

СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет інформаційних технологій та електроніки

Кафедра програмування та математики

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної випускної роботи

освітній ступінь бакалавр
(бакалавр, магістр)

спеціальність 123 «Комп'ютерна інженерія»
(шифр і назва спеціальності)

спеціалізація _____
(назва спеціалізації)

на тему _____

Виконав: студент групи _____
(підпис) (ініціали і прізвище)

Керівник _____
(підпис) (ініціали і прізвище)

Завідувач кафедри _____
(підпис) (ініціали і прізвище)

Рецензент _____

ЛИСТ ПОГОДЖЕННЯ І ОЦІНЮВАННЯ
дипломної роботи студента гр. СКС-163 Мохнач М.П.

Науковий керівник

Доцент, к.т.н.

Захожай О.І.

Оцінка наукового керівника: _____

Рецензент _____

ПІБ, місто роботи, посада

Оцінка рецензента: _____

Кінцева оцінка за результатами захисту:

Голова ЕК

_____ Меняйленко О.С.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ СХІДНОУКРАЇНСЬКИЙ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет: Інформаційних технологій та електроніки

Кафедра: Програмування та математики

Освітньо-кваліфікаційний рівень: бакалавр

Напрямок підготовки: 6.050102 «Комп'ютерна інженерія»

(шифр і назва)

Спеціальність _____

(шифр і назва)

ЗАТВЕРДЖУЮ:

В.о. завідувач кафедри _____

_____ В.О. Лифар

« _____ » _____ 20 ____ р

ЗАВДАННЯ
НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ

Мохнач Микита Павлович

(прізвище, ім'я, по батькові)

1. Тема роботи Система перевірки температурного режиму

керівник проекту (роботи) Захожай О.І. кандидат технічних наук, доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від " " 2020р. № _____

2. Строк подання студентом роботи 07.06.2020р

3. Вихідні дані до роботи Сервіс моніторингу температури на основі Raspberry Pi

4.Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Аналітичний огляд, розробка апаратного забезпечення комп'ютерної системи, розробка програмного забезпечення комп'ютерної системи, висновки.

5. Перелік графічного матеріалу

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 22.03.2020

Керівник

_____ (підпис)

Завдання прийняв до виконання

_____ (підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Складання плану роботи	22.03.2020 - 24.03.2020	
2	Аналіз літератури	24.03.2020 - 29.03.2020	
3	Вивчення і підбирання матеріалу	29.03.2020 - 2.04.2020	
4	Написання розділів	02.04.2020 - 15.04.2020	
5	Написання програмного коду	16.04.2020 - 12.05.2020	
6	Оформлення пояснювальної записки	14.05.2020 - 26.05.2020	
7	Оформлення графічного матеріалу	28.05.2020 - 02.06.2020	
8	Підготовка доповіді і слайдів для презентації	05.06.2020 - 07.06.2020	

Студент

_____ (підпис)

_____ (прізвище та ініціали)

Науковий керівник

_____ (підпис)

_____ (прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка включає 52 сторінки, 15 рисунків, 6 таблиць, 8 джерел за переліком посилань.

Ключові слова: Raspberry Pi , одноплатний комп'ютер, API.

Мета роботи : Визначити способи отримання даних про температуру і реалізації збору даних різних систем, а також розробити комп'ютерну систему моніторингу температури з використанням онлайн-сервісів

ЗМІСТ

1 АНАЛІТИЧНИЙ ОГЛЯД.....	9
1.1 Прилади для визначення температури повітря.....	12
1.1.1 Ртутні термометри.	12
1.1.2 Спиртові термометри.....	13
1.1.3 Електричний термометр.....	14
1.2 Види термометрів за принципом дії.....	15
1.2.1 Контактные.....	15
1.2.2 Термометри опору:.....	15
1.2.3 Манометричні:.....	16
1.2.4 Безконтактні пірометри:.....	16
1.3 Апаратне забезпечення системи зчитування даних з датчиків температури.....	17
1.3.1 Процесор.....	20
1.3.2 Оперативна пам'ять.....	20
1.3.3 Мережеві адаптери.....	21
1.3.4. Відео підсистема.....	21
1.4 Цифровий датчик температури DS18B20.....	22
1.5 Приєднання датчика температури.....	24
1.6 Інтерфейс 1-wire.....	24
1.6.1 Апаратна реалізація інтерфейсу 1-wire.....	25
1.6.2 Програмна реалізація інтерфейсу 1-wire.....	26
1.6.3 Принципи адресуванні пристроїв та управління ними.....	29
2 РОЗРОБКА АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ.....	33
2.1 Розробка структурної схеми комп'ютерної системи.....	33
2.2 Розробка принципової схеми апаратного забезпечення комп'ютерної системи.....	35

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	41
3.1 Програмне забезпечення контролю температури.....	41
3.2 Програмне забезпечення веб-сервера.....	45
ВИСНОВКИ	51
ПЕРЕЛІК ЛІТЕРАТУРИ.....	52

1 АНАЛІТИЧНИЙ ОГЛЯД

Температурою називається статистична величина, що характеризує тепловий стан тіла і пропорційна середньої кінематичній енергії молекул тіла

Температура - параметр, який можна виміряти лише побічно, по зміні інших фізичних параметрів. Термометрію розрізняють на первинну і вторинну. У первинній термометрії температура явно описується через інші фізичні параметри, наприклад для газових термометрів цей тиск і об'єм. Прикладами вторинних термометрів є термометри опору і термопари. В промисловості термометри опору і термопари є Основними засобами контролю температури, закриваючи діапазон вимірювання від мінус 200 до 25000С і більше.

Температурні шкали розрізняються числовими значеннями, прийнятими для однакових реперних точок.

В температурної шкалою Цельсія за початок відліку прийнята температура затвердіння води (танення льоду), а інтервал між температурами танення льоду і кипіння води при нормальному атмосферному тиск (101325 Паскалів або на 760 мм рт. ст.) розділений на 100 рівних частин ($n = 100$). Назва дана шкала отримала на честь який запропонував її (в 1742 році) шведського вченого а. Цельсія (A. Celsius, 1701 1744). Температура за шкалою Цельсія виражається в градусах Цельсія, $^{\circ}\text{C}$), при цьому температура танення льоду приймається рівною 0°C , кипіння води 100°C .

Практична температурна шкала Реомюра, запропонована в 1730 році Р.А. Реомюр, який описав винайдений ним спиртовий термометр. Одиниця шкали Реомюра - градус Реомюра ($^{\circ}\text{R}$). 1°R дорівнює $1/80$ частині температурного інтервалу між опорними точками: танення льоду (0 градусів R) і кипіння води (80°R), т. см. $1^{\circ}\text{R} = 1,25^{\circ}\text{C}$. В даний час шкала вийшла з ужитку, найдовше вона зберігалася у Франції, на батьківщині автора.

В температурній шкалою Фаренгейта опорними точками є температура танення суміші снігу та нашатирю (0 градусів F) і нормальна температура людського тіла (100 ° F), а величина градуси в шкалі термометра визначається як сота частина інтервалу між опорними точками. Таким чином, температурний інтервал між точками танення льоду і кипіння води при нормальному атмосферний тиск) розділений на 180 частин - градусів Фаренгейта, ° F), причому точці танення льоду приписана до 32 ° F, а точки кипіння води 212 ° F. Дана шкала запропонована в 1724 ньому. фізиком Д.Г. Фаренгейтом (1686 1736); традиційно застосовується в ряді країн (зокрема, в США).

Для визначення значення температури якого-небудь тіла необхідно вибрати Еталон температури, тобто тіло, яке за певних умов, рівноважних і досить легко відтворюваних, мало б певне значення температури. Це значення температури є реперною точкою відповідної шкали температур - впорядкованої послідовності значень температури, що дозволяє кількісно визначати температуру того чи іншого тіла. Температурна шкала дозволяє непрямым чином визначати температуру тіла шляхом прямого вимірювання будь-кого з його фізичного параметра, яке залежить від температури.

Існують два основних способи для вимірювання температур - контактні та безконтактні. Контактні способи засновані на безпосередньому контакті вимірювального перетворювача температури з досліджуваним об'єктом, в результаті чого домагаються стану теплового рівноваги перетворювача та об'єкта.

Цьому способу притаманні свої недоліки. Температурне поле об'єкта спотворюється під час введення до нього термоприймача. Температура перетворювача завжди відрізняється від справжньої температури об'єкта. Верхня межа вимірювання температури обмежений властивостями матеріалів, з яких виготовлені температурні датчики.

Крім того, ряд завдань вимірювання температури в недоступних що обертаються з великою швидкістю об'єктах не може бути вирішене контактним способом.

У металевому термометрі опору вимір температури засновано на явищі зростання опору металу зі зростанням температури. Для більшості металів поблизу кімнатної температури ця залежність близька до лінійної, а для чистих металів відносно зміну їх опору при підвищенні температури на 1 К (температурний коефіцієнт опору) має величину близьку до $4 \cdot 10^{-3}$ 1/К. термометричною ознакою є електричний опір термометричного тіла - металевого дроту. Найчастіше використовують платинову дріт, а також мідну дріт або їх різні сплави. Діапазон застосування таких термометрів від водневих температур (~ 20 К) до сотень градусів Цельсія. При низьких температурах у металевих термометрах залежність опору від температури стає суттєво нелінійною, і термометр вимагає ретельної калібрування.

В напівпровідниковому термометрі опору (термісторі) вимір температури засновано на явищі зменшення опору напівпровідників зі зростанням температури. Так як температурний коефіцієнт опору напівпровідників за абсолютною величиною може значно перевершувати відповідний коефіцієнт металів, то і чутливість таких термометрів може значно перевершувати чутливість металевих термометрів.

Спеціально виготовлені напівпровідникові термосупротиву можуть бути використані при низьких (гелієвих) температурах близько декілька кельвін. Однак слід враховувати те, що у звичайних напівпровідникових опорах виникають дефекти, зумовлені впливом низьких температур.

Інший принцип вимірювання температури реалізований в термопар. Термопара являє собою електричний контур, спаяний із двох різних металевих провідників, один спай яких знаходиться при вимірюваній температурі (вимірювальний спай), а інший (еталонний спай) - за відомого температури,

наприклад, при кімнатній температурі. Через різницю температур сплавів виникає електрорушійна сила (термо ЕРС), часовий вимір яких дозволяє визначати різницю температур сплавів, а отже, температуру вимірювального спаяний.

Для вимірювання температури найбільшого поширення набули такі методи, засновані:

- на тепловому розширенні рідких, газоподібних і твердих тіл (термомеханічний ефект);
- зміні тиску всередині замкнутого обсягу при зміні температури (манометричні);
- зміні електричного опору тіл при зміні температури (терморезистори);
- термоелектричному ефекті;
- використанні електромагнітного випромінювання нагрітих тіл.

1.1 Прилади для визначення температури повітря

1.1.1 Ртутні термометри.

Вимірюють температуру повітря в межах -35°C до $+357^{\circ}\text{C}$. Точка замерзання ртуті дорівнює $-38,89^{\circ}\text{C}$, температура кипіння $+357,25^{\circ}\text{C}$. Вигляд ртутного термометру наведений на рисунку 1.1.

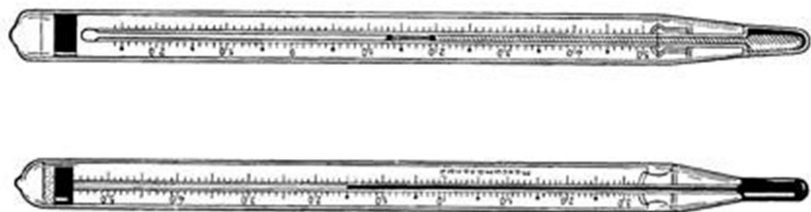


Рисунок 1.1 – Ртутний термометр

Термометр, зазвичай, має невеликий запас рідкої ртуті у своїй основі, який найчастіше перебуває під тиском; вище зазвичай скляна або синтетична трубка з каліброваною шкалою температур.

Ртутні термометри використовують метеорологами для вимірювання температури повітря. Стандартна ртуть замерзає при $-37,89\text{ F}$ ($-38,83\text{ C}$). Щоб виміряти температуру нижче цієї точки, метеорологи часто змішують Крейдяна таліюєвум сплавом. Талій знижує температуру замерзання до -78 F ($-61,1\text{ C}$). Якщо ртуть замерзає, за металом може потрапити азот, що призведе до припинення роботи термометра.

Рівень ртуті легко зчитується в скляних капілярних трубках, так як ртуть не змочує і не прилипає до трубки на відміну від інших рідин. Причина цього криється в тому, що молекули ртуті притягаються до один одного сильніше, ніж вони прилипають до скла або іншим матеріалам.

Єдиним недоліком ртуті є те, вона небезпечна для здоров'я і отруєє навколишнє середовище.

1.1.2 Спиртові термометри.

Спирт збільшується в об'ємі нерівномірно. За допомогою їх можна вимірювати більш низькі температури - до $-130\text{ }^{\circ}\text{C}$ (з урахуванням точки замерзання етилового спирту). У побутових умовах використання спиртових термометрів також цілком допустимо, тому що дозволяє фіксувати температуру повітря в реальних межах (точка закипання спирту - $+78,39\text{ }^{\circ}\text{C}$).

Спиртові термометри являють собою пристрої із скляного посудини, робочого наповнювача, який не замерзає при морозі, а також шкали. Сучасні аналоги не залежать також від тиску атмосфери і інших кліматичних факторів.

Недоліки спиртового термометра:

- має меншу точність в порівнянні зі ртутним;
- як правило, корпус дуже крихкий і є не придатним до ремонту;

-даний вид є непоширеним, а Отже знайти його буде досить складно.

1.1.3 Електричний термометр.

Датчики електричних термометрів (електротермометрів) використовуються термопари і термістори. Принцип дії термопари заснований на різній теплоємності металів, в результаті чого різні метали, з'єднані в пару при зміні температури мають різну ступінь нагріву. При цьому виникає термоелектричний струм, величина якого прямо пропорційна температурі, що фіксується потенціометром, градуйованому в °С. Вигляд електричний термометр наведений на рисунку 1.2.



Рисунок 1.2 – електричний термометр

Для визначення ходу температури повітря протягом певного проміжку часу (доба, тиждень і т.д.) використовують термограф(рисунок 1.3).

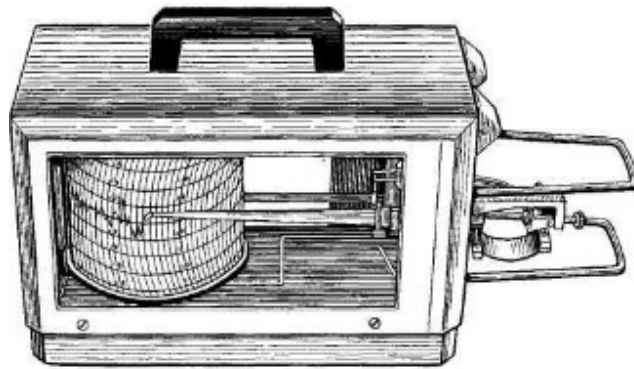


Рисунок 1.3 – Термограф

В цілому ряді сучасних приладів вимірювання температури повітря або поверхонь поєднане з вимірюванням інших показників метео- та мікрокліматичних умов

1.2 Види термометрів за принципом дії

1.2.1 Контактні

Такі прилади ще називають термометрами розширення. Вони засновані на відстежуванні зміни обсягу тіл під дією змінної температури. Зазвичай вимірюється діапазон температур становить від -190 до +500 градусів за Цельсієм.

1.2.2 Термометри опору

До цієї категорії відносяться прилади, які здатні вимірювати електричний опір речовин, мінливий залежно від температурних показників. Робочий діапазон цих пристроїв - від -200 до +650 градусів.

1.2.3 Електронні термомари

При нагріванні ці прилади генерують струм, що і дозволяє вимірювати температуру. Принцип дії заснований на вимірах термоелектродвіжущей сили. Діапазон вимірювань в цьому випадку - від 0 до +1800 градусів.

1.2.4 Манометричні

Такі термометри враховують залежність між температурними показниками і тиском газу. У вимірювану середу поміщають термобаллон, з'єднаний з манометром латунної трубкою. При нагріванні термобаллона тиск всередині нього збільшується, і ця величина вимірюється манометром. Таким чином проводять заміри температури в діапазоні від -160 до +600 градусів.

1.2.5 Безконтактні пірометри

В основі цих приладів – інфрачервоні датчики, зчитувальні рівень випромінювання. Вони поділяються на два види: яскравості, які проводять вимірювання випромінювань на певній довжині хвилі (діапазон - від +100 до +6000 градусів), і радіаційні, коли визначається теплову дію випромінювання (від -50 до +2000 градусів). Вони можуть використовуватися в тому числі і для визначення температури нагрітого металу, а також при налагодженні та випробуваннях котлів.

У термометрах використовують такі матеріали як:

- рідинні;
- газові;
- механічні;
- електричні;
- термоелектричні;

- волоконно-оптичні;
- інфрачервоні.

1.3 Апаратне забезпечення системи зчитування даних з датчиків температури

Для побудови комп'ютерної системи перевірки температурного режиму необхідно обрати апаратне забезпечення, яке відповідає наступним критеріям:

- мала потужність споживання (економічність);
- малі габарити та маса;
- наявність відкритих інтерфейсів для підключення датчиків температури;
- Можливість реалізації веб-сервера для відображення контенту в режимі веб-сторінки.

Найбільш прийнятним для такого переліку критеріїв є одноплатні комп'ютери типу OrangePi, Thinking Pad, RaspberryPi та ін. Всі ці системи мають приблизно однакові експлуатаційні показники, тому для реалізації проекту можна обирати будь-який з них. Оберемо, як найбільш розповсюджений – одноплатний комп'ютер RaspberryPi. Зовнішній вигляд цього комп'ютеру наведений на рисунку 1.4.

Міні-комп'ютер Raspberry Pi випускався або випускається в наступних версіях: «A», «A +», «B», «B +», «2B», «Zero», «Zero W», «3B», «3B +», «3A +» і «4B».

Однією з найцікавіших особливостей Raspberry Pi є наявність портів GPIO (general purpose input / output). Завдяки цьому його можливо використовувати для управління різними пристроями.

Дана плата, як і звичайний ПК, працює під управлінням однієї із спеціалізованих операційних систем.



Рисунок 1.4 – Зовнішній вигляд одноплатного комп'ютеру RaspberryPi

Операційна система Raspbian пропонує користувачам зручний модуль для програмного управління, який називається GPIO. GPIO є стандартним інтерфейсом підключення периферійного обладнання та датчиків.

Серія одноплатних комп'ютерів завбільшки з кредитну картку, розроблена у Великобританії Фондом Raspberry Pi з метою сприяння викладанню основних комп'ютерних наук у школах в країнах, що розвиваються. Оригінальні Raspberry Pi і Raspberry Pi 2 виробляються в кількох конфігураціях через ліцензовані угоди на виробничих потужностях RS Components і Egoman. Обладнання однаково у всіх виробників. Прошивка має закритий вихідний код.

Перші три версії оснащені ARM11 процесором Broadcom BCM2835 із тактовою частотою 700 МГц і модулем оперативної пам'яті на 256МБ/512МБ,

розміщеними за технологією "Package on package" безпосередньо на процесорі. Модель "2 B" оснащується процесором з 4 ядрами Cortex A7 з частотою 1ГГц і оперативною пам'яттю розміром 1 ГБ. Модель "A" оснащується одним USB 2,0 портом, модель "B" двома, а моделі "B" і "2 B" - чотирма. Також в моделях "B", "B" і "2 B" присутній порт Ethernet. Крім основного ядра, BCM2835 включає в себе графічне ядро з підтримкою OpenGL ES 2,0, апаратного прискорення і FullHD - відео і DSP ядро. Однією з особливостей є відсутність годин реального часу.

Відеосигнал можливо передати через композитний роз'єм RCA або через цифровий HDMI - інтерфейс. У версії "B" і "2B" виведення можливий через аудіораз'єм 3,5. Коренева файлова система, образ ядра та файли користувача розміщуються на карті пам'яті SD, MMC, microSD (тільки в моделі "B") або SDIO. Надалі розглядаються основні компоненти одноплатного комп'ютера Raspberry Pi.

Установка операційної системи на Raspberry Pi.

1. В першу чергу карту пам'яті потрібно відформатувати, вказуючи файлову систему FAT32.
2. Скачування архіву з NOOBS, його розпакування на карту пам'яті так, щоб файли були в кореневої директорії.
3. Вставка у Raspberry Pi необхідної периферії: клавіатури, миші і монітора через USB, підключення харчування. В якості монітора можна використовувати телевізор, під'єднаний через RCA.
4. Якщо екран підключений через RCA, натиснути "3" на клавіатурі.
5. У вікні вибрати операційну систему Raspbian. Натиснути "Install", дочекатися закінчення процесу установки та включення.
6. З контекстного меню "Configuration Tool" у третьому пункті вибрати другий варіант для установки графічного інтерфейсу LXDE.
7. Натиснути "Done", дочекатися перезавантаження.

8. Якщо потрібен вхід за допомогою логіна та пароля, у графі логін ввести `pi`, пароль `raspberrypi`, після цього відкриється робочий стіл.

1.3.1 Процесор

Система на кристалі (SoC), яку використовує в першому поколінні Raspberry Pi кілька еквівалентна чипам, використовуваним в старих smartfonkh (наприклад, iPhone, 3G, 3GS). Raspberry Pi заснований на Broadcom BCM2835 SoC, який включає в себе 700 МГц ARM Cortex A5 S процесор, VideoCore IV графічний процесор (GPU), і оперативну пам'ять. Він має кеш - пам'ять 1-го рівня 16 КБ і кеш - пам'ять 2-го рівня 128 Кб. Кеш - пам'ять 2-го рівня використовується переважно в GPU. SoC вкладається під мікросхеми ОЗП, так що видно тільки його кромка.

Raspberry Pi 2 використано Broadcom BCM2836 SoC з 900 МГц 32 бітовий чотирьохядерний процесор з ARM Cortex A7, з 256 Кб L2 загальною кеш - пам'яті і 1 Гб оперативної пам'яті. Графічний процесор ідентичний до оригіналу.

Raspberry Pi 3 використовує Broadcom BCM2837 SoC 64-розрядний чотирьохядерний ARM процесор Cortex-A53 з частотою 1,2 ГГц, з 512 Кб загального кешу L2.

1.3.2 Оперативна пам'ять

На старих бета платах моделі B, 128 МБ було виділено за замовчуванням на GPU, залишаючи 128 МБ для CPU. На першому 256 МБ релізі моделі B (і моделі A), були можливі 3 поділу. Ділення типовий було 192 Мб (RAM для CPU), якого має бути достатньо для автономного декодування відео 1080p, а також для простого 3D, але не разом. Для нової моделі B з 512 Мб оперативної пам'яті спочатку були опубліковані конфігурації поділу (`arm256_start.elf`, `arm384_start.elf`, `arm496_start.elf`) для 256 МБ, 384 МБ та 496 МБ оперативної

пам'яті CPU і 256 МБ, 128 МБ і 16 МБ відеопам'яті відповідно. Але через тиждень або трохи пізніше була випущена нова версія start.elf, яка могла прочитати новий контакт в config.txt (gpu_mem = xx) і динамічно призначити обсяг оперативної пам'яті (від 16 до 256 МБ з кроком 8 мегабайт) ГПУ, так що старий метод поділу пам'яті став застарілим, і один start.elf працював однаково для 256 і 512 Мб Raspberry Pis.

Raspberry Pi 2 і Raspberry Pi 3 мають 1 Гб оперативної пам'яті. Raspberry Pi Zero має 512 Мб оперативної пам'яті.

1.3.3 Мережеві адаптери

Хоча модель А, і Zero не мають 8P8С ("RJ45") порт Ethernet, вони можуть бути підключені до мережі за допомогою зовнішнього USB Ethernet або Wi-Fi адаптеру. На моделі В порт Ethernet забезпечується вбудованим USB адаптером Ethernet за допомогою мікросхеми SMSC LAN9514. Raspberry Pi 3 оснащений 2,4 ГГц Wi-Fi 802,11n (600 Мбіт / с) і Bluetooth 4,1 (24 Мбіт / сек) на Додаток до порту Ethernet 10/100.

1.3.4. Відео підсистема

Відео контролер працює з сучасними телевізійними дозволами, такими як HD і Full HD, а також більш високих або низьких дозволах моніторів або старих CRT телевізорів. При постачанні (тобто без розгону) він може працювати з наступними дозволами: 640 350 EGA; 640 480 VGA; 800 600 SVGA; 1024 768 XGA; 1280 720 720p HDTV; 1280 768 WXGA; 1280 800 WXGA; 1280 1024 SXGA; 1366 768 WXGA; 1400 1050 SXGA; 1600 1200 UXGA; 1680 1050 WXGA; 1920 1080 HDTV 1080p; 1920 1200 WUXGA.

Більш висока роздільна здатність, наприклад, до 2048 1152, або навіть 3840 2160 можуть працювати при 15 Гц. Можливість роботи з таким високим

дозволом не означає, що GPU зможе декодувати відео такого розміру. Відео дозволу FULL HD і менше відтворюються стабільно.

Хоча Raspberry Pi 3 не має апаратної підтримки декодування файлів H.265, процесор, більш потужний, ніж у його попередників, потенційно здатний декодувати H.265-відео програмно.

Raspberry Pi також можуть генерувати 576i і 480i композитні відеосигнали, використовувані в старих CRT екранах телевізорів (через нестандартні роз'єми, в залежності від моделі): PAL bghid, PAL M, PAL N, NTSC і NTSC J.

1.4 Цифровий датчик температури DS18B20

DS18B20 – це повноцінний цифровий термометр, здатний вимірювати температуру в діапазоні від -55оС до + 125оС з програмованої точністю 9-12 біт (рисунок 1.5). При виготовленні на виробництві, кожному датчику присвоюється свій унікальною 64-бітову адресу, а обмін інформацією з провідним пристроєм (мікро контролером або платою Arduino) здійснюється по шині 1-wire. Такий підхід дозволяє підключати до однієї лінії цілу групу датчиків, аж до 264.

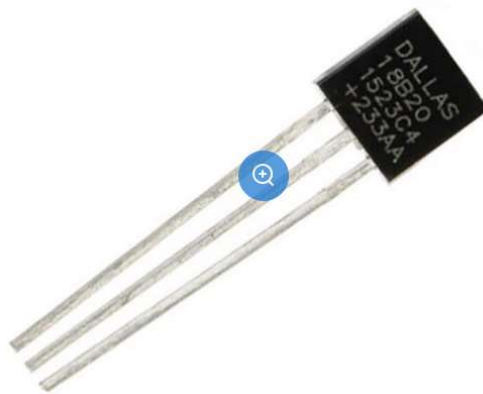


Рисунок 1.5 - Цифровий датчик температури DS18B20

Основна функція DS18B20 - перетворення температури датчика в цифровий код. Дозвіл перетворення задається 9, 10, 11 або 12 біт. Це відповідає розрізняювальній спроможності - 0,5 (1/2) °C, 0,25 (1/4) °C, 0,125 (1/8) °C і 0,0625 (1/16) °C. При включенні харчування, стан Регістру конфігурації встановлюється на роздільну здатність 12 біт.

Після включення живлення DS18B20 знаходиться в низько споживає стані спокою. Щоб ініціювати вимір температури майстер (мікроконтролер) повинен виконати команду перетворення температури [44h]. Після виконання перетворення результат вимірювання температури буде перебувати в 2 байтах Регістру температури, і датчик знову перейде в стан спокою.

Якщо DS18B20 включений по схемі із зовнішнім харчуванням, то цей майстер може контролювати стан команди конвертації. Для цього він повинен читати стан лінії (виконувати часовий слот читання), по завершенню команди, лінія перейде у високе стан. Під час виконання команди конвертації лінія утримується в низькому стані.

Температурний датчик DS18B20 має різноманітні види корпусу. Можна вибрати один з трьох - 8 PIN SO (150 mils), 8 PIN μ SOP, і 3 PIN to 92. Останній є найбільш поширеним і виготовляється у спеціальному вологозахисному корпусі, так що його сміливо можна використовувати під водою. У кожного датчика є 3 контакти. Для корпусу PIN to 92 потрібно дивитися на колір проводів: чорний - земля, червоний - живлення і білий / жовтий / синій - сигнал.

Пам'ять датчика складається з двох видів: оперативної та енергонезалежною – SRAM і EEPROM. В останню записуються регістри конфігурації та регістри TH, TL, які можуть використовуватись як регістри загального призначення, якщо не використовуються для вказівки діапазону допустимих значень температури.

1.5 Приєднання датчика температури

Перше, що буде потрібно - це приєднати датчик температури до GPIO Raspberry Pi 3. Після того, як фізичний підключення датчика температури до Raspberry Pi 3 було виконано, можна поставити певні модулі ядра. Їх всього 2. З-під команди (для цього ви обрали у консолі команду команди - і і ввести пароль) вимагається виконати пару команд: `modprobe W1 GRI` і `modprobe W1 therm`.

Перший модуль потрібен, щоб на порту GPIO4 активувався протокол 1 Wire. Другий же необхідний, щоб система могла безпосередньо зчитувати температурні показники з відповідною шини.

Після перезавантаження модулі знову будуть відключені, тому їх доведеться знову завантажувати. Але виправити це дуже легко. Потрібно всього лише вписати їх назви у файлі `/etc/modules`. Для цього з-під команди слід набрати команду `nano /etc/modules` і додати у кінці журналу запис `W1 GPIO` і `W1 therm` (кожен має розпочинати рядок), а потім зберегти зміни.

1.6 Інтерфейс 1-wire

Інтерфейс 1-wire розробила фірма Dallas Semiconductor, всі права на цей інтерфейс належать виключно цій фірмі (яка, втім, вже давно припинила своє самостійне існування і перейшла під крило корпорації MAXIM Inc.).

Чим привабливий цей інтерфейс? Зрозуміло, малою кількістю висновків МК, потрібних для підключення практично необмеженої кількості мікросхем. Двосторонній обмін вимагає всього 1 лінію! Крім того, асортимент пристроїв з цим інтерфейсом досить широким. Нарешті, протокол обміну з цього

інтерфейсу дуже простий і легко реалізується програмно практично на будь-яких МК хоча є і спеціальні мікросхемні адаптери).

1.6.1 Апаратна реалізація інтерфейсу 1-wire

На рисунку 1.6 можемо побачити апаратну реалізацію інтерфейсу 1-wire

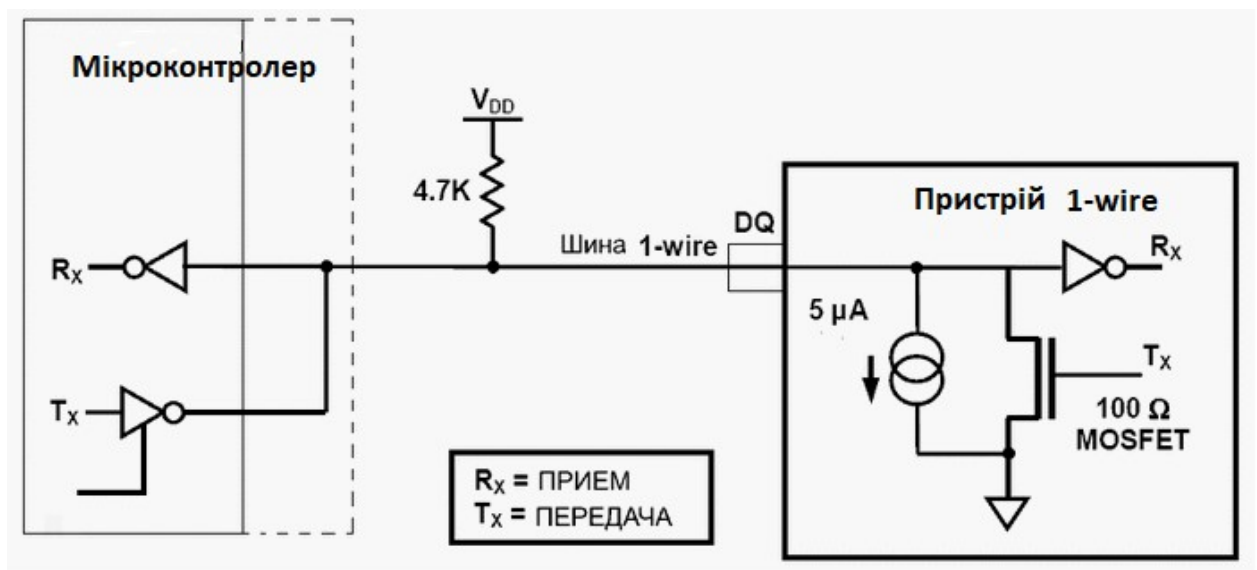


Рисунок 1.6 Апаратна реалізація інтерфейсу 1-wire

На рисунку 1.6 показана спрощена схема апаратній реалізації інтерфейсу 1-Wire. Висновок DQ пристрою являє собою вхід КМОП логічного елемента, який може бути розташований (замкнений на загальний провід) польовим транзистором. Опір каналу цього транзистора у відкритому стані - близько 100 Ом. Коли транзистор зачинений - мається невеликий струм витіку (приблизно 5 мкА) на загальний провід.

Шина 1-Wire повинна бути підтягнута окремим резистором до напруги живлення пристроїв (яке, до речі, може бути від 3 до 5В - уточнюється за характеристиками конкретного пристрою). Опір цього резистора 4,7 К, Однак,

це значення рекомендовано тільки для дуже коротких ліній. Якщо шина 1-Wire використовується для підключення віддалених на велику відстань пристроїв, то опір цього резистора слід зменшити. Мінімально допустима його опір - близько 300 Ом, а максимальне - близько пари-трійки десятків кіло. Дані величини - орієнтовні, ви завжди повинні уточнити за характеристиками конкретного пристрою 1-Wire його максимальний струм лінії DQ, який, власне, і визначає мінімум зовнішнього опору.

Підключення шини 1-Wire до МК показано умовно в двох варіантах: з використанням 2 окремих висновків МК (один у якості виходу, а інший в якості входу), так і одного, працюючого та на введення і на виведення. Поділ цих способів показано пунктирною лінією, що умовно позначає кордон корпусу МК. З деякою натяжкою можна уявити собі логічне будову шини 1-Wire як усім відоме з'єднання висновків мікросхем з відкритим колектором за схемою "монтажне або". Очевидно, що передача будь-якої інформації при цьому можлива тільки видачею низького рівня в лінію, тобто замиканням її на загальний провід, а у високий логічний рівень лінія повернеться сама, завдяки наявності зовнішнього підтягує резистора. Так само очевидно, що одночасна передача декількох пристроїв приречена на невдачу через повного спотворення інформації (усі що передаються одиниці одного пристрою будуть придушені, які передаються нулями від іншого пристрою).

1.6.2 Програмна реалізація інтерфейсу 1-wire

Основні постулати:

1. Обмін завжди ведеться за ініціативою одного ведучого пристрою, який в більшості випадків є мікроконтролером (МК).
2. Будь-який обмін інформацією починається з подачі імпульсу скидання ("Reset Pulse" або просто RESET) в лінію 1-Wire провідним пристроєм.

3. Для інтерфейсу 1-Wire у загальному випадку передбачається "гаряче" підключення і відключення пристроїв.

4. Будь-який пристрій, приєднаний до 1-Wire після отримання харчування видає в лінію DQ імпульