разів швидше, ніж циліндр FR4 такого ж зовнішнього діаметра. Таким чином, розміщення кількох паралельних незаповнених отворів може бути дуже ефективним методом швидкої передачі тепла від одного боку друкованої плати до іншого в межах локалізованої області.

2.4. Рекомендації з проектування для вимірювання температури повітря

Для деяких застосувань бажано розробити низьку теплопровідність між друкованою платою та датчиками температури. Це бажано, коли потрібно виміряти температуру навколишнього повітря, а не температуру друкованої плати або сусідніх компонентів.

Приклади таких програм включають

- Вимірювання температури навколишнього повітря термостатом
- Внутрішні та відкриті метеостанції

- Бездротовий сенсорний вузол

У таких системах цікаві повільні зміни температури повітря, тоді як джерела тепла, такі як тепло сусіднього процесора, призведуть до неточних показань.

2.4.1. Розгляд заземлення

Через вищу теплопровідність міді заземлення між іншими мікросхемами та датчиком призведе до небажаної теплопередачі, якої слід уникати. Найкраще уникати мідних площин поблизу датчика температури, які з'єднані з мідними площинами інших мікросхем, як показано на рис. 2.12.

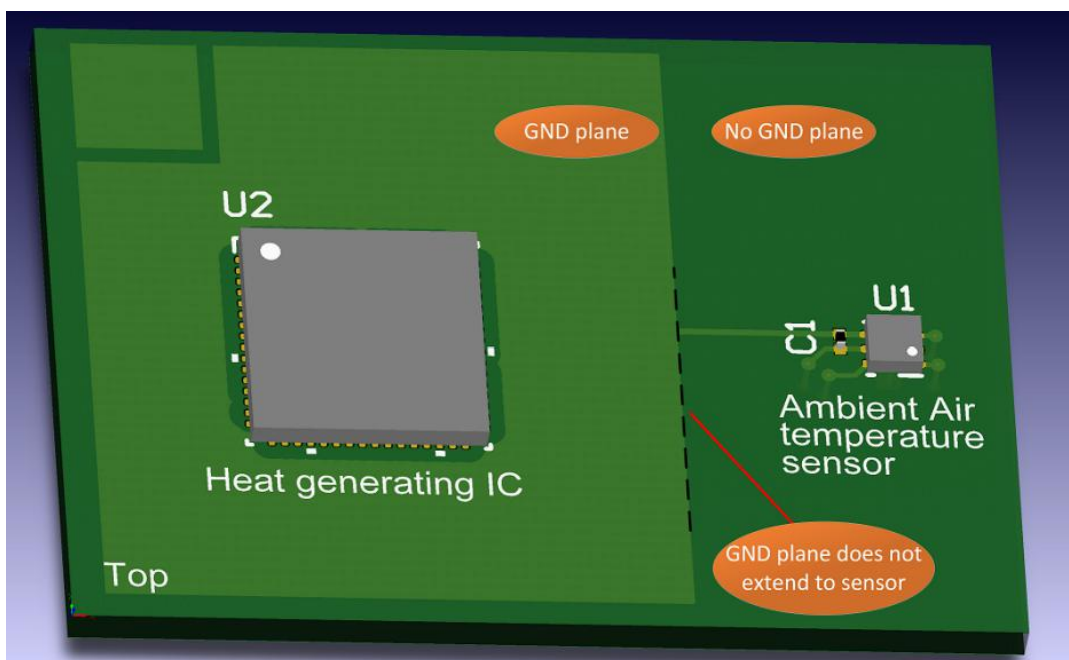


Рис.2.12. Площина GND не тягнеться до датчика температури
(вид зверху)

Для ще кращих результатів створіть окрему невелику мідну площину з обох боків датчика, як показано на рисунках 13 і 14, і додайте кілька отворів для термічного з'єднання верхньої та нижньої площин. Через низьку теплопровідність паяльної маски порівняно з міддю рекомендується створити виріз для паяльної маски навколо мідної площини. Це дозволить датчику реагувати на вимірювання температури навколишнього повітря значно швидше, ніж у системах, у яких мідна площина покрита паяльною маскою.

Додайте фізичний зазор між площиною навколо датчика та площиною решти друкованої плати. Заштриховані площини заземлення в головній частині друкованої плати додатково зменшують потік тепла від інших мікросхем до датчика.

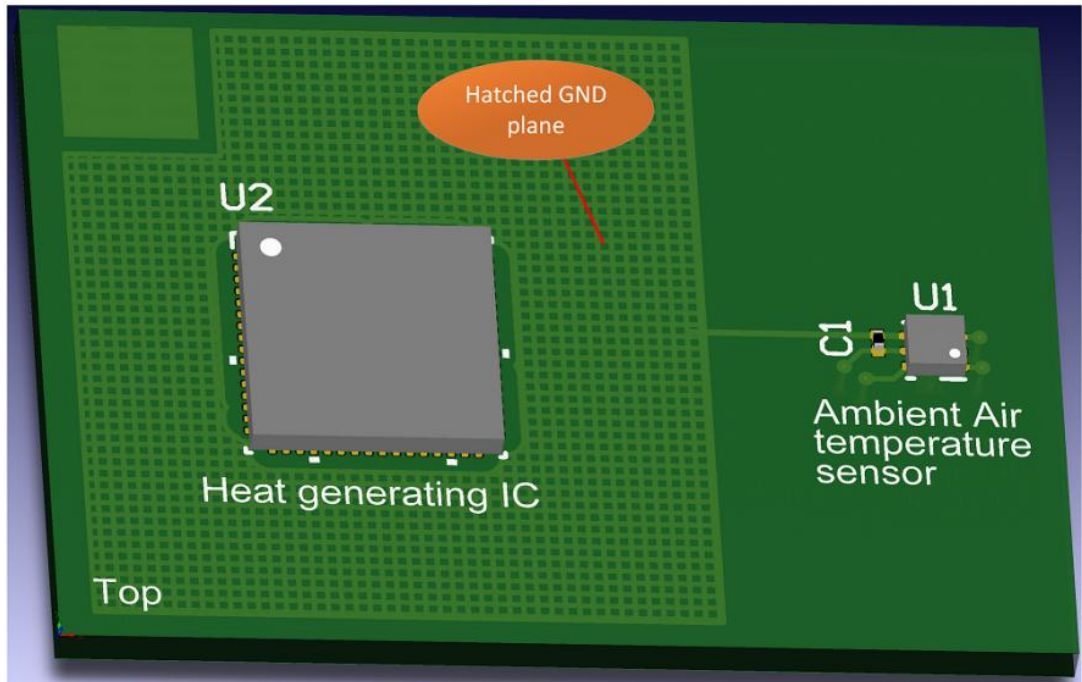


Рис.2.13. Заштрихована площина GND (вид зверху)

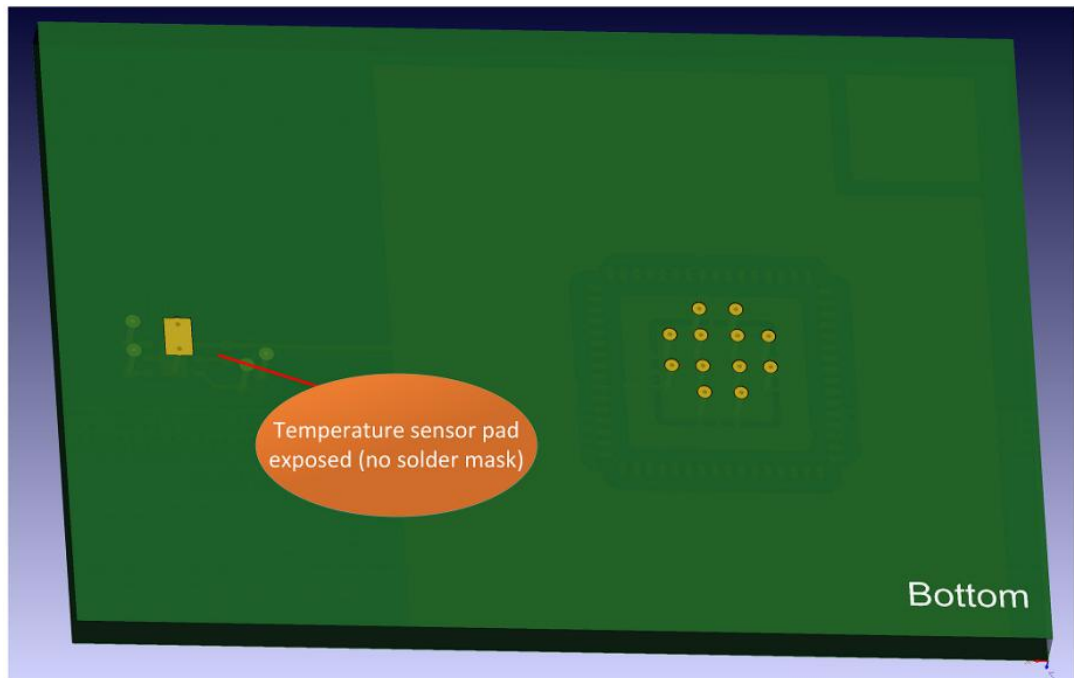


Рис.2.14. Відкрита контактна площадка GND датчика (вигляд знизу)

2.4.2. Розбиття друкованої плати на розділи

Датчик температури має бути в зоні друкованої плати, яка знаходиться якомога далі від основних мікросхем, що виробляють тепло, як показано на рис.2.15. Цього можна досягти, розмістивши датчик у кутку друкованої плати, подалі від інших компонентів. Це мінімізує вплив інших компонентів на платі на показання температури.

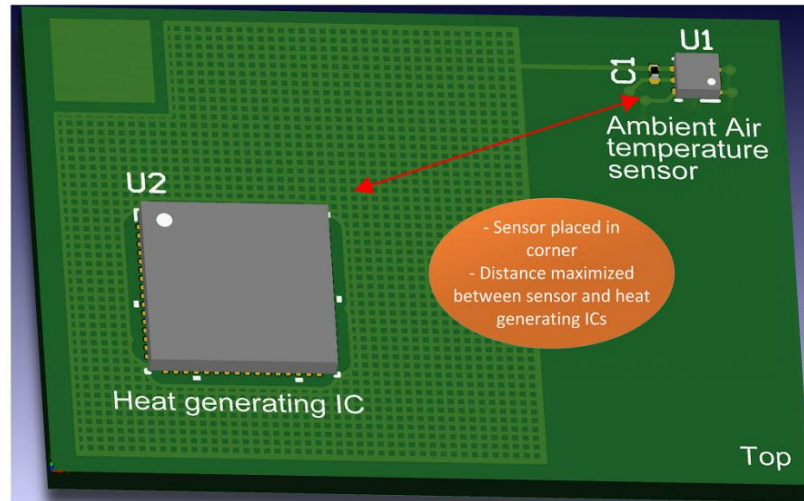


Рис.2.15. Датчик розміщений у кутку друкованої плати

2.4.3. Острів ізоляції

Якщо це можливо, часткова траса маршрутизатора навколо датчика температури створює ізоляційний острівець, який значно зменшує передачу тепла від основного джерела тепла до датчика. Тепловіддача знижується, тому що теплопровідність повітря в порівнянні значно нижча, ніж теплопровідність FR4. Приклад ізоляційного острова показано на рис.2.16.

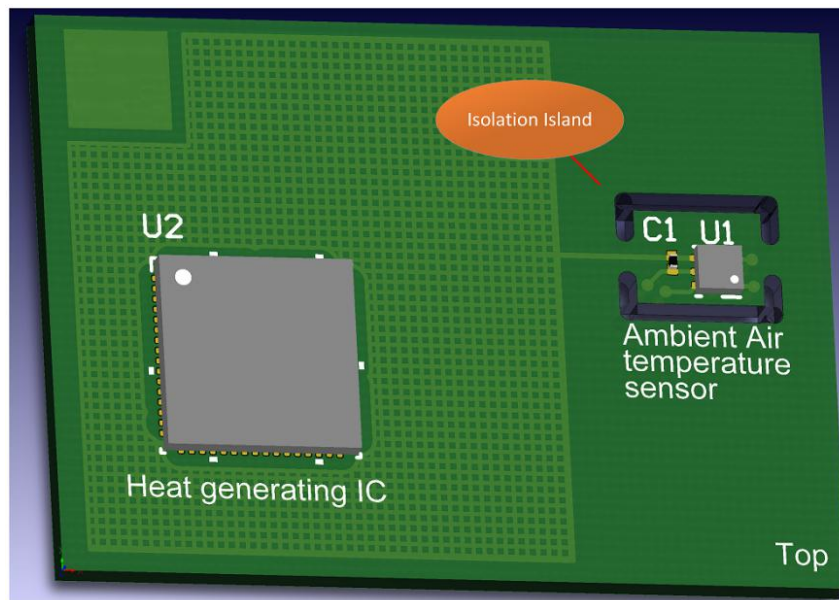


Рис.2.16. Ізоляційний острів значно зменшує передачу тепла від основного джерела тепла до датчика температури

2.4.4. Перфорація

Як альтернативу ізоляційному островцю, описаному в розділі 2.4.3, можна додати перфорацію навколо секції з датчиком температури, як показано на рис.2.17. Це значно мінімізує передачу тепла через матеріал FR4.

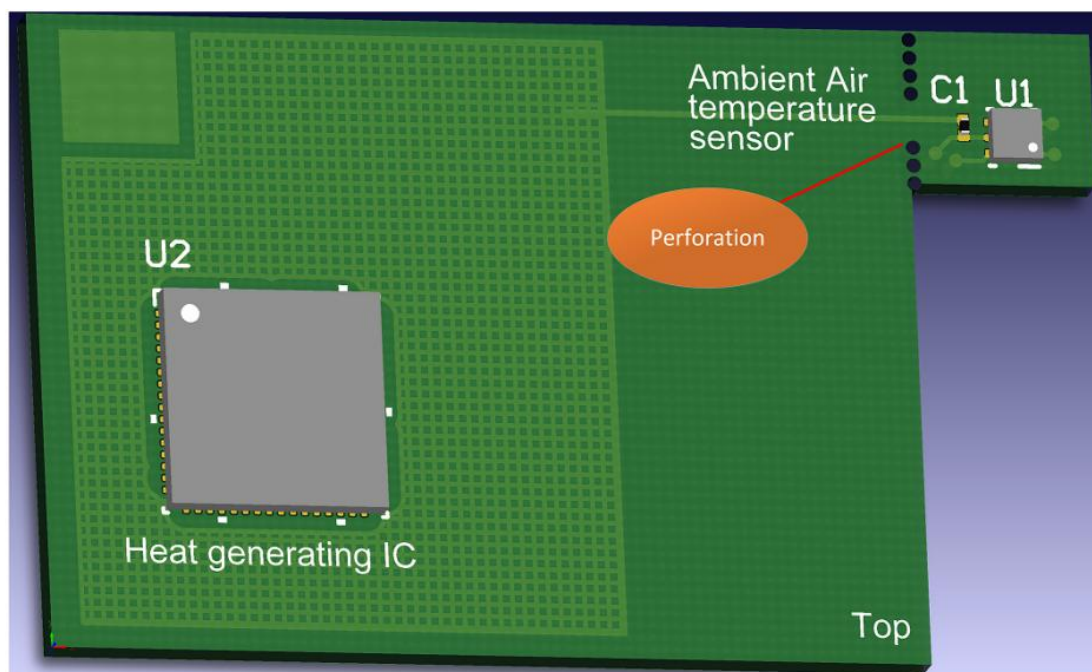


Рис.2.17. Перфорація зменшує передачу тепла від джерела тепла до датчика температури

2.4.5. Крайовий з'єднувач

Мініатюрна друкована плата, яка містить лише датчик температури та монтується за допомогою крайового з'єднувача до основної друкованої плати, є дуже ефективним методом уникнення значної передачі тепла від основної друкованої плати до датчика температури. Крайовий роз'єм в ідеалі має бути встановлений подалі від основних джерел тепла на головній друкованій платі, щоб тепло, що випромінюється від мікросхем, не заважало вимірюванню температури. Ця техніка показана на рис.2.18:

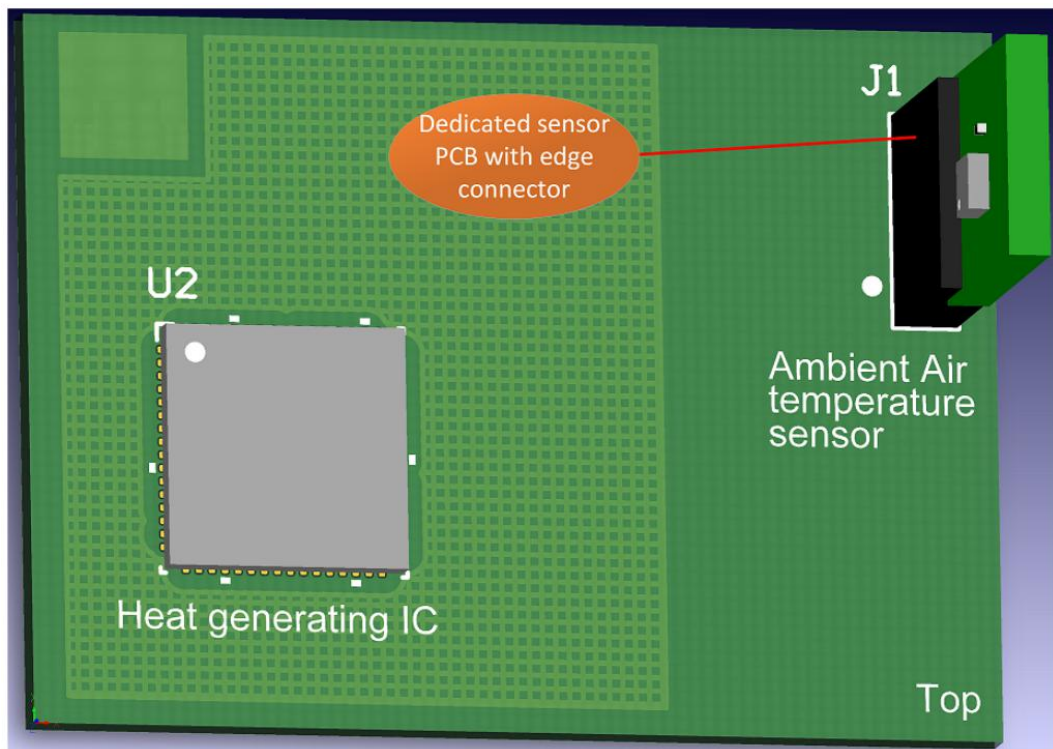


Рис.2.18. Спеціальна друкована плата датчика з крайовим роз'ємом

2.4.6. Контроль теплової маси друкованої плати

Теплова маса - це здатність матеріалу зберігати теплову енергію. Матеріал з високою термічною масою реагуватиме на коливання температури повільніше, ніж матеріал із меншою термічною масою. Щоб зберегти теплову масу друкованої плати якомога меншою, рекомендується використовувати тонку друковану плату (наприклад, 0,8 мм, а не стандартну товщину 1,6 мм FR4) або навіть розмістити датчик температури на гнучкій друкованій платі, як показано на рис.2.19. У поєднанні з одним із методів зменшення площі поверхні друкованої плати (див. розділи 2.4.3 і 2.4.4) тонка

друкована плата може відповідати змінам температури повітря набагато швидше, ніж велика товста друкована плата з високою термічною масою.

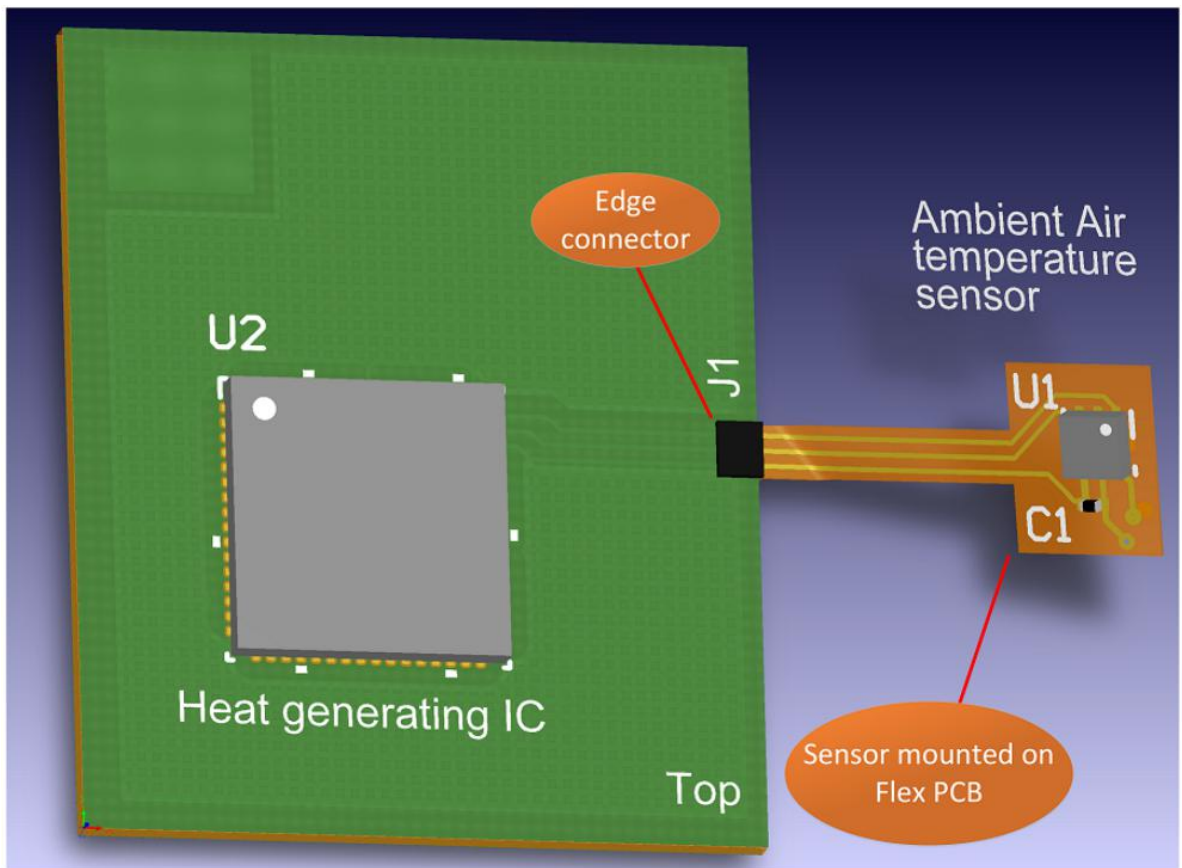


Рис.2.19. Спеціальна гнучка друкована плата сенсора з крайовим роз'ємом

РОЗДІЛ 3 ДОСЛІДЖЕННЯ ЦИФРОВОГО ДАТЧИКА ТЕМПЕРАТУРИ

3.1. DS18B20 цифровий вимірювач температури

DS18B20 це цифровий вимірювач температури, з роздільною здатністю перетворення 9 - 12 розрядів і функцією тривожного сигналу контролю за температурою. Параметри контролю можуть бути задані користувачем та збережені в незалежній пам'яті датчика.

DS18B20 обмінюється даними з мікроконтролером по однопровідній лінії зв'язку, використовуючи протокол інтерфейсу 1-Wire.

Живлення датчик може отримувати безпосередньо від лінії даних без використання зовнішнього джерела. У цьому режимі живлення датчика походить від енергії, запасеної на паразитній ємності.

Діапазон вимірювання температури становить від -55 до +125 °C. Для діапазону від -10 до +85 °C похибка не перевищує 0,5 °C.

Кожна мікросхема DS18B20 має унікальний серійний код довжиною 64 розряду, який дозволяє кільком датчикам підключатися на одну загальну лінію зв'язку. Тобто. через один порт мікроконтролера можна обмінюватись даними з кількома датчиками, розподіленими на значній відстані. Режим дуже зручний для використання в системах екологічного контролю, моніторингу температури в будинках, вузлах обладнання.

3.1.1. Особливості DS18B20

- Для однопровідного інтерфейсу 1-Wire достатньо одного порту зв'язку з контролером.
- Кожен пристрій має унікальний серійний код завдовжки 64 розряди.
- Можливість підключення кількох датчиків через лінію зв'язку.
- Немає потреби у зовнішніх компонентах.
- Можливість отримання живлення безпосередньо від лінії зв'язку. Напруга живлення в межах 3,0 ... 5,5 В.
- Діапазон вимірювання температури -55...+125°C.

- Похибка не перевищує 0,5 °C у діапазоні -10 ... +85 °C.
- Дозвіл перетворення 9...12 біт. Задається користувачем.
- Час виміру, не перевищує 750 мс, при максимально можливій роздільній здатності 12 біт.
- Можливість програмування параметрів тривожного сигналу.
- Тривожний сигнал передає дані адресу датчика, у якого температури вийшла за задані межі.
- Сумісність програмного забезпечення з DS1822.
- Вкрай широкі сфери застосування.

Призначення виводів мікросхеми

SOIC-8	TO-92	СИГНАЛ	ОПИС
5	1	GND	Земля
4	2	DQ	Виведення сигналу даних (вхід/вихід). Вихід типу відкритий колектор інтерфейсу 1-Wire. Також через нього відбувається живлення у режимі "паразитне живлення".
3	3	V _{DD}	Виведення зовнішнього живлення. У режимі «паразитного живлення» має бути підключений до землі.

Оперативна пам'ять містить:

- значення вимірної температури (2 байти);
- верхній та нижній порогові значення спрацьовування тривожного сигналу (Th, Tl);
- регістр конфігурації (1 байт).

Через регістр конфігурації можна встановити роздільну здатність перетворення термодатчика. Дозвіл може бути заданий 9, 10, 11 або 12 біт. Регістр конфігурації та пороги тривожного сигналу містяться в енергонезалежній пам'яті (EEPROM).

У мікросхемі DS18B20 для обміну даними використовується спеціалізований протокол 1-Wire корпорації Dallas. Для лінії зв'язку потрібно слабкий резистор, що підтягує, т.к. всі пристрої фізично підключені до однієї загальної шини і використовують вихід із трьох станів або вихід типу

відкритий стік. У цій системі з однією шиною, мікроконтролер (майстер) визначає наявність пристроїв на шині та обмінюється з ними, використовуючи унікальну адресу для кожного пристрою – 64-розрядний код. Так як кожен термодатчик має унікальний код, кількість пристроїв, підключених до шини, практично ні чим не обмежена.

Інша особливість DS18B20 – працювати без зовнішнього джерела живлення. Живлення відбувається через підтягуючий резистор шини та виведення DQ під час високого рівня шини. Сигнал високого рівня заряджає через виведення DQ внутрішній конденсатор (C_{PP}), енергією якого живиться мікросхема при низькому рівні лінії зв'язку. Цей метод специфікації протоколу 1-Wire називається «паразитне живлення». Нічого не заважає використовувати зовнішнє живлення для DS18B20. Подається на вивід V_{DD} .

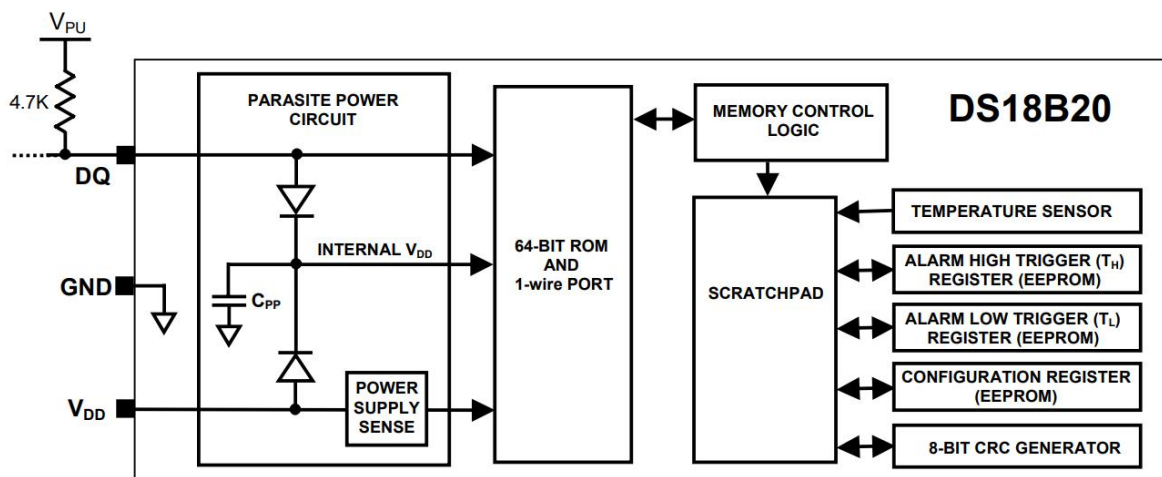


Рис. 3.1. Блок-схема DS18B20

Основна функція DS18B20 – перетворення температури датчика на цифровий код. Дозвіл перетворення визначається 9, 10, 11 або 12 біт. Це відповідає роздільній здатності - 0,5 (1/2) °C, 0,25 (1/4) °C, 0,125 (1/8) °C та 0,0625 (1/16) °C. При включенні живлення стан регістру конфігурації встановлюється на дозвіл 12 біт.

Після включення живлення DS18B20 перебуває у низько-споживаючому стані спокою. Щоб ініціювати вимірювання температури, майстер (мікроконтролер) повинен виконати команду перетворення температури 44h. Після завершення перетворення результат вимірювання температури буде

перебувати в 2 байтах регістра температури, і датчик знову перейде в стан спокою.

Якщо DS18B20 увімкнено за схемою із зовнішнім живленням, то майстер може контролювати стан команди конвертації. Для цього він повинен читати стан лінії (виконувати тимчасовий слот читання), по завершенні команди лінія перейде у високий стан. Під час виконання команди конвертації лінія утримується у низькому стані.

При живленні від заряду паразитної ємності такий спосіб не допустимо, т.к. під час операції перетворення на шині необхідно утримувати високий рівень сигналу живлення датчика.

DS18B20 вимірює температуру у градусах за шкалою Цельсія. Результат виміру представляється як 16-розрядне, знакове число у додатковому коді (рис.). Біт знака (S) дорівнює 0 для позитивних чисел та дорівнює 1 для негативних. При роздільній здатності 12 біт, у регістра температури всі біти значні, тобто мають достовірні значення. Для роздільної здатності 11 біт, не визначений біт 0. Для роздільної здатності 10-бітного не визначені біти 0, 1. При роздільній здатності 9 біт, не достовірне значення мають біти 0, 1 і 2. На рис. показані приклади відповідності цифрових кодів значенню температури.

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
LS Byte	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}
	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
MS Byte	S	S	S	S	S	2^6	2^5	2^4

Рис. 3.2. Формат регістрів температури

TEMPERATURE	DIGITAL OUTPUT (Binary)	DIGITAL OUTPUT (Hex)
+125°C	0000 0111 1101 0000	07D0h
+85°C*	0000 0101 0101 0000	0550h
+25.0625°C	0000 0001 1001 0001	0191h
+10.125°C	0000 0000 1010 0010	00A2h
+0.5°C	0000 0000 0000 1000	0008h
0°C	0000 0000 0000 0000	0000h
-0.5°C	1111 1111 1111 1000	FFF8h
-10.125°C	1111 1111 0101 1110	FF5Eh
-25.0625°C	1111 1110 0110 1111	FE6Fh
-55°C	1111 1100 1001 0000	FC90h

*The power-on reset value of the temperature register is +85°C

Рис. 3.3. Відповідність даних та температури

При позитивному значенні ($S=0$) код перевести в десятковий і помножити на $0,0625^{\circ}\text{C}$.

При негативному значенні ($S=1$) спочатку необхідно перевести додатковий код у прямий. Для цього треба інвертувати кожен розряд двійкового коду та додати 1. Потім перевести в десятковий і помножити на $0,0625^{\circ}\text{C}$.

3.1.2. Режим – передача сигналу тривоги

Після виконання команди перетворення температури вимірне значення порівнюється з верхнім і нижнім порогами з регістрів T_h , T_l (формат на малюнку 3). Це байтові значення, знакові, додатковому коді, $S=0$ означає, що число позитивне, а $S=1$ – негативне. Зберігаються граничні значення в енергонезалежній пам'яті (EEPROM). T_h і T_l доступні для читання та запису через байти 2, 3 оперативної пам'яті.

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
S	2^6	2^5	2^5	2^5	2^2	2^1	2^0

Рис. 3.4. Формат регістрів T_h і T_l

Через різну довжину регістрів TH, TL і температури, вони порівнюються тільки з бітами 11 по 4 регістра температури. Якщо значення вимірної температури перевищує TH або нижче, ніж TL, формується ознака аварії в DS18B20. Ознака перезаписується з кожним виміром температури, і якщо температура повертається в задані межі, вона скидається.

Провідний пристрій може перевірити стан ознаки аварії за допомогою команди «пошук тривожного сигналу» ECh. Будь-який датчик із активною ознакою відповість на команду пошуку. Таким чином майстер точно визначить, який DS18B20 виробляє сигнал тривоги. Після зміни значень регістрів TH і TL тільки наступне перетворення температури сформує достовірну ознаку тривоги.

3.1.3. Живлення термодатчика DS18B20

DS18B20 може бути підключений до зовнішнього джерела живлення через вивід Vdd або працювати за схемою паразитного живлення, без додаткового джерела. Режим «паразитного живлення» дуже зручний для додатків із віддаленими датчиками температури. На рис наведено схему, яка використовує "паразитне живлення". За високого рівня напруги шини, енергія накопичується на "паразитному" конденсаторі C_{pp} через виведення DQ. При низькому рівні шини енергія конденсатора забезпечує живлення датчика. У режимі "паразитного живлення" висновок Vdd обов'язково має бути з'єднаний із загальним дротом схеми (землею).

У режимі паразитного живлення, сигнал шини та заряд C_{pp} здатні забезпечити необхідний струм для роботи DS18B20, за умови, що часові параметри та вимоги до рівнів сигналів відповідатимуть електричним характеристикам.

Однак коли DS18B20 виконує операцію перетворення температури або копіювання даних пам'яті в EEPROM, струм може досягати величини 1,5 мА. Такий струм може спричинити зниження напруги живлення пристрою до неприпустимого значення. Струму підтягуючого резистора та енергії, запасеної на C_{pp}, недостатньо для живлення в цих двох режимах. Для

забезпечення достатнього живлення пристрою необхідно забезпечити потужну підтяжку шини до високого рівня в той час, коли відбувається перетворення температури або копіювання даних пам'яті в EEPROM. Це можна зробити за допомогою транзистора MOSFET, як показано на схемі (рис.).

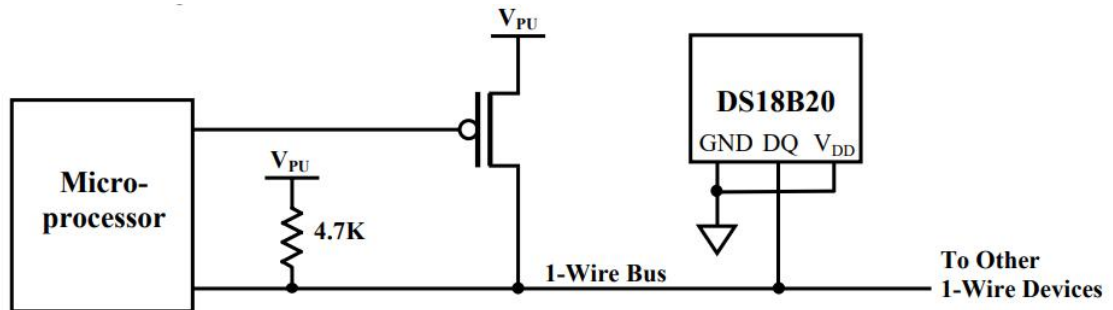


Рис. 3.5. Режим «паразитного живлення» DS18B20 під час перетворення температури

Шина даних має бути підключена до потужного живлення:

- протягом 10 мкс після команд конвертування 44h та копіювання пам'яті 48h;
- протягом часу перетворення (t_{conv}) та передачі даних (не менше $t_{WR} = 10\text{мс}$).

Жодних інших операцій у цей час на шині допускати не можна.

Як правило, у сучасних мікроконтролерів вихідного струму високого рівня цілком достатньо для живлення DS18B20. Тоді у MOSFET транзисторі потреби немає.

Для живлення DS18B20 можна використовувати звичайний метод – підключення зовнішнього живлення через вивід VDD (рис.). Очевидні переваги цього методу у відсутності необхідності в MOSFET транзисторі і в тому, що під час перетворення шина залишається вільною і може використовуватися з іншою метою.

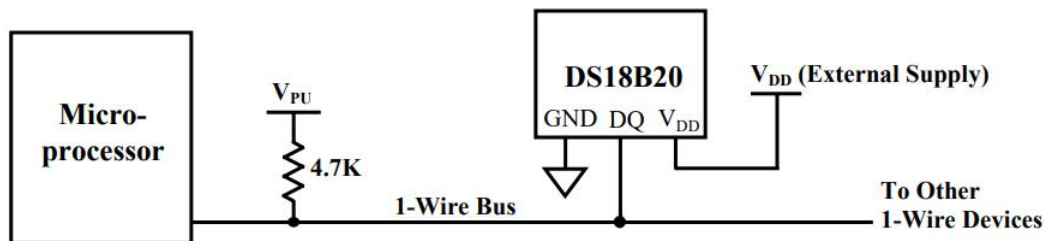


Рис. 3.6. Режим живлення із зовнішнім джерелом

Використання «паразитного живлення» настійно не рекомендується при температурі понад 100 °С через великий струм витoku. Енергії, запасеної на C_{pp} , може не вистачити для нормальної роботи. При вимірі такої температури слід використовувати зовнішнє живлення.

В таких випадках використовується наступна схема підключення DS18B20 (рис. 3.7)

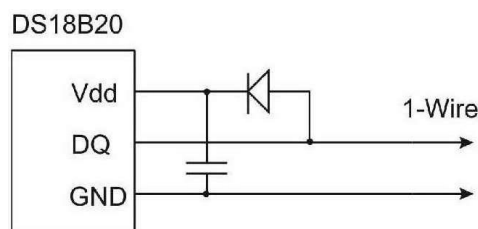


Рис. 3.7. Схема підключення DS18B20

У цій схемі термодатчик працює у режимі із зовнішнім живленням, яке запасється на додатковому конденсаторі через діод.

У деяких випадках майстер не може знати, який режим живлення використовується. Ця інформація може знадобитися, щоб правильно керувати транзистором потужної підтяжки рівня. Для визначення режиму живлення, провідний пристрій може виконати команду «пропустити ROM» CCh , потім команду «читання електроживлення» $B4h$, а потім виконати слот читання. Під час читання термодатчик з «паразитним живленням» відповідь станом 0. А датчик із зовнішнім живленням це високий рівень.

3.1.4. 64-розрядний серійний код пристрою

Кожен датчик DS18B20 містить у ROM унікальний код, 64 біта завдовжки (рис. 6). Молодші вісім біт містять код сімейства мікросхем

DS18B20 28h. Середній блок із 48 біт це унікальний серійний номер пристрою. Старші вісім біт це циклічний код (CRC) для всіх попередніх 56 біт.

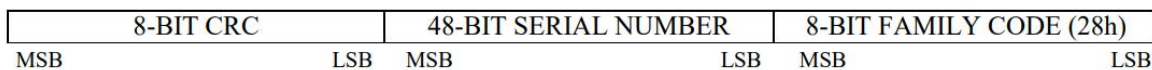


Рис. 3.8. 64-бітний ROM код

3.1.5. Пам'ять датчика

Організація пам'яті DS18B20 показана малюнку 7. Вся пам'ять включає оперативну (SRAM) і енергонезалежну (EEPROM) пам'ять. У EEPROM зберігаються регістри T_H, T_L та регістр конфігурації. Якщо функція тривожного сигналу не використовується, регістри T_H і T_L можуть використовуватися як регістри загального призначення.

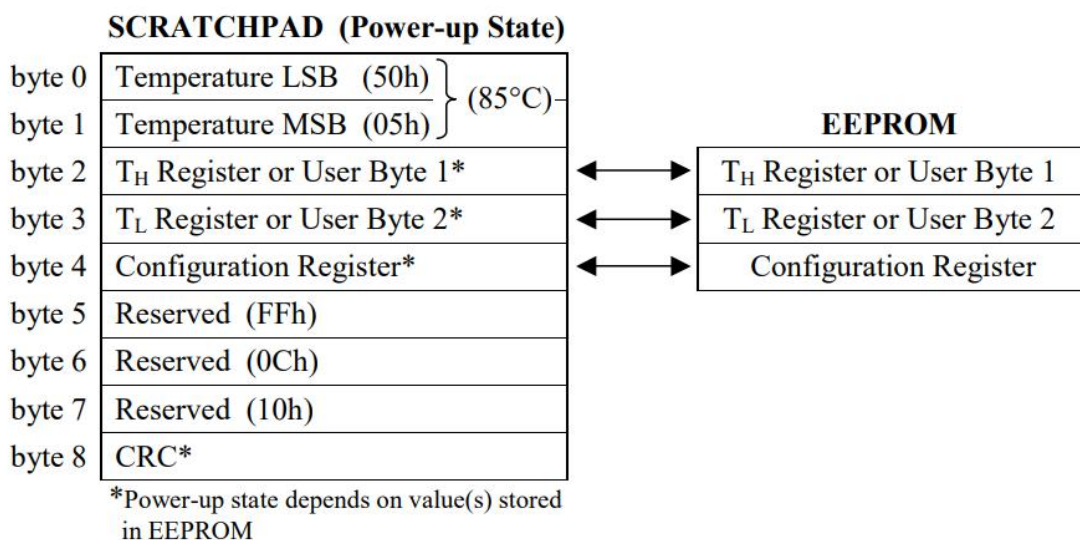


Рис. 3.9. Карта пам'яті DS18B20

У байтах з адресами 0 та 1 зберігаються молодший та старший байти регістру вимірної температури. Ці байти доступні лише для читання. 2-й і 3-й байти - T_H і T_L регістри. Байт 4 – регістр конфігурації. Байти 5, 6, 7 зарезервовані, не можуть бути записані і, читаючи, завжди повертають 1. Байт 8 доступний лише для читання. Він містить циклічний код (CRC) перших восьми байтів.

Запис даних у байти 2, 3 і 4 відбувається командою «запис пам'яті» 4Eh. Дані повинні передаватися, починаючи з молодшого біта байта 2. Для перевірки запису даних можна прочитати пам'ять командою «читання пам'яті» BEh. При читанні дані передаються по шині, послідовно починаючи з молодшого біта байта 0. Запис даних TH, TL і регістра конфігурації в EEPROM відбувається за командою «копіювання пам'яті» 48h.

При включенні живлення дані з енергонезалежної пам'яті EEPROM перевантажуються в оперативну пам'ять (SRAM). Перезавантаження даних з EEPROM можна також зробити командою ПЕРЕЗАВАНТАЖЕННЯ E 2 [B8h]. Майстер повинен контролювати стан шини, щоб визначити завершення перезавантаження. Слот читання низького рівня означає, що перезавантаження ще не закінчилося. Після завершення перезавантаження DS18B20 передає слот читання 1.

3.1.6. Регістр конфігурації термодатчика

Байт 4 пам'яті це регістр конфігурації (рис. 3.10). Бітами R0, R1 можна встановити роздільну здатність перетворення (рис. 3.11). При включенні живлення стан бітів R0, R1 = 11, що відповідає роздільній здатності 12 біт. Потрібно пам'ятати, що існує пряма залежність часу перетворення від роздільної здатності. Біти 7 та 0...4 зарезервовані, не можуть використовуватися, при читанні повертають 1.

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	R1	R0	1	1	1	1	1

Рис. 3.10. Регістр конфігурації

R1	R0	Resolution	Max Conversion Time	
0	0	9-bit	93.75 ms	($t_{CONV}/8$)
0	1	10-bit	187.5 ms	($t_{CONV}/4$)
1	0	11-bit	375 ms	($t_{CONV}/2$)
1	1	12-bit	750 ms	(t_{CONV})

Рис. 3.11. Конфігурація розмодільної здатності перетворювача

3.1.7. Генерація циклічного коду (CRC)

Байти циклічного коду (CRC) розташовані в 64-бітовому ROM код і в дев'ятому байті пам'яті SRAM. Циклічний код з ROM обчислюється для 56 бітів коду ROM і розташовується в старшому байті ROM. Циклічний код із SRAM обчислюється з байтів 0...7 SRAM. Циклічний код дозволяє контролювати правильність читання даних із DS18B20. Майстер обчислює циклічний код для даних і порівнює з прийнятим кодом. З цього приймається рішення про коректність даних.

Утворюючий поліном циклічного коду виглядає так:

$$CRC = X^8 + X^5 + X^4 + 1$$

Майстер може обчислити циклічний код використовуючи генератор полінома, за схемою рис. Він складається з регістру зсуву і логічних елементів типу «виключаюче АБО». Регістр зсуву спочатку перебуває у стані 0. Біти надходять у регістр зсуву, починаючи з молодшого біта, коду з ROM чи SRAM, один біт одного такт зсуву. Після зсуву 56-го біта ROM або старшого біта 7-го байта SRAM, в регістрі зсуву буде обчислений циклічний код. Якщо зрушити в генератор 8 біт ROM або SRAM, прийняті з DS18B20, то у разі правильних даних регістр зсуву буде містити всі 0.

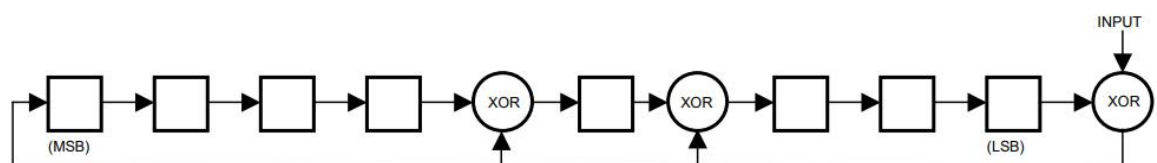


Рис. 3.12. Генератор CRC

3.1.8. Однопровідний інтерфейс 1-Wire

Система з шиною 1-Wire складається з одного провідного пристрою (майстер), який управляє одним або декількома веденими пристроями (слейв). DS18B20 може бути лише веденим. Система, в якій один ведений пристрій, називається одноточковою. Система з кількома веденими – багатоточковою. Усі команди та дані обміну передаються по шині молодшим бітом уперед. У подальшій інформації про інтерфейс 1-Wire виділено три розділи: апаратна конфігурація, послідовність операцій та сигнали (типи та часові вимоги).

Апаратна конфігурація

Інтерфейс 1-Wire має одну лінію зв'язку. Кожен пристрій (провідний чи ведений) підключено до шини даних портом з виходом типу відкритий колектор або з трьома станами. Така конфігурація дозволяє кожному пристрою системи не займати лінію зв'язку, коли він не активний, і тримати шину вільною для інших пристроїв. У мікросхемі DS18B20 вихід (DQ) – відкритий стік. Його еквівалентна схема наведена на рис. . Шина 1-Wire вимагає застосування зовнішнього резистора, що підтягує, опором приблизно 5 кОм, для забезпечення високого рівня сигналу при неактивному стані пристроїв. Якщо операція має бути припинена, шина має бути встановлена в неактивний стан до наступної операції. Шина може бути в стані високого рівня як завгодно довгий час.

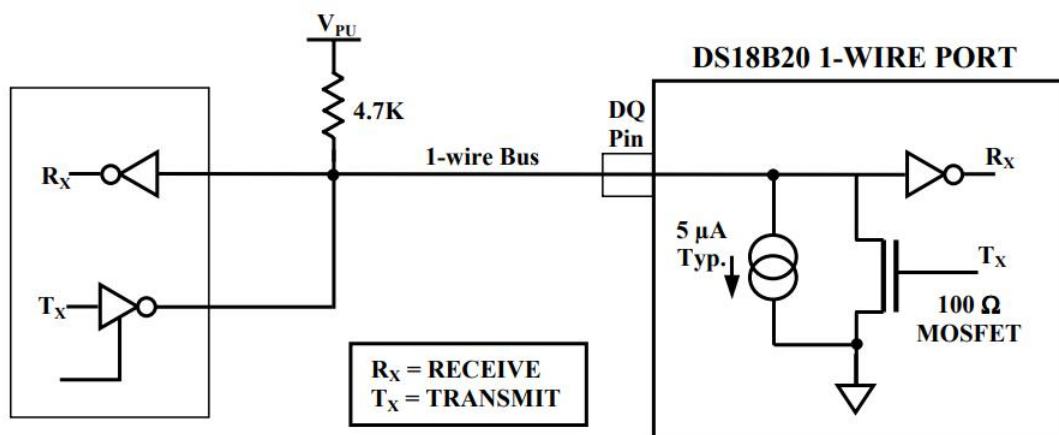


Рис. 3.13. Апаратна конфігурація 1-Wire

Послідовність операцій

Черговість операцій для доступу до термодатчика DS18B20 має такий вигляд.

Ініціалізація.

Команда ROM (необхідна будь-якого обміну даними).

Функціональна команда (необхідна будь-якого обміну даними).

Така послідовність повинна суворо дотримуватися. В іншому випадку DS18B20 не реагуватиме на команди. Винятком є команди ПОШУК ПЗУ [код F0h] та ПОШУК АВАРІЇ [код ECh]. Після формування цих двох команд провідний пристрій (майстер) повинен повернутися до першого кроку (ініціалізація).

Ініціалізація

Обмін по шині завжди починається з операції ІНІЦІАЛІЗАЦІЇ. Для ініціалізації провідний пристрій виробляє імпульс скидання, за ним повинен наслідувати імпульс присутності від веденого пристрою. Імпульс присутності повідомляє провідного пристрою, що ведений пристрій присутній в системі і готовий до виконання операції. Часові параметри імпульсів скидання та присутності описані в розділі СИГНАЛИ 1-Wire .

Команди ROM-кодів

Після того, як провідний пристрій отримує імпульс присутності, він може оперувати командами ROM. Це команди для операцій із 64-бітними індивідуальними кодами кожного веденого пристрою. Вони дозволяють провідному пристрою вибрати конкретний ведений пристрій серед багатьох інших. Також, використовуючи ці команди, можна дізнатися, скільки ведених пристроїв підключено до шини, їх типи, виділити пристрої може тривоги. Існує 5 команд ROM, завдовжки 8 біт кожна. Перед виконанням функціональних команд DS18B20 провідний пристрій повинен надіслати команду ROM.

- Пошук ROM [F0h]

Після включення живлення, провідний пристрій повинен вважати ROM коди всіх керованих пристроїв, підключених до шини. Це дозволить

визначити кількість ведених пристроїв та їх типи. Провідний пристрій вивчає коди ROM через процес ідентифікації кодів кожного пристрою на шині. Воно має виконати команду пошуку ROM стільки разів, скільки необхідно для ідентифікації всіх керованих пристроїв. При одному веденому пристрої у системі простіше використовувати команду ЧИТАННЯ ROM. Після пошуку ROM операції на шині повинні знову початися з ініціалізації.

- Читання ROM [33h]

Команда застосовується в однокерованих системах з одним веденим пристроєм. Вона дає можливість провідному пристрою прочитати 64-бітний ROM код без використання команди ПОШУК ROM. Застосування команди ЧИТАННЯ ROM у багатокерованій системі призведе до конфліктів даних між керованими пристроями.

- Збіг ROM [55h]

Команда Збіг ROM, після якої повинен слідувати 64-бітний код ROM, дозволяє майстру звертатися до конкретного веденого пристрою. Тільки один ведений пристрій, код якого збігається з переданим кодом, прореагує на функціональні команди. Інші керовані пристрої будуть неактивними до наступного імпульсу скидання.

- Перепустка ROM [CCh]

Команда дозволяє провідному пристрою звертатися до всіх пристроїв шин одночасно, без використання ROM кодів. Наприклад, можна запустити на всіх пристроях операцію перетворення температури, виконавши команду пропуск rom, а потім конвертація температури. Команда читання пам'яті [BEh] може йти за командою пропуск ROM лише за одному веденому пристрої, підключеним до лінії зв'язку. Така послідовність команд значно заощаджує час обміну з датчиками. Особливо ефективна при використанні в системі одного веденого пристрою.

- Пошук сигналу тривоги [ECh]

Команда діє ідентично команді ПОШУК ROM. Відрізняється тим, що на неї дадуть відповідь тільки ведені пристрої в стані тривоги. Команда

дозволяє веденому пристрою визначити, які термодатчики перебувають у стані тривоги після останнього перетворення температури. Після кожного пошуку тривоги необхідно повертатися на ініціалізацію.

Група функціональних команд

Після виконання ROM команди для вибору DS18B20 з потрібним кодом, провідний пристрій може надсилати функціональні команди датчика. Вони дозволяють записати та прочитати дані з оперативної пам'яті DS18B20, ініціювати перетворення температури та визначити режим живлення.

- **Перетворення температури [44h]**

Команда ініціалізує одне перетворення температури. Після неї значення температури виявляється в 2-байтному регістрі температури оперативної пам'яті і датчик перетворюється на неактивне, низькоспоживаючий стан. Для режиму "паразитного живлення" необхідно виконати вимоги щодо підключення шини до джерела живлення на час перетворення з розділу ЖИВЛЕННЯ DS18B20. У режимі зовнішнього живлення провідний пристрій може визначити стан операції перетворення за допомогою слота читання. При не закінченій операції стан шини буде низьким, після завершення перетворення на шині буде високий рівень. За "паразитного живлення" такий спосіб відстеження процесу перетворення не можливий.

- **Запис пам'яті [4Eh]**

Команда дозволяє завантажити 3 байти в оперативну пам'ять датчика. Перший байт записується в регістр Th (2 байт пам'яті), другий байт Th (байт 3 пам'яті) і третій байт в регістр конфігурації (байт 4). Провідний пристрій передає дані починаючи з молодшого біта. Всі три байти необхідно записати до того, як провідний пристрій сформує сигнал скидання.

- **Читання пам'яті [BEh]**

Команда використовується для читання пам'яті пристрою. Передача даних відбувається починаючи з молодшого біта байта 0 пам'яті, і продовжується до тих пір, поки всі 9 байтів будуть раховані. Якщо потрібна

лише частина даних, пристрій може перервати передачу, сформувавши імпульс скидання.

- Копіювання пам'яті [48h]

Команда копіює з оперативної пам'яті в EEPROM три байти: Th, Tl та регістр конфігурації. Для режиму паразитного живлення необхідно виконати вимоги щодо підключення шини до джерела живлення на час виконання команди з розділу 3.1.3.

- Перезавантаження EEPROM [B8h]

Команда перезавантажує значення регістрів Th, Tl та регістра конфігурації з EEPROM в оперативну пам'ять. Після надсилання команди перезавантаження, провідний пристрій може виконати слот читання, і DS18B20 повідомить стан перезавантаження. Передача 0 означатиме, що операція ще виконується, 1 – операція завершена. Операція перезавантаження автоматично відбувається при увімкненні живлення. Тому в оперативній пам'яті містяться достовірні дані одразу після подачі живлення.

- Читання режиму живлення [B4]

Команда дозволяє визначити режим живлення будь-якого датчика. Після передачі команди провідний пристрій виконує тайм-слот читання. Термодатчик із "паразитним живленням" встановить вихід у низький стан, тоді як датчик із зовнішнім живленням триматиме шину у високому стані. Детальніше в розділі живлення DS18B20.

Таблиця 1. Функціональні команди DS18B20.

КОМАНДА	ОПИС	КОД	ОПЕРАЦІЇ НА ШИНІ	ПРИМІТКА
КОМАНДА КОНВЕРТУВАННЯ ТЕМПЕРАТУРИ				
Вимірювання температури	Ініціалізує вимірювання температури	44h	DS18B20 передає майстру стан операції перетворення температури	1
КОМАНДИ РОБОТИ З ПАМ'ЯТТЮ				
Читання пам'яті	Читає всю оперативну пам'ять, включаючи	BEh	DS18B20 передає майстру до 9 байт	2

	циклічний код CRC			
Запис пам'яті	Записує в оперативну пам'ять байти 2, 3 та 4 (TH, TL та регістр конфігурації)	4Eh	Майстер передає 3 байти на DS18B20.	3
Копіювання пам'яті	Копіює TH, TL, і регістр конфігурації з оперативної пам'яті EEPROM	48h		1
Перезавантаження	Перезавантажує TH, TL, і регістр конфігурації EEPROM в оперативну пам'ять.	B8h	DS18B20 передає стан перезавантаження майстру	
Читання режиму живлення	Повідомляє майстра про режим живлення DS18B20.	B4h	DS18B20 передає майстру режим живлення	

Примітки.

1. Для режиму «паразитного живлення» провідний пристрій повинен мати можливість підключати шину 1-Wire до шини живлення під час виконання перетворення температури та копіювання пам'яті.
2. Провідний пристрій може перервати передачу, скинувши систему.
3. Усі три байти необхідно записати до команди скидання.

Інтерфейс 1-Wire

Для обміну даними DS18B20 використовує протокол 1-Wire інтерфейсу, що забезпечує контроль цілісності даних. Цей протокол визначає сигнали:

- імпульс скидання,
- імпульс присутності,
- запис біта зі значенням 0,
- запис біта зі значенням 1,
- читання біта зі значенням 0,
- читання біта зі значенням 1.

Всі ці сигнали, крім імпульсу присутності, формує провідний пристрій.

Ініціалізація – імпульси скидання та присутності

Будь-які комунікаційні операції DS18B20 починаються з послідовності ініціалізації, яка складається з імпульсу скидання від провідного пристрою веденому, і імпульсу відповідності присутності з DS18B20. Цей процес показаний на рис. . Термодатчик посилає імпульс присутності у відповідь на імпульс скидання, щоб повідомити провідний пристрій, що він підключений до шини і готовий до використання.

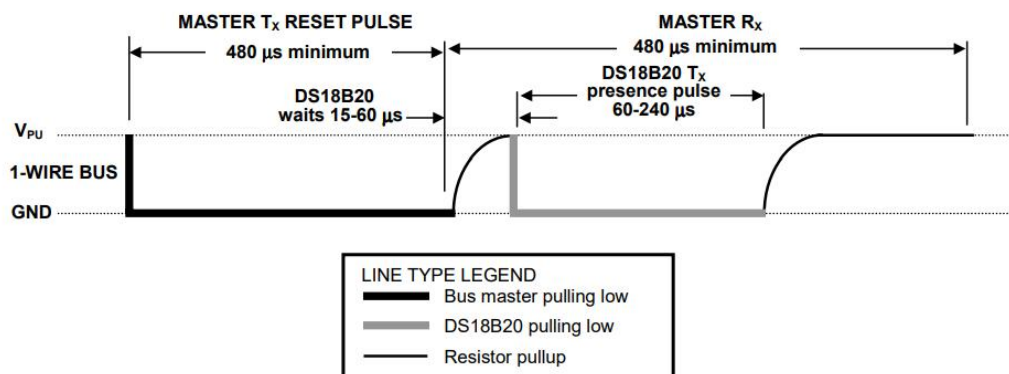


Рис. 3.14. Часові послідовності ініціалізації

Під час послідовності ініціалізації провідний пристрій передає імпульс скидання (Tx), формуючи на шині сигнал низького рівня протягом не менше 480 мкс. Далі провідний пристрій звільняє шину і переходить в режим прийому (Rx). Коли шина звільняється, підтягується до високого логічного рівня резистором 5 кОм. Датчик виділяє позитивний фронт, чекає 15-60 мкс та передає імпульс присутності, утримуючи низький рівень лінії на час 60-240 мкс.

Тимчасові слоти читання та запису

Обмін даними по шині 1-Wire відбувається тимчасовими слотами (тайм-слотами). Один часовий слот передає один біт інформації.

Тимчасові слоти запису

Протокол визначає два типи тайм-слотів запису даних DS18B20: для запису значення 1 і запису значення 0. Тривалість слота запису - не менше 60 мкс з паузою на відновлення між слотами 1,0 мкс, як мінімум. Ініціюється будь-який слот запису негативним фронтом сигналу шини (рис. 14).

Для формування слота запису 1 після переключення шини в низький стан, провідний пристрій повинен звільнити шину на час 15 мкс. Підтягуючий резистор 5 ком створить на шині напруга високого рівня.

Для формування слота запису 0, після переведення шини в низький стан, провідний пристрій повинен продовжувати утримувати шину в низькому стані протягом усього часу слота (як мінімум 60 мкс).

DS18B20 перевіряє стан сигналу у відрізку часу між 15 і 60 мкс, відраховуючи його від початку слота запису. Стан шини на цьому відрізку відповідає значенню біта для запису датчика.

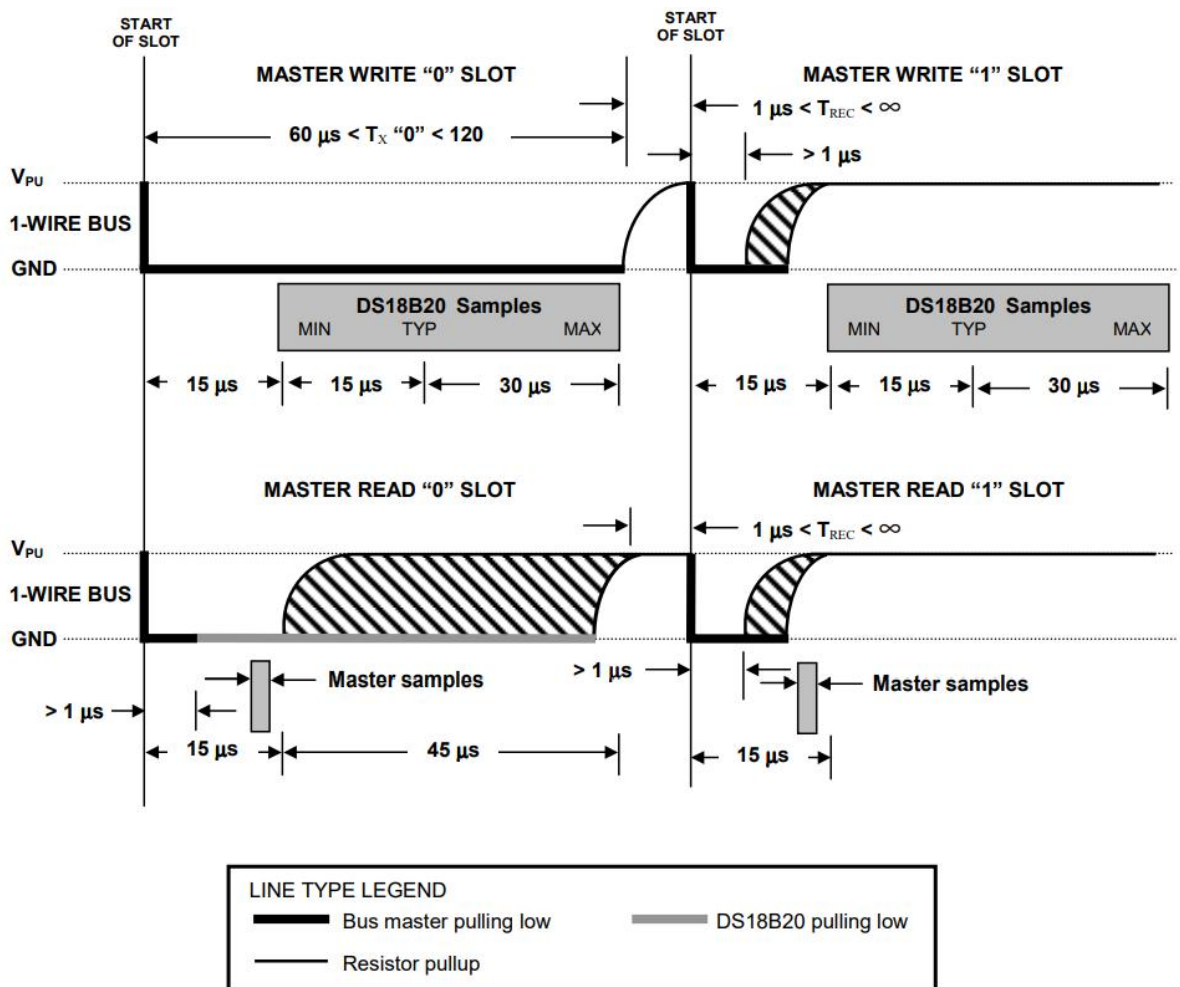


Рис. 3.15. Часові параметри слотів

Термодатчик DS18B20 не лише приймає, а й передає дані. Провідному пристрою необхідно отримувати дані відразу після формування команд читання пам'яті код [BEh] або читання режиму живлення [код B4h]. Крім

того, провідний пристрій формує слоти читання після посилки команд перетворення температури [44h] та перезавантаження [B8h], для того щоб визначити стан операції.

Тривалість слота читання, як і слота запису, повинна бути не менше 60 мкс з паузою відновлення між слотами 1 мкс, як мінімум. Ініціюється будь-який слот читання негативним фронтом сигналу шини (рис.).

Після того, як провідний пристрій ініціалізував слот читання, DS18B20 передає біт даних. Для передачі 1 датчик залишає шину вільною (у високому стані), а передачі 0 – формує на шині низький рівень.

При передачі 0 DS18B20 повинен звільнити шину в кінці слота. Підтягуючий резистор сформує на ній високий рівень. Вихідні дані DS18B20 достовірні протягом 15 мкс від початку слота читання.

На рис. показано, що загальна сума часових інтервалів слота читання T_{init} , T_{RC} та T_{SAMPLE} повинна бути не більше 15 мкс.

Рис. показано, що для максимальної надійності прийому даних необхідно зменшити T_{init} та T_{RC} та читати стан шини в кінці відрізка 15 мкс.

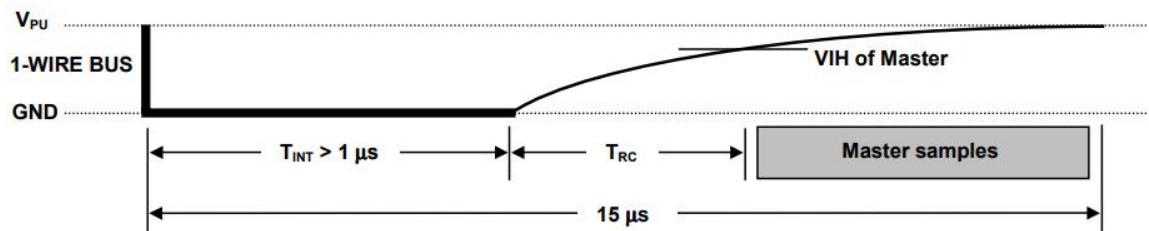


Рис. 3.16. Слот читання в деталях

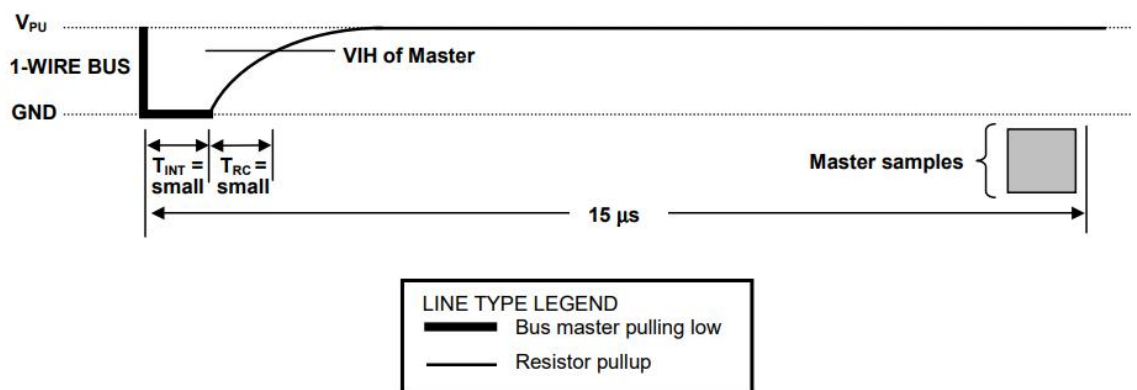


Рис. 3.18. Рекомендованні параметри слота читання

3.1.9. Гранично допустимі параметри DS18B20

В наступних таблицях вказано граничні величини параметрів. Перевищення цих параметрів є неприпустимим. Експлуатація тривалий час із граничними значеннями параметрів може зменшити надійність пристрою.

Напруга на будь-якому піні датчика відносно землі	- 0,5...6
Робоча температура	- 55 ... + 125 °С
Температура зберігання	Електричні характеристики постійного струму (DC характеристики).
Температура паяння	Див. J-STD-020A

Електричні характеристики постійного струму (-55...+125°C, VDD = 3,0 ... 5,5 В).

ПАРАМЕТР	ПОЗНАЧЕННЯ	УМОВИ	МІН.	ТИП.	МАКС.	ОД. ВИМ.	ПРИМІТКА
Напруга живлення	V _{DD}	Зовнішнє живлення	+3.0		+5.5	В	1
Підтягуюча напруга	V _{PU}	Паразитне живлення	+3.0		+5.5	В	1,2
		Зовнішнє живлення	+3.0		V _{DD}		
Похибка вимірювання температури	t _{ERR}	-10°C - +85°C			±0.5	°С	3
		-55°C - +125°C			±2		
Вхідний логічний низький рівень	V _{IL}		-0.3		+0.8	В	1,4,5
Вхідний логічний високий рівень	V _{IH}	Зовнішнє живлення	+2.2		Нижче 5.5 або V _{DD} +	В	1, 6
		Паразитне живлення	+3.0				

					0.3		
Текущий струм	I_L	$V_{IO} = 0,4$ В	4.0			мА	1
Струм спокою	I_{DDs}			750	1000	нА	7,8
Робочий струм	I_{DD}	$V_{DD} = 5В$		1	1.5	мА	9
Струм витоку	I_{DQ}			5		мкА	10
Дрейф температури				± 0.2		$^{\circ}C$	11

Примітки:

1. Значення всіх напруг зазначені щодо землі.
2. Передбачається, що пристрій, напруга, що підтягує, ідеальне, тому напруга підтяжки дорівнює $V_{ри}$.
3. Дивись типові робочі характеристики (рис. 17).
4. Напруга низького рівня вказана для струму виходу 4 мА.
5. Щоб гарантувати напругу імпульсу присутності не вище напруги паразитного живлення, V_{ilmax} може бути знижено на 0,5 В.
6. Напруга високого рівня вказана для струму виходу 1 мА.
7. Струм спокою визначений для $70^{\circ}C$.
8. Щоб мінімізувати I_{DDs} , напруга на виводі DQ має бути в межах $GND \leq DQ \leq GND + 0,3$ або $VDD - 0,3 В \leq DQ \leq VDD$.
9. Робочий струм вимірюється під час операції перетворення температури.
10. DQ висновок у високому стані.
11. Дрейф даних визначений випробуваннями протягом 1000 годин при $125^{\circ}C$, $VDD = 5,5 В$.

Електричні характеристики EEPROM змінного струму (-55 ... + 125 ° C, $V_{DD} = 3,0 ... 5,5 В$).

ПАРАМЕТР	ПОЗНАЧЕННЯ	УМОВИ	МІН.	ТИП.	МАКС.	ОД. ВИМ.
Час циклу запису	t_{wr}			2	10	мс

Число записів	N_{EEWR}	-55°C - 50000 +55°C			цикл
Час зберігання	t_{EEDR}	-55°C - 10 +55°C			років

**Електричні характеристики змінного струму
(-55 ... + 125 ° C, $V_{DD} = 3,0 ... 5,5$ В).**

ПАРАМЕТРІВ	ПОЗНАЧЕННЯ	УМОВИ	МІН.	ТИП	МАКС	ОД. ВИМ.
Час перетворення температури	$t_{конв}$	роздільна здатність 9 біт			93.75	мс
		роздільна здатність 10 біт			187.5	мс
		роздільна здатність 11 біт			375	мс
		роздільна здатність 12 біт			750	мс
Час підключення до потужного живлення	t_{SPON}	Посилання команди конвертації температури			10	мкс
Час слоту	t_{SLOT}		60		120	мкс
Час відновлення	t_{REC}		1			мкс
Час запису 0	$t_{низкий0}$		60		120	мкс
Час запису 1	t_{LOW1}		1		15	мкс
Час читання даних	$t_{рДВ}$				15	мкс
Час високого рівня скидання	t_{RSTH}		480			мкс
Час низького	t_{RSTL}		480			мкс

рівня скидання						
Час високого рівня присутності	t_{PDHIGH}		15		60	мкс
Час низького рівня присутності	t_{PDLOW}		60		240	мкс
Місткість	$C_{IN/OUT}$				25	пкF

3.2. Моделювання та дослідження датчика DS18B20

Весь контроль температури та реєстрація проводилися за допомогою комп'ютера Raspberry Pi з платою розширення UniPi. Ця плата дозволяє використовувати 1-провідні цифрові датчики температури DS18B20 і має цифрові виходи та входи, готові для використання з іншим обладнанням. Датчик температури має вихідну роздільну здатність $0,05^{\circ}\text{C}$ і точність $0,5^{\circ}\text{C}$. Він був відкалібрований за допомогою калібровача температури Omega CL26 з точністю $0,1^{\circ}\text{C}$.

Для контролю вихідної потужності нагрівача використовувався контроль шпаруватості. В основі цього методу ШІМ з періодом 10 с. За допомогою цього методу потужність можна контролювати від 0 до 100%. Відповідно до необхідної потужності нагрівач встановлюється на повну потужність протягом певної частини довжини циклу. Повна потужність представлена повним часом 10 с, 50% 5 с, 11% 1,1 с і так далі. Для керування нагрівачем використовувалося твердотільне реле переходу через нуль, яке вмикається цифровим виходом плати UniPi.

Алгоритм дискретного PID керування використовувався як метод контролю температури. Алгоритм керування реалізовано на мові Python 3.2, що спілкується за допомогою API evok, який використовується для зв'язку між Raspberry Pi і платою UniPi. Програма керування також записує температуру кожні 5 секунд. Об'єктно-орієнтоване програмування

використовувалося для створення повної керуючої програми. Ідентифікація системи була потрібна для розробки належного контролю температури. Це було зроблено за допомогою нагрівача у ванні з нагріванням реакторів на повну потужність. Ступінчасту реакцію системи було отримано таким чином шляхом реєстрації підвищення температури у ванні кожні 5 с. Цю систему можна описати як систему з одним накопичувачем енергії, отже, її можна апроксимувати безперервною передаточною функцією першого порядку з затримкою у перетворенні Лапласа.

$$G_S(s) = \frac{K}{T_S + 1} \cdot e^{-T_d s}$$

Ідентифікацію системи виконано в Matlab 2014a. Система була апроксимована передаточною функцією системи першого порядку з затримкою. Затримка становила 100 с. Передатна функція G_S міститься в наступному рівнянні

$$G_S(s) = \frac{46}{13200s + 1} \cdot e^{-100s}$$

Ступінчаста характеристика системи з апроксимованою системою представлена на рисунку 3.19.

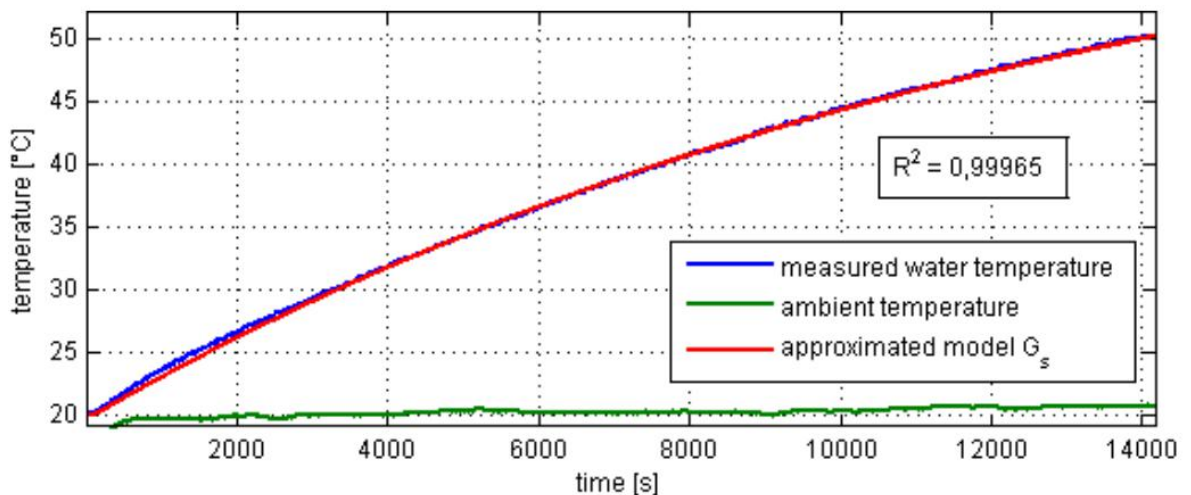


Рис. 3.19. Ідентифікація системи

З метою моделювання ідентифікована неперервна система була дискретизована за допомогою Z-перетворення в програмному забезпеченні

Matlab 2014a з використанням функції `c2d`. Час вибірки було встановлено на 10 секунд. Функція дискретного часу системи в наступному рівнянні

$$G_S(z) = \frac{0.03484}{z - 0.9992}$$

Моделювання проводилося в програмному забезпеченні Matlab 2014a Simulink. Блок-схему можна побачити на рис. 3.20. Блоки округлення сигналу були додані для представлення фактичних значень, що надходять від датчиків температури. Додано насичення контролера для імітації обмеження потужності нагрівача.

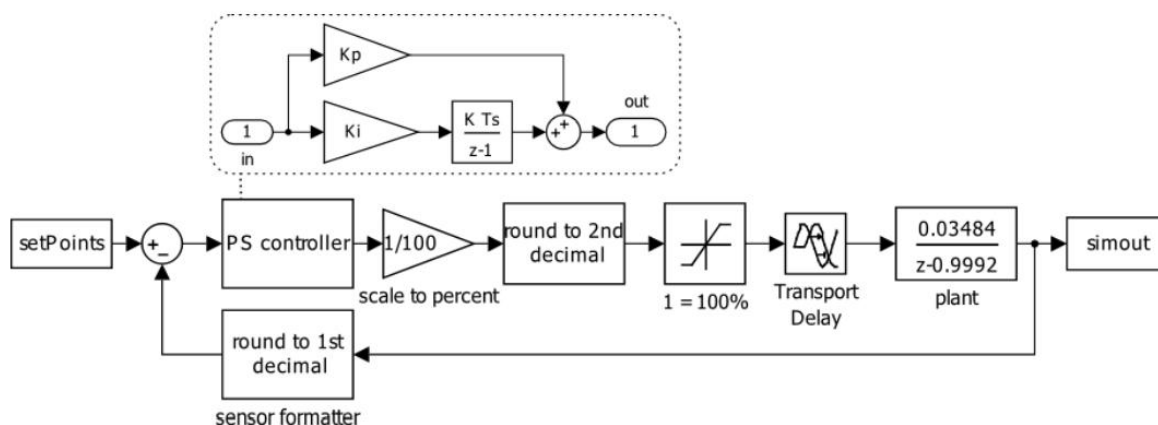


Рис. 3.20. Блок-схема вимірювання температури

Для першої оцінки посилення K_p та K_i використовувалася функція PIDtool у Matlab. Було виявлено, що константи підсилення пропорційності та інтеграції мають максимальне перевищення на $0,2^\circ\text{C}$ на кроці 20°C (збільшення від поточної температури). Після початкового моделювання посилення K_p було встановлено на 24. Щоб відповідати критеріям постійного перевищення для кожного розміру кроку, значення посилення K_i були змодельовані для кроків $2,5\text{--}22,5^\circ\text{C}$ з кроком $2,5^\circ\text{C}$. Приблизну функцію підсилення K_i для кроків температури показано на рисунку 3.21. Ця функція використовується для кожного розміру кроку в алгоритмі керування, щоб знайти та встановити коефіцієнт підсилення K_i .

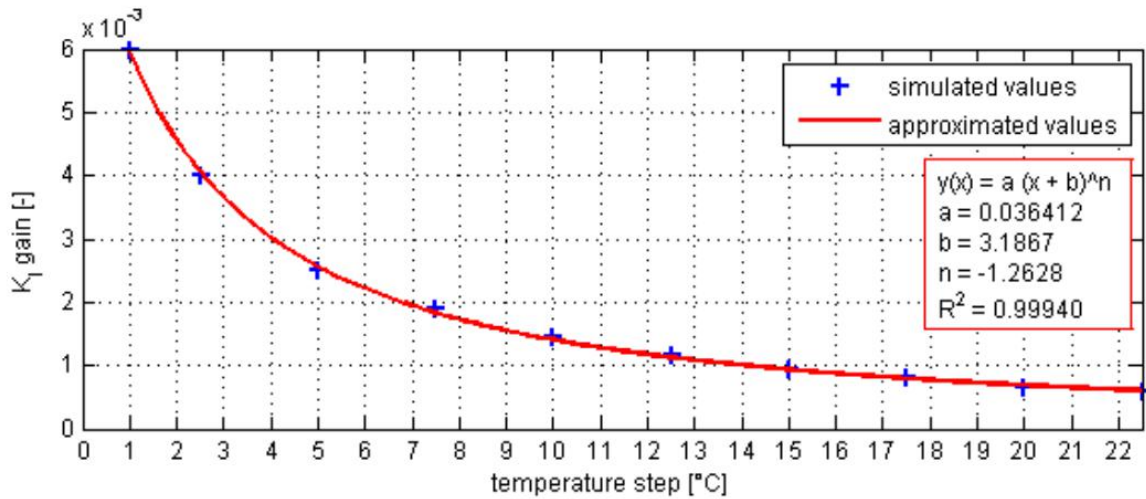


Рис. 3.21. Апроксимація функції посилення K_I

Основний алгоритм керування використовує 10-секундний цикл, у якому вимірюється температура, розраховується новий вихід для нагрівача, а виміряні температури записуються у файл журналу. Кожного разу, коли задане значення змінюється (зазвичай користувачем), нове K_I посилення обчислюється на основі апроксимованої функції (рис. 3.21). Тільки таким чином можна досягти постійного перевищення для будь-якого розміру кроку температури. Якщо використовується фіксована константа K_I , перевищення встановленого значення зростає зі збільшенням розміру кроку. Це викликано в основному насиченістю виходу контролера (нагрівач має обмежену максимальну потужність).

Змодельоване керування температурою та фактичні виміряні дані наведено на рисунках 3.22 і 3.23. На рис. 3.22 показано постійне перевищення для зміни заданого значення (30°C , 35°C , 40°C , 42°C) кожні 4 години.

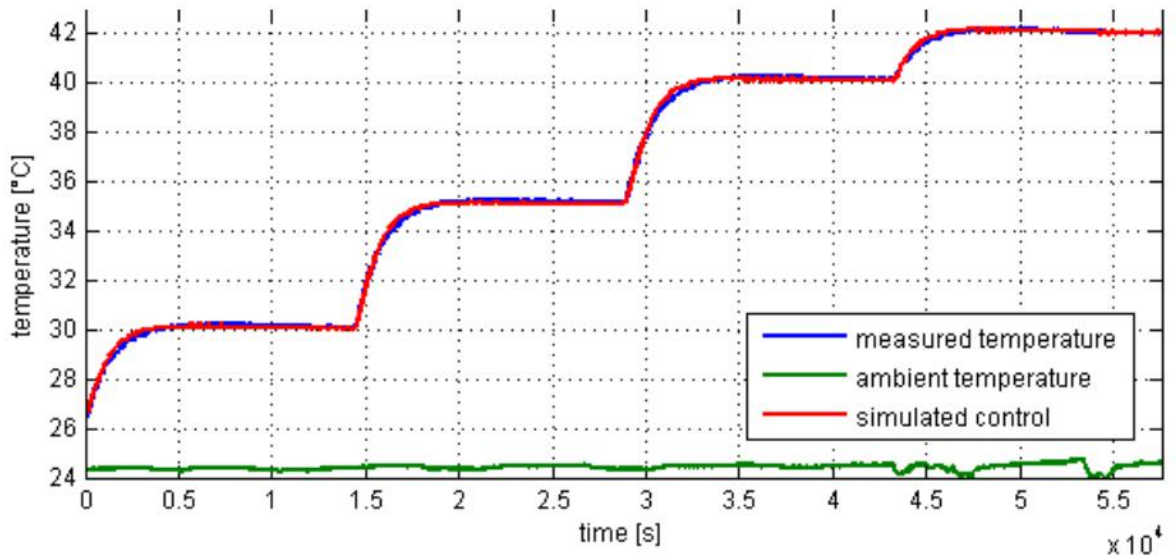


Рис. 3.22. Змодельовані кроки та виміряна температура

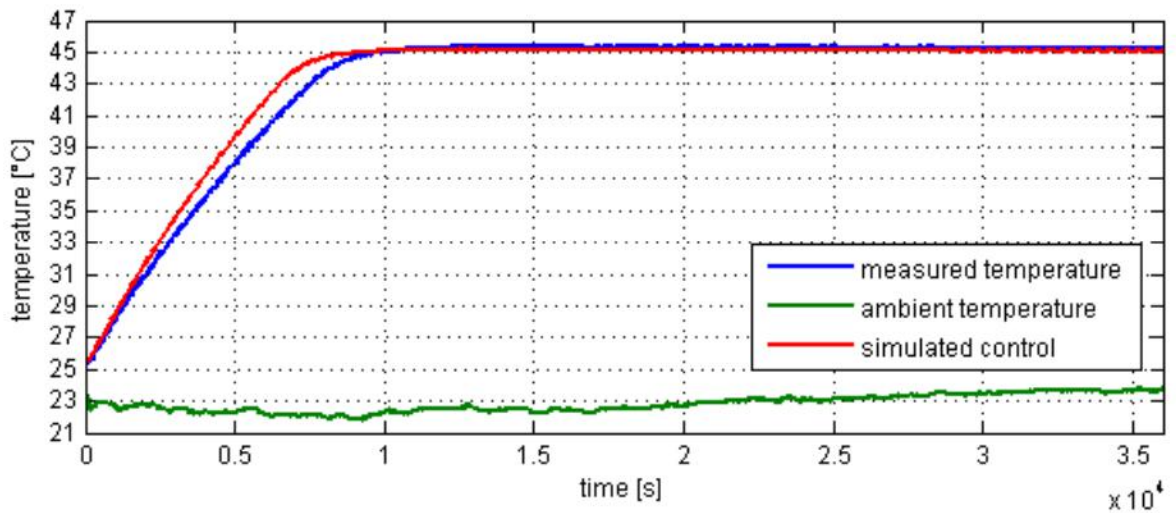


Рис. 3.23. Контроль температури для заданого значення 45°C (крок 20°C)

3.3. Програмне забезпечення

Основний цикл триває 10 с. Кожні 10 с вимірюється задане значення та температура та обчислюється вихідний сигнал контролера. Якщо є зміна заданого значення, алгоритм обчислює нову константу K_i та повторно запускає контролер. Якщо фактична температура на 0,1 °C нижча за встановлене значення, він розраховує нову константу K_i . Блок-схема алгоритму представлена на рис. 3.24.

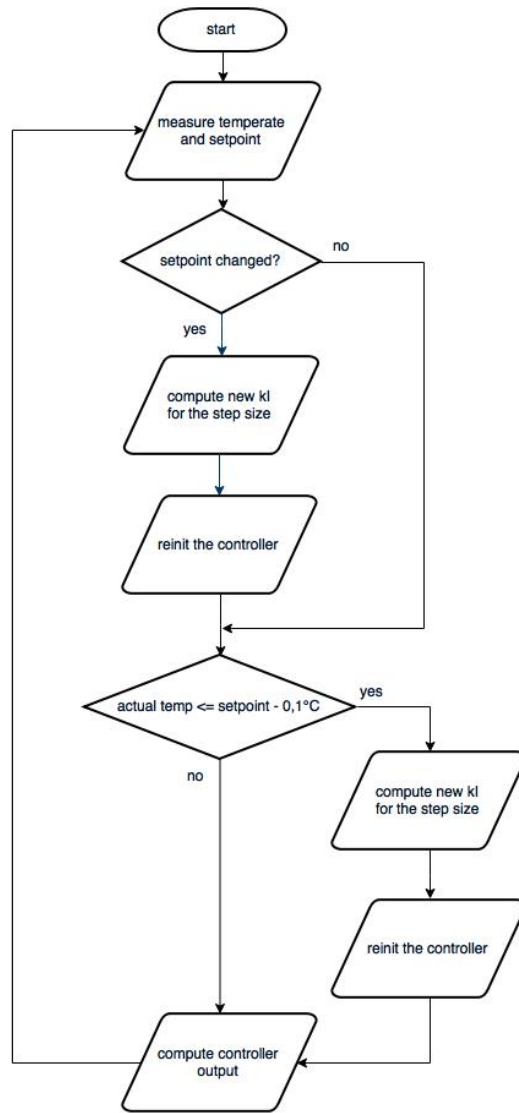


Рис.3.24. Блок-схема алгоритму

Далі представлено підключення для вихідного терміналу SSH

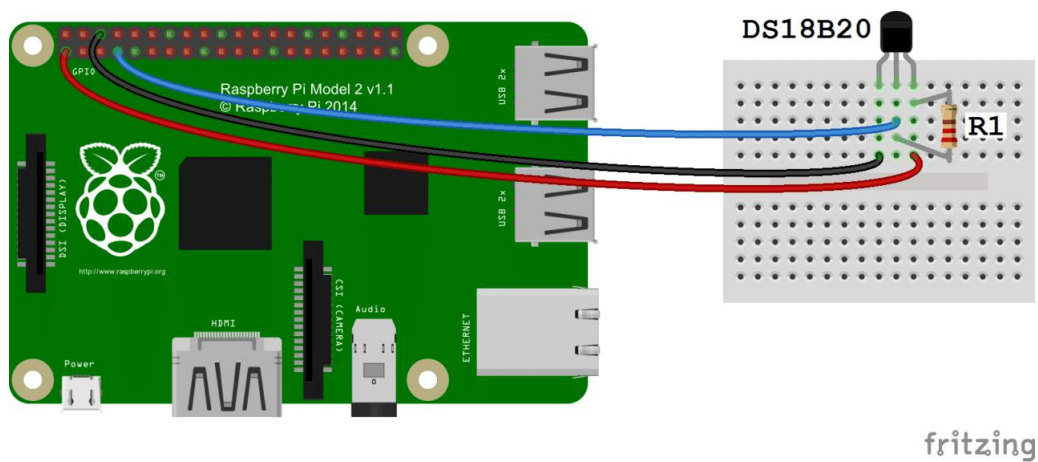


Рис. 3.25. Виведення температури на термінал SSH

Особливості ПЗ:

- використовує формат csv
- зчитування температури з sysfs linux за допомогою драйвера ядра
- резервне копіювання за допомогою файлу rw пристрою w1 у sysfs
- показання ковзної середньої температури
- жорстко закодована зона нечутливості 1С, протягом якої компонент Кр вимкнено
- відображення кількох датчиків на веб-інтерфейсі за допомогою вбудованого веб-сервера
- коментування точок даних
- перемикання ПД-регулювання від одного головного датчика
- ручний контроль стану перемикача
- живе оновлення графіка з використанням невеликого мережевого трафіку
- журналювання виконується до /tmp, а синхронізація з постійним сховищем виконується лише через певні проміжки часу або вимикається, щоб зменшити використання сховища



Рис. 3.26. Вигляд веб-інтерфейсу

У додатку А наведено програмний код розробленого ПЗ.

РОЗДІЛ 4 ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА

4.1. Директива щодо обмеження небезпечних речовин

Директива щодо обмеження використання небезпечних речовин 2002/95/EC (RoHS 1), скорочення від Директиви про обмеження використання певних небезпечних речовин в електричному та електронному обладнанні, була прийнята Європейським Союзом у лютому 2003 року.

Ініціатива мала на меті запобігти надлишку хімічних речовин в електроніці. Директива RoHS 1 набула чинності 1 липня 2006 року. Ця директива обмежує використання десяти небезпечних матеріалів у виробництві різних типів електронного та електричного обладнання. Крім винятків, існують винятки для таких продуктів, як сонячні панелі. Вона тісно пов'язана з Директивою щодо відходів електричного та електронного обладнання (WEEE) 2002/96/EC, яка встановлює цілі збору, переробки та відновлення електричних товарів і є частиною законодавчої ініціативи щодо вирішення проблеми величезна кількість токсичних електронних відходів.

Кожна держава-член Європейського Союзу прийме власну політику забезпечення виконання та впровадження, використовуючи директиву як керівництво.

RoHS часто називають «безсвинцевою директивою», але вона обмежує використання наступних десяти речовин:

- Свинець (Pb)
- Меркурій (Hg)
- Кадмій (Cd)
- Шестивалентний хром (Cr 6+)
- Полібромовані біфеніли (PBV)
- Полібромований дифеніловий ефір (PBDE)
- Біс (2-етилгексил) фталат (DEHP)
- Бутилбензилфталат (BBP)
- Дибутилфталат (DBP)
- Діізобутилфталат (DIBP)

Директива поширюється на наступне обладнання:

- Велика побутова техніка
- Дрібна побутова техніка
- ІТ і телекомунікаційне обладнання
- Споживче обладнання
- Освітлювальне обладнання – включаючи лампочки
- Електронні та електричні інструменти
- Іграшки, інвентар для відпочинку та спорту
- Медичні прилади
- Інструменти моніторингу та контролю
- Автоматичні дозатори
- Інше ЕЕО, не охоплене жодною з категорій вище.

Це не стосується стаціонарних промислових установок та інструментів. Відповідність є відповідальністю компанії, яка випускає продукт на ринок, як визначено в Директиві; компоненти та вузли не несуть відповідальності за відповідність продукту. Звичайно, враховуючи той факт, що регулювання застосовується на рівні однорідного матеріалу, дані про концентрацію речовин необхідно передати через ланцюг поставок до кінцевого виробника.

RoHS поширюється на ці продукти в ЄС незалежно від того, вироблені в ЄС чи імпортовані.

4.2. Приклади компонентів продукту, що містять заборонені речовини

Речовини, обмежені RoHS, використовувалися в широкому спектрі побутової електроніки. Приклади компонентів, які містять свинець, включають:

- фарби і пігменти
- ПВХ (вінілові) кабелі (наприклад, шнури живлення, кабелі USB)
- припої
- обробка друкованих плат, виводи, внутрішні та зовнішні з'єднання

- скло в телевізійних і фотографічних виробках (наприклад, телевізійні екрани кінескопних трубок і об'єктиви камер)
- металеві частини
- лампи та лампочки
- батареї
- інтегральні схеми або мікрочіпи

Кадмій міститься в багатьох компонентах вище; приклади включають пластикову пігментацію, нікель-кадмієві (NiCd) батареї та фотоелементи CdS (використовуються в нічних світильниках). Ртуть використовується в освітлювальних приладах і автомобільних вимикачах; приклади включають люмінесцентні лампи та ртутні перемикачі нахилу (нині вони рідко використовуються). Шестивалентний хром використовується для обробки металу для запобігання корозії. Полібромовані біфеніли та дифенілові ефіри/оксиди використовуються переважно як антипірени.

4.3. Свинець в електроніці

В електронних виробках є три основні компоненти, що містять свинець: припій, покриття контактних площадок друкованих плат та покриття виводів електронних компонентів. У типовому паяному з'єднанні припій є основним фактором, що визначає наявність свинцю. Відповідно покриття друкованої плати та виводи електронних компонентів мають на утримання свинцю значно менший вплив. В результаті, перші кроки до скорочення вмісту свинцю в електронних виробках були спрямовані на пошук сплаву, що дозволяє замінити традиційні припої, що містять свинець. В даний час припій використовується в більшості випадків встановлення електронних компонентів на друковану плату:

- технологія поверхневого монтажу (використовується припій у вигляді пасти, що наноситься на поверхню друкованої плати за допомогою спеціального шаблону чи трафарету);
- монтаж в отвори плати (використовується розплавлений припій, що знаходиться у спеціальній ванні);

- ручне паяння за допомогою паяльника (як правило, припій застосовується у вигляді дроту, тонкої трубки або стрічки).

В результаті численних експериментів, які проводилися протягом останніх декількох років, в якості заміни припоїв, що містять свинець, було запропоновано сімейство сплавів олова, срібла і міді (SnAgCu).

Для технології поверхневого монтажу сплав SnAgCu, ймовірно, стане найпопулярнішим рішенням. Можливо, найбільшою проблемою для використання такого сплаву як припой є більш висока температура плавлення. Наприклад, температура плавлення сплаву SnAgCu становить 217°C , тоді як сплав олова і свинцю Sn37Pb плавиться при температурі 183°C . Відповідно, технологічний процес вимагатиме збільшення температури паяння до $240\text{--}260^{\circ}\text{C}$.

Зрештою, відповідність вимогам директиви RoHS є зоною відповідальності не виробника. Вам необхідно контролювати вибір безсвинцевих компонентів, фінішного покриття та не забувати вказувати усі необхідні примітки у конструкторській документації для того, щоб виробнику було максимально зручно. Це колективне завдання, яке ви повинні вирішувати разом для того, щоб наш світ став менш токсичним.

РОЗДІЛ 5 ОХОРОНА ПРАЦІ

5.1. Аналіз шкідливих та небезпечних факторів

Робота ведеться в приміщенні 5 м на 4 м, висота приміщення 3 м: площа приміщення становить 35 метрів квадратних, а об'єм - 60 метрів кубічних. Кількість робочих місць - 6. Система опалення централізованого типу, складається з двох батарей. Наявне природне і штучне освітлення комбінованого типу: природне освітлення - вікно 2 на 2 метри; штучне освітлення представлене світильником загального освітлення і настільною лампою на робочому місці.

При роботі на інженера-програміста, що працює можуть діяти наступні небезпечні та шкідливі виробничі фактори відповідно [ГОСТ-12.0.003-74, 29]:

- Підвищений рівень шуму на робочому місці;
- Підвищений рівень електромагнітного випромінювання;
- Недостатнє штучне освітлення;
- Розумове перенапруження;
- Перенапруження аналізаторів(зорових);

Підвищений рівень шуму на робочому місці [30] (дБ):

Фактичне значення: 64 дБ ГДР: 54 дБ

№ п/п	Вид трудової діяльності, робоче місце	Рівні звукового тиску в дБ в октавних смугах з середньгеометричними частотами, Гц									Рівні шуму та еквівалентні рівні шуму, дБА, дБАекв.
		31,5	63	125	250	500	1000	2000	4000	8000	
1	2	3	4	5	6	7	8	9	10	11	12
1	Творча діяльність, керівна робота з підвищеними вимогами, наукова діяльність, програмування.	86	71	61	54	49	45	42	40	38	50

Підвищений рівень електромагнітного випромінювання [31] (кВ/м):

Фактичне значення: 25 ГДР: 20

Недостатнє штучне освітлення [31] (КПО, %):

Фактичне значення: 1 ГДР: 1,5

Розумове перенапруження [31] (годин безперервної роботи):

Фактичне значення: 2 ГДР: 1

Перенапруження зорових аналізаторів [31] (годин безперервної роботи):

Фактичне значення: 2 ГДР: 1

Підвищений рівень шуму на робочому місці:

Підвищений рівень шуму на робочому місці генерується системою охолодження ЕОМ та системою охолодження радіо випромінювача. Рівень шуму є постійним у часі.

При шумі, що перевищує ГДР, у людини, що працює поряд з джерелом шуму, може виникати дратівливість, відчуття неспокою, порушення слухової активності, зниження уваги і головний біль [30].

Підвищений рівень електромагнітного випромінювання:

Підвищений рівень електромагнітного випромінювання на робочому місці виникає в наслідок роботи комп'ютера, від розеток і випромінювача

радіохвиль.

Тривале перебування у зоні підвищеного рівня електромагнітного випромінювання викликає у людини порушення ЦНС, дратівливість и зниження рівня уваги, знижує можливість концентруватись [30].

Недостатнє природне освітлення

Недостатнє природне освітлення виникає через неправильне розміщення робочого місця відносно вікна, недостатній розмір вікна і наявність об'єктів, що поглинають природне світлове випромінювання. Нестача природного освітлення може викликати проблеми із зором, порушення сприйняття навколишніх об'єктів і неможливість виконувати необхідні робочі операції відповідно до вимог класу роботи [31].

Розумове перенапруження:

Розумове перенапруження виникає в наслідок необхідності оператора швидко та своєчасно приймати відповідальні рішення. Тривале розумове напруження зумовлює появу стресового стану у людини і порушень сприйняття та обробки інформації [31].

Перенапруження аналізаторів(зорових):

Перенапруження зорових аналізаторів виникає в наслідок постійної необхідності сприймати інформацію з екрана ЕОМ і відслідковувати навіть незначні зміни.

Тривале перенапруження зумовлює погіршення зору, біль в очах, неможливість чітко сприймати зображення, неможливість зосередитись [31].

Найбільш небезпечним є фактор нестача природного освітлення, оскільки цей фактор впливає на якість роботи оператора і може спричинити серйозні, але малопомітні на перший погляд наслідки.

5.2. Розробка заходів з охорони праці

Зниження рівня шуму в приміщенні як наслідок використання облицювання з більш високим коефіцієнтом звукопоглинання, дБ, можна визначити за формулою

$$\Delta L_{\text{обл}} = 10 \lg \frac{B_2}{B_1},$$

де B_1 та B_2 – сталі приміщення відповідно до та після облицювання.

У загальному випадку

$$B = \frac{A}{1 - \alpha_{\text{ср}}},$$

де A – еквівалентна площа звукопоглинання, $A = \alpha_{\text{ср}} \cdot S_{\text{нов}}$;

$\alpha_{\text{ср}}$ – середній коефіцієнт звукопоглинання внутрішніх поверхонь приміщення площею $S_{\text{пов}}$;

$$S_{\text{пов}} = S_{\text{підл}} + S_{\text{ст}} + S_{\text{стелі}},$$

$S_{\text{підл}}$ – площа підлоги, м²;

$S_{\text{ст}}$ – площа стін, м²;

$S_{\text{стелі}}$ – площа стелі, м².

За таких означень стала приміщення до облицювання складає

$$B_1 = \frac{\alpha_{\text{ср1}} \cdot S_{\text{ном}}}{1 - \alpha_{\text{ср1}}}.$$

При визначенні сталої приміщення після облицювання треба звернути увагу на зміну середнього коефіцієнта звукопоглинання. До облицювання він складав для всіх внутрішніх поверхонь за умовами задачі $\alpha_{\text{ср1}} = 0,1$. Після облицювання підлога залишилась з попереднім коефіцієнтом звукопоглинання ($\alpha_{\text{ср}} = 0,1$), а у стелі та стін він буде дорівнювати $\alpha_2 = 0,9$. Середній коефіцієнт звукопоглинання після облицювання можна визначити як середньозважену величину від площі внутрішніх поверхонь, що мають різні коефіцієнти звукопоглинання:

$$\alpha_{\text{ср2}} = \frac{(S_{\text{ст}} + S_{\text{стелі}}) \cdot \alpha_2 + S_{\text{підл}} \cdot \alpha_{\text{ср1}}}{S_{\text{підл}} + S_{\text{ст}} + S_{\text{стелі}}}.$$

$$\alpha_{сер} = \frac{(280 + 250) \cdot 0.9 + 250 \cdot 0.1}{280 + 250 + 250} = \frac{502}{780} = 0.64$$

$$B_1 = \frac{0,64 * (280 + 250 + 250)}{1 - 0.64} = 1387$$

$$A = 0.1 \cdot (280 + 250 + 250) = 78$$

$$B = \frac{78}{1 - 0.1} = 86,66$$

$$\Delta L_{обл} = 10lg \frac{1387}{86.66} = 12.04$$

Рівень шуму в приміщенні після облицювання, дБА, визначається за формулою

$$L_2 = L_1 - \Delta L_{обл}.$$

$$L_2 = 71 - 12.04 = 58,96$$

Порівнюючи рівень шуму в приміщенні після облицювання з допустимим за ГОСТом 12.1.003-83.

5.3. Пожежна безпека приміщення

Приміщення з повинні бути оснащені системою автоматичної пожежної сигналізації відповідно до вимог Переліку однотипних за призначенням об'єктів, які підлягають обладнанню автоматичними установками пожежогасіння та пожежної сигналізації з димовими пожежними сповіщувачами та переносними вуглекислотними вогнегасниками з розрахунку 2 шт. на кожні 20 кв. м площі приміщення з урахуванням граничнодопустимих концентрацій вогнегасної рідини відповідно до вимог Правил пожежної безпеки в Україні [33].

Приміщення, в якому експлуатується пристрій, відноситься до категорій приміщень типу Д "Знижена пожежна небезпечність" (відсутні матеріали, що легко займаються) і має II клас вогнестійкості(стіни та перекриття - бетонні).

Потенційну пожежну небезпеку становлять електромережа та ЕОМ: коротке замикання може спричинити суттєве нагрівання структурних елементів ЕОМ і спричинити їх займання.

Для запобігання займання, поряд з розетками електромережі, блоків живлення та силових контактів не повинні лежати жодні предмети, що можуть проводити струм та спричинити коротке замикання, або які є пожежонебезпечними.

Є неприпустимим:

- Експлуатація кабелів та проводів з пошкодженою або такою, що втратила захисні властивості за час експлуатації, ізоляцією; залишення під напругою кабелів та проводів з неізольованими провідниками;

- Застосування саморобних продовжувачів, які не відповідають вимогам до переносних електропроводок;

- Застосування для опалення приміщення нестандартного (саморобного) електронагрівального обладнання або ламп розжарювання;

- Користування пошкодженими розетками, розгалужувальними та з'єднувальними коробками, вимикачами та іншими електровиробами, а також лампами, скло яких має сліди затемнення або випинання;

- Підвішування світильників безпосередньо на струмопровідних проводах, обгортання електроламп і світильників папером, тканиною та іншими горючими матеріалами, експлуатація їх зі знятими ковпаками (розсіювачами);

- Використання електроапаратури та приладів в умовах, що не відповідають вказівкам (рекомендаціям) підприємств-виготовлювачів.

Осіб, які не пройшли інструктаж з пожежної безпеки, не можна допускати до роботи. Кожен працівник зобов'язаний виконувати вимоги щодо пожежної безпеки, а також вживати заходів щодо усунення порушень правил пожежної безпеки, ліквідації пожеж і загорянь. Кожен працівник повинен знати місце розташування первинних засобів пожежогасіння і вміти ними користуватися, працівники повинні знати правила поведінки при пожежі, шляхи евакуації. У разі виникнення пожежі працівники повинні негайно повідомити про це пожежну охорону (по телефону) та керівництво установи і розпочати ліквідацію пожежі всіма наявними засобами.

У разі виявлення запаху горілого, диму або іскор негайно вимкнути напругу на щитку і візуально перевірити чи не виникло займання. У разі займання - сповістити оточуючих про наявність пожежі та докласти можливих зусиль для її локалізації.

5.4. Інструкція з охорони праці інженера-програміста

1. ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 До роботи допускаються Інженери, що пройшли:

- Професійну підготовку
- Медичний огляд
- Вступний інструктаж
- Спеціальне навчання і перевірку знань з охорони праці
- Первинний інструктаж на робочому місці

1.2 Перевірка знань з охорони праці проводиться раз на 12 місяців в обсязі інструкцій з охорони праці.

1.3 Інструкція має переглядатись не рідше одного разу на 3 роки.

1.4 Професія Інженера пов'язана з такими видами небезпеки:

- Перенапруження зорових аналізаторів
- Підвищений рівень електромагнітного поля
- Тривалий вплив штучного світлового випромінювання

2. ВИМОГИ БЕЗПЕКИ ПЕРЕД ПОЧАТКОМ РОБОТИ

2.1 За 10 хвилин до початку роботи Інженер має прибути на робоче місце і підготувати його до роботи

2.2 Інженер має оглянути робочий стіл; протерти його; перевірити правильність підключення електроприладів.

2.3 Інженер має ввімкнути основні електроприлади і освітлення.

2.4 Інженер має розкласти необхідні для роботи інструменти на столі. Інші предмети, не призначені для роботи не мають знаходитися на робочому місці під час робочого дня.

2.5 У разі виявлення проблем, Інженер має повідомити безпосереднього начальника.

3. ВИМОГИ БЕЗПЕКИ ПІД ЧАС РОБОТИ

3.1 Під час роботи з приладом, Інженер має керуватися правилами електробезпеки: не торкатися оголених провідників струму; при роботі з чутливою електронікою користуватись заземленням; не відключати обладнання від мережі, якщо воно ще працює і не вимкнене; не залишати ввімкненими електроприлади, якщо необхідно покинути робоче місце з важливої причини.

3.2 Якщо під час роботи виявлено наявність забруднення в ході робочого процесу, яке заважає подальшій роботі, то таке забруднення має бути усунуте тим Інженером, що його спричинив, в найближчий термін.

3.3 Якщо природного освітлення достатньо для роботи, Інженер може вимкнути штучне освітлення свого робочого місця, для зниження навантаження на очі.

3.4 Інженер не має відволікати інших Інженерів під час роботи.

3.5 У разі виявлення порушень, Інженер має повідомити безпосереднього начальника.

4. ВИМОГИ БЕЗПЕКИ ПІСЛЯ ЗАКІНЧЕННЯ РОБОТИ

4.1 Після завершення робочого дня, Інженер має зберегти результати роботи електроприладів, якщо це необхідно.

4.2 Інженер має вимкнути всі електричні прилади на робочому місці і перевірити їх стан.

4.3 Інженер має скласти інструменти до відповідних відділів робочого місця і перевірити, чи наявні всі інструменти, що були використані під час робочого дня і в якому вони стані.

4.4 При виявленні будь-яких проблем, Інженер зобов'язаний повідомити безпосереднього начальника.

5. ВИМОГИ БЕЗПЕКИ В АВАРІЙНИХ СИТУАЦІЯХ

- 5.1 При виникненні аварійної ситуації, Інженер має негайно припинити роботу і повідомити безпосереднього начальника про випадок.
- 5.2 При загорянні електропроводки або самих електроприладів, Інженер має знеструмити робоче місце, повідомити про випадок безпосереднього начальника та вжити заходів щодо локалізації наслідків аварійної ситуації. Далі проводиться розслідування причин виникнення аварійної ситуації, встановлюються винні, проводяться заходи щодо попередження виникнення подібних ситуацій надалі.
- 5.3 При травмуванні Інженера в наслідок ураження струмом, необхідно негайно відключити джерело струму ураження, сповістити безпосереднього начальника, викликати швидку та надати постраждалому першу медичну допомогу. Далі проводиться розслідування причин виникнення аварійної ситуації, встановлюються винні, проводяться заходи щодо попередження виникнення подібних ситуацій надалі.
- 5.4 При руйнуванні робочого місця в наслідок техногенних причин, Інженер має негайно покинути робоче місце, сповістити про випадок свого безпосереднього начальника. Начальник приймає рішення про подальші дії стосовно пошкодженого робочого місця. Далі проводиться розслідування причин виникнення аварійної ситуації, встановлюються винні, проводяться заходи щодо попередження виникнення подібних ситуацій надалі.

ВИСНОВКИ

В дипломній роботі було проаналізовано літературні джерела, типи датчиків температури, теплопровідність та шляхи теплопровідності; визначено теплопровідність через друковану плату і на основі цього надані інструкції з проектування для вимірювання температури повітря;

змодельовано та експериментально досліджено цифровий датчик вимірювання температури DS18B20.

Попит на вимірювання температури в сучасних передових виробничих середовищах зріс і охоплює майже всі додатки в автомобільній, промисловій, персональній електроніці, комунікаціях і підприємствах. Враховуючи різноманітність застосувань, які вимагають моніторингу температури, існує багато міркувань при виборі найефективнішого типу датчика, включаючи точність датчика, розмір, розміщення, дрейф і калібрування.

Розробляючи друковану плату з датчиком температури, розробник системи повинен враховувати, чи є метою вимірювання температури навколишнього повітря чи моніторинг температури сусідньої енергоємної мікросхеми. Для вимірювання температури навколишнього повітря фізична ізоляція між датчиком і теплогенеруючими компонентами на одній друкованій платі є критичною. Крім того, врахування теплопровідних шляхів, таких як площини GND, відіграє важливу роль, щоб переконатися, що сусідні компоненти не спричиняють помилкових показань температури навколишнього середовища.

За допомогою ідентифікації та моделювання системи розроблено та реалізовано алгоритм керування. Цей пристрій базувався на комп'ютері Raspberry Pi під керуванням Raspbian Linux із платою розширення UniPi. Програма управління була написана на мові Python 3.2 з використанням об'єктно-орієнтованого програмування.

Перевищення встановленого значення було змодельовано на $0,2^{\circ}\text{C}$. Максимальне вимірне перевищення становило $0,45^{\circ}\text{C}$ для кроку 20°C . Час досягнення заданого значення був на 799 с довшим, ніж при моделюванні. Максимальне перевищення заданого значення для кроків 5°C становило $0,25^{\circ}\text{C}$ для кожного кроку, а час досягнення заданого значення був на 800 с довшим, ніж у симуляції для першого кроку заданого значення (30°C) і на 60 с довшим для наступних заданих значень (35°C , 40°C , 42°C).

Після досягнення максимального перевищення температура знижується до заданого значення, де вона залишається стабільною на $\pm 0,1^{\circ}\text{C}$. Всі вимоги до пристрою були дотримані.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Domínguez-Castro, Fernando, José M. Vaquero, Manuela Marín, María C. Gallego, und Ricardo García-Herrera. "How useful could Arabic documentary sources be for reconstructing past climate?" *Weather* 67, Nr. 3 (2012): 76–82. <https://doi.org/10.1002/wea.835>.
2. Moberg, Roland. "Linné on line – Linnaeus' thermometer", 2008. http://www.linnaeus.uu.se/online/life/6_32.html.
3. Thomson, William. "On an Absolute Thermometric Scale". *Philosophical Magazine*, Oktober 1848. http://zapatopi.net/kelvin/papers/on_an_absolute_thermometric_scale.html.
4. T. D. McGee, *Principles and Methods of Temperature Measurement*, New York, John Wiley and sons, 1988, p.3
5. R. J. Soulen Jr, A brief history of the development of temperature scales: the contributions of Fahrenheit and Kelvin. *Superconductor Science and Technology*, vol. 4, no. 11, 1991, pp. 696-699. p. 696
6. M. P. Silverman, *A Universe of Atoms, An Atom in the Universe*, New York, NY, Springer-Verlag, 2002, p. 49
7. S. Blundell, *Superconductivity: A Very Short Introduction*, Oxford, Oxford University Press, 2009, p.5
8. H. E. Landsberg, A note on the history of thermometer scales. *Weather*, vol. 19, no. 1, 1964, 2-6. p. 2
9. R. Boyle, *The Works of the Honourable Robert Boyle In Six Volumes*, Volume 2, London, W. Johnson and colleagues, 1772, p.585
10. U. Besson, Teaching about thermal phenomena and thermodynamics: The Contribution of the History and Philosophy of Science, In M. R. Matthews (Ed.), *International Handbook of Research in History, Philosophy and Science Teaching*, Dordrecht, Springer, 2014, pp.245-283, p. 276
11. H. Chang, *Inventing Temperature: Measurement and Scientific Progress*, Oxford, Oxford University Press, 2004, p. 170
12. D. Lindley, *Degrees Kelvin: A Tale of Genius, Invention, and Tragedy*,

Washington, DC, Joseph Henry Press, 2004, pp. 100-101

13. R. Boyle, *The Philosophical Works of the Honourable Robert Boyle Esq*, London, W. Inys, R. Manby and T. Longman, 1738, p. 640

14. Ramon Pallas-Areny and John G. Webster, *Sensors and Signal Conditioning*, John Wiley, New York, 1991.

15. Dan Sheingold, Editor, *Transducer Interfacing Handbook*, Analog Devices, Inc., 1980.

16. Walt Kester, Editor, *1992 Amplifier Applications Guide*, Section 2, 3, Analog Devices, Inc., 1992.

17. Walt Kester, Editor, *System Applications Guide*, Section 1, 6, Analog Devices, Inc., 1993.

18. Jim Williams, *Thermocouple Measurement*, Linear Technology Application Note 28, Linear Technology Corporation.

19. Dan Sheingold, *Nonlinear Circuits Handbook*, Analog Devices, Inc.

20. James Wong, *Temperature Measurements Gain from Advances in Highprecision Op Amps*, *Electronic Design*, 15 May 1986.

21. *OMEGA Temperature Measurement Handbook*, Omega Instruments, Inc.

22. *Handbook of Chemistry and Physics*, Chemical Rubber Co.

23. Paul Brokaw, *A Simple Three-Terminal IC Bandgap Voltage Reference*, *IEEE Journal of Solid State Circuits*, Vol. SC-9, December, 1974.

24. Ahmed S., Huang B., Shah S. L. 2007. Novel identification method from step response. *Control Engineering Practice*. 15(5): 545–556. URL: <http://www.sciencedirect.com/science/article/pii/S0967066106001778>

25. Franklin G., Powell F. J., Emami-Naeini A. 2006. *Feedback control of dynamic systems*. 5th ed. Upper Saddle River, N.J.: Pearson Prentice Hall.

26. Gerardi M. H. 2003. *The microbiology of anaerobic digesters*. Hoboken, N.J.: Wiley-Interscience.

27. Noskievic P. 1999. *Modelling and identification of systems*. Ostrava: Montanex. (Czech).

28. Sulc B., Viteckova M. 2004. Theory and practice of regulation circuit design. 1st ed. Praha: CVUT press. (Czech)
29. ГОСТ 12.0.003-74
30. ГОСТ 12.1.006-84
31. ДСН 3.3.6.037-99
32. ДСанПіН 3.3.2-007-98
33. БНіП 23-05-95
34. НПАОП 0.00-1.28-10

```

#Основний файл програми
#!/usr/bin/env python3

import gzip
import sys
import copy
import io
import os
import time
import shutil;
import json;
import http.server
from os import curdir, sep, listdir, path
import email.utils as eut
import datetime;
import subprocess;

import cfg

lastActive = ''
lastSync = datetime.datetime.now()
latest = []
comment = None
csv = None

def task_update_temp(s):
    if not s.enabled:
        s.avg = None
        s.curr = None
        return

    if s.ma is None: s.ma = []

    retried = False
    for i in range(5):
        if not s.read():
            if retried: break
            retried = True
            continue

    if s.curr is None: s.avg = None
    else:
        s.ma.append(s.curr)
        if len(s.ma) > cfg.config['movingavg']: s.ma.pop(0)
        s.avg = round(sum(s.ma) / len(s.ma), 3)

    if cfg.debug: print("data: %s curr:%s avg:%s" % (s.id, s.curr, s.avg))

def sync(toTemp = False):
    print('syncing.. ' + str(toTemp))

```

```

f = cfg.datadir + '/' + cfg.config['active'] + '.csv'
t = '/tmp/' + cfg.config['active']
if toTemp:
    if path.isfile(f): shutil.copyfile(f, t)
else:
    if path.isfile(t):
        shutil.copyfile(t, f + '_tmp')
        shutil.move(f + '_tmp', f)
    else:
        print("temp not present")

def task_temp(now):
    global lastSync, lastActive
    global csv, latest, comment

    config = cfg.config

    # write file
    if not cfg.isRunning():
        latest = []
        lastActive = ''
        if csv != None:
            csv.close()
            csv = None
            cfg.unuseSensors()
            sync()
    else:
        if lastActive != config['active']:
            latest = []
            if csv != None:
                csv.close()
                sync()
            try:
                if config['sync']:
                    sync(True)
                    csv = open('/tmp/' + config['active'], 'a+')
            else: csv = open(cfg.datadir + '/' + config['active'] +
'.csv', 'a+')

            size = csv.tell()

            print('appending: ' + config['active'])
            # get used sensors
            csv.seek(0, io.SEEK_SET)
            l = csv.readline().strip()
            if l[:6] != '#date,': raise Exception
            l = l[6:]

            cfg.unuseSensors()
            idx = 1
            cfg.acquire()
            for item in l.split(','):

```

```

        for s in cfg.sensors:
            if s.name == item:
                s.used = idx
                idx += 1
                break
    cfg.release()

    # date recovery
    csv.seek(size - min(128, size), io.SEEK_SET)
    tail = csv.read(128)
    if len(tail) > 5 and tail.rfind('\n', 0, -2) != -1:
        tail = tail[tail.rfind('\n', 0, -2) + 1:].strip()
        if cfg.debug: print('tail: ' + tail)
        tail = tail[:tail.find(',')]
        last = datetime.datetime.strptime(tail,
'%Y/%m/%d %H:%M:%S')
        if cfg.debug: print('last: ' + str(last))
        if last > now:
            print('time adjust')
            subprocess.call(['date', '-s', tail])
    csv.seek(0, io.SEEK_END)
    lastActive = config['active']
except:
    print('csv open failed: ' + str(sys.exc_info()[0]))
    cfg.config['running'] = False
    if csv != None:
        csv.close()
        csv = None

# csv writer
if csv != None:
    newdata = []
    newdata.append(int(now.timestamp() * 1000))
    csv.write(now.strftime('%Y/%m/%d %H:%M:%S'))

    ssens = cfg.getSensorView()
    ssens.sort(key=lambda s: s.used)
    for s in ssens:
        if not s.used: continue
        csv.write(',')

        if not s.enabled:
            v = None
        elif s.isSwitch():
            if s.curr:
                v = 1
                csv.write('true')
            else:
                v = 0
                csv.write('false')
    # pid out

```

```

        csv.write(',')
        pv = s.get()
        if pv is not None:
            csv.write(str(round(pv, 3)))
            newdata.append(pv)
    elif s.isTemp():
        v = s.avg
        if s.avg is not None: csv.write(str(v))
    # TODO - none
    #if v is None:
        newdata.append(v)

    if comment != None:
        s = cfg.getSensorByName(comment['sensor'])
        if s is not None:
            csv.write(' #' + str(s.used - 1) + ' ' +
comment['string'])
            comment = None

        csv.write('\n')
        csv.flush()
        latest.append(newdata)
        if len(latest) > 10: latest.pop(0)
        if config['sync'] and (now - lastSync).total_seconds() >
config['sync']:
            lastSync = now
            sync()

    if cfg.debug: print('waiting: ', config['update'])

class BrewHTTPHandler(http.server.BaseHTTPRequestHandler):
    def sendStatus(self):
        global latest
        self.send_response(200)
        self.send_header('Content-type', 'application/json')
        self.end_headers()

        status = copy.deepcopy(cfg.config)
        status['sensors'] = []
        for s in cfg.getSensorView():
            status['sensors'].append(s.dict())
        status['tail'] = latest
        status['date'] = datetime.datetime.now().strftime('%Y/%m/%d %H:%M:%S')
        self.wfile.write(bytearray(json.dumps(status), 'utf-8'))

    def log_message(self, format, *args):
        if not cfg.debug: return
        http.server.BaseHTTPRequestHandler.log_message(self, format, *args)

    def handle_one_request(self):

```

```

try:
    http.server.BaseHTTPRequestHandler.handle_one_request(self)
except:
    if cfg.debug: raise
    else: print('Unknown error: %s' % sys.exc_info()[0])

def do_GET(self):
    if self.path=="/": self.path="/index.html"

    if self.path=="/data/": self.handleData()
    else: self.handleFile()

def do_POST(self):
    varLen = int(self.headers['Content-Length'])
    postVars = str(self.rfile.read(varLen), 'utf-8')

    if not len(postVars):
        self.sendStatus()
        return;

    if cfg.debug:
        print('post: ' + postVars)

    j = json.loads(postVars)
    if self.path == '/comment': self.handleComment(j)
    elif self.path == '/main': self.handleMain(j)
    elif self.path == '/command': self.handleCommand(j)
    elif self.path == '/remove': self.handleRemove(j)
    elif self.path == '/toggle': self.handleToggle(j)
    else: self.handleConfig(j)

def handleComment(self, post):
    global comment
    if comment != None:
        self.send_error(500, 'Comment pending!')
        return

    if not cfg.isRunning():
        self.send_error(500, 'Not running')
        return

    s = cfg.getSensorByName(post['sensor'])
    if s is None or not s.enabled:
        self.send_error(500, 'Sensor not used in current brew!')
        return

    comment = {}
    comment['sensor'] = post['sensor']
    comment['string'] = post['comment']

    self.send_response(200)

```

```

self.end_headers()

cfg.event.set()
return

def handleMain(self, post):
    cfg.updateMainSensor(post['sensor'])

    self.send_response(200)
    self.end_headers()

    cfg.event.set()
    return

def handleRemove(self, post):
    cfg.removeSensor(post['sensor'])
    self.send_response(200)
    self.end_headers()
    return

def handleToggle(self, post):
    print('toggle sensor: ' + post['sensor'])

    s = cfg.getSensor(post['sensor'])
    if s is not None:
        s.force(post['force'])
        s.write(post['value'], post['force'])
        s.read()

    self.send_response(200)
    self.end_headers()

def handleCommand(self, nc):
    if 'command' in nc:
        cmd = nc['command']
        print('command: ' + cmd)

        if cmd == 'shutdown' or cmd == 'reboot':
            open('.clean_shutdown', 'w').close()
            if cmd == 'shutdown': subprocess.call(['shutdown', '-h', 'now'])
            else: subprocess.call(['reboot'])

        elif cmd == 'kill':
            try:
                for b in cfg.config['brewfiles']:
                    if b == nc['name']: cfg.config['brewfiles'].remove(b)
                    os.remove('data/' + nc['name'] + '.csv')
            except:
                return
            self.send_response(200)
        else:

```



```

        self.send_response(500, 'no command')
        self.end_headers()

def handleConfig(self, nc):
    config = cfg.config

    if cfg.debug: print('current config: ' + json.dumps(config))

    if 'sensors' in nc:
        for k,v in nc['sensors'].items():
            s = cfg.getSensor(k)
            if s != None:
                cfg.acquire()
                if cfg.debug: print('updating ' + s.name + ' ' +
json.dumps(v))
                if s.name != v[0]: s.name = v[0]
                if s.enabled != v[1]: s.enabled = v[1]

                if s.isTemp():
                    if s.min != v[2]: s.min = int(v[2])
                    if s.max != v[3]: s.max = int(v[3])
                elif s.isSwitch():
                    # override state if s.state != v[2]: s.state = int(v[2])
                    s.setpoint = float(v[3])
                    s.mode = int(v[4])
                cfg.release()
            cfg.updateMainSensor(None)

    #if 'main' in nc: config['main'] = nc['main']
    if 'active' in nc and config['active'] != nc['active']:
        print('new data file: ' + nc['active'])

        if config['sync']:
            sync(True)
            csv = open('/tmp/' + nc['active'], 'a+')
        else: csv = open(cfg.datadir + '/' + nc['active'] + '.csv', 'a+')
        csv.write('#date')

        cfg.acquire()
        idx = 1
        for s in cfg.sensors:
            if s.enabled:
                s.used = idx
                idx += 1
                csv.write(',') + s.name)
                if s.isSwitch(): csv.write(',') + s.name + '_pid')
            else: s.used = 0
        cfg.release()

        csv.write('\n')
        csv.close()

```

```

        sync()
        config['active'] = nc['active']
        config['brewfiles'].append(config['active'])

    if 'running' in nc: config['running'] = nc['running']
    if 'update' in nc: config['update'] = int(nc['update'])
    if 'sync' in nc: config['sync'] = int(nc['sync'])
    if 'debug' in nc: config['debug'] = nc['debug']
    if 'movingavg' in nc: config['movingavg'] = nc['movingavg']
    if 'date' in nc:
        try:
            datetime.datetime.strptime(nc['date'], '%Y/%m/%d %H:%M:%S')
            subprocess.call(['date', '-s', nc['date']])
        except:
            self.send_error(500, 'Bad date format!')
            print('bad date passed: ' + nc['date'])
            return

    cfg.store_config()
    cfg.event.set()
    self.sendStatus()

def handleFile(self):
    mime = {
        '.csv': 'text/html',
        '.html': 'text/html',
        '.map': 'application/json',
        '.jpg': 'image/jpg',
        '.gif': 'image/gif',
        '.js': 'application/javascript',
        '.css': 'text/css',
    }
    ext = self.path[self.path.rfind('.'): ]
    if cfg.debug: print('ext: ' + ext)

    if not ext in mime:
        m = 'text/html'
        print('extension not found: ' + ext)
    else: m = mime[ext]
    if ext == '.csv': sync()

    fn = curdir + sep + self.path
    try:
        ft = os.path.getmtime(fn)
        if self.headers.get('If-Modified-Since'):
            ts = datetime.datetime(*eut.parsedate(self.headers.get('If-
Modified-Since'))[:6])
            if ts >= datetime.datetime.utcfromtimestamp(round(ft)):
                self.send_response(304)
                self.end_headers()
                return

```

```

        f = open(fn)
        self.send_response(200)
        self.send_header('Content-type', m)
        self.send_header('Content-Encoding', 'gzip')
        self.send_header('Last-Modified', self.date_time_string(ft))
        self.send_response(200)
        self.end_headers()
        self.wfile.write(gzip.compress(bytearray(f.read()), 'utf-8'))
        f.close()
        return
    except IOError:
        self.send_error(404, 'File Not Found: %s' % self.path)
    except:
        self.send_error(500, 'Unknown error: %s' % sys.exc_info()[0])

def handleData(self):
    self.send_response(200)
    self.send_header('Content-type', 'text/html')
    self.end_headers()

    d = ''
    for f in listdir(cfg.datadir):
        d += '<a href="' + f + '>' + f + '</a><br>'
    self.wfile.write(bytearray(d, 'utf-8'))

```

```

# pids - Generic proportional-integral-derivative controllers.

```

```

...

```

```

Generic proportional-integral-derivative controllers.

```

ABSTRACT

A proportional-integral-derivative controller (PID controller) is a generic control loop feedback mechanism widely used in industrial control systems. A PID controller calculates an "error" value as the difference between a measured process variable and a desired setpoint. The controller attempts to minimize the error by adjusting the process control inputs.

For instance, a sensor may return a voltage representing a thermocouple reading for an oven temperature, and an actuator may be sent a signal using completely different voltage scale to describe the amount of fuel to burn in the oven. A PID controller can link the two systems as a thermostat: add more fuel when the oven cools, reduce fuel when the oven is overly hot, closing in on a desired temperature.

SYNOPSIS

```

>>> import pids

>>> p = pids.Pid( Kproportional, Kintegral, Kderivative )
>>> p.range( get_my_actuator_minimum(), get_my_actuator_maximum() )

>>> while True:
...     output = p.step( my_clock.get_dt(), get_my_sensor_value() )
...     set_my_actuator_value( output )

```

The tuning of the PID parameters is complicated, and depends on the design of the devices involved. Experimentation is key.

The meaning of the input, output and limit parameters are explained through a couple of simple example applications.

EXAMPLE APPLICATIONS

A PID Controller is often selected to control a servomotor.

Many common servomotors swing an arm through an arc, from a minimum position (e.g., -80 degrees) to a maximum (+80 degrees). The angle of the arm may be measured by eye in degrees, but electrically, this is sensed by an electrical resistance value (e.g., 100 Ohms to 1000 Ohms).

The input signal is a pulse width modulated signal, with pulses between a minimum width (e.g., 1ms high out of every 20ms) and a maximum width (2ms high out of every 20ms).

For the PID, this describes the units required for ranges and the set point.

- * the input is the measured angle (say, in ohms)
- * the set point is the desired angle (also in ohms)
- * the output is the pwm rate applied to motor (in milliseconds)

Another purpose for a PID controller is as an oven thermostat.

A measuring thermocouple is used to detect the current temperature in the oven. It has a reading in electrical resistance (e.g., 10 K Ohms at 300 degrees Celsius, and +100 Ohms per 10 degrees Celsius below that level). Sensors may have a non-linear relationship but the curve is generally sufficiently smooth to work here.

A heating coil is used to produce heat in the oven. It is rated to produce heat with a known level of electrical current (e.g., anywhere from 0 to 3 Amperes). An actual coil may simply be "on"

or "off," but by alternating the state, heat can be regulated more smoothly. The output of the controller should decide the level of alternation requested (e.g., from 0 or always off, to 1 or always on).

- * The input is measured temperature (in ohms)
- * The set point is desired temperature (in ohms)
- * The output is the heating coil activation (from zero to one)

```
class Pid (object):
```

```
    '''A discrete PID (Proportional-Integral-Derivative) controller.'''

    def __init__(self, P=1.0, I=1.0, D=1.0, point=0.0, below=-1.0, above=1.0):
        '''Sets up basic operational parameters for the controller.
        Three constants for the "tuning" of the controller can be given.
        * P (proportional gain) scales acceleration to new setpoints
        * I (integral gain) scales correction of error buildup
        * D (derivative gain) scales bounded rate of output change
        The initial desired output value or "point" can be given.
        The overall output range ("below" and "above") can be given.
        ...

        self.tune(P, I, D)
        self.range(below, above)
        self.output = below
        self.set(point)
        self.input = self.measure()

    def reset(self):
        self._integral = 0.0
        self._previous = 0.0
        self.output = self.setpoint

    def dead(self):
        return abs(self._previous) < 0.5

    def step(self, dt=1.0, input=None):
        '''Update the controller with a new input, to get new output.
        The time step "dt" can be given, or is assumed as an arbitrary 1.0.
        If a new "input" value a callable object, it is called for a value.
        If a new "input" value is not given here, measure() is called.
        ...

        if input is None:
            self.input = self.measure()
        elif callable(input):
            self.input = input()
        else:
            self.input = input

        err = self.setpoint - self.input
        err *= -1
```

```

self.output = 0.0
if not self.dead(): self.output += self.Kp * err
self._integral += err * dt
I = self._integral
self.output += self.Ki * I

D = (err - self._previous) / dt
self._derivative = D
self._previous = err
self.output += self.Kd*D

def bound(self, output):
    '''Ensure the output falls within the current output range.
    May be overridden with a new method if overshoot is allowed.
    '''
    return max(min(output, self.maxout), self.minout)

def range(self, below, above):
    '''Set the overall output range.
    Outputs are bounded to remain within this range with the bound()
    overridable method.
    '''
    if below > above:
        (above, below) = (below, above)
    self.minout = below
    self.maxout = above

def tune(self, P, I, D):
    '''Sets the three constant tuning parameters, P, I, and D.'''
    self.Kp = P
    self.Ki = I
    self.Kd = D
    self.reset()

def set(self, point):
    '''Sets the desired output value to which the controller seeks.'''
    self.setpoint = point

def get(self):
    '''Returns the current output value at any time.'''
    return self.bound(self.setpoint + self.output)

def measure(self):
    '''May be overridden to calculate a new input value.'''
    return 0.0

#-----

if __name__ == '__main__':

```

```

def linear(*args):
    '''Find a point along a linear path of two controls.
    Controls may be scalars, or vectors if arithmetic operators defined.

    May take three arguments, linear(i, A, B).
    Interpolated values equal control A at i==0.0, and control B at i==1.0.

    May take five arguments, linear(x, y, i, A, B).
    Interpolated values equal control A at i==x, and control B at i==y.
    ...
    if len(args) == 3:
        (xmin, xmax) = (0., 1.)
        (x, hmin, hmax) = args
    elif len(args) == 5:
        (xmin, xmax, x, hmin, hmax) = args
    else:
        raise ValueError('linterp() takes 3 or 5 arguments')
    return ( ((x)-(xmin)) * ((hmax)-(hmin)) / ((xmax)-(xmin)) + (hmin) )

def worm(terms, width=120):
    line = ' '*width
    for term in terms:
        left, x, right, sym = term
        h = int( linear(left, right, x, 0, width-2) )
        line = line[:h] + sym + line[h+1:]
    print("|" + line + "|")

class ServoPid (Pid):
    where = 0
    speed = 1
    maxwhere = 90
    minwhere = -90

    def __init__(self, **config):
        super(ServoPid, self).__init__(**config)
        self.speed = 1
        self.where = 0
        self.maxwhere = 90
        self.minwhere = -90
    def measure(self):
        self.where += self.output / 3.0
        self.where = max(min(self.where, self.maxwhere), self.minwhere)
        return self.where

import math
import random

pid = ServoPid()
pid.range(-10.0, 10.0)
#pid.tune(.8,.1,.1)

```

```
pid.tune(5, 0.25, -1.5)
pid.set(10)
for i in range(70):
    pid.step()
    worm( [ (pid.minwhere, pid.setpoint, pid.maxwhere, '+'),
            (pid.minwhere, pid.where, pid.maxwhere, '*') ] )
    if random.random() < 0.18:
        pid.set(random.random() * 55 - 27)
```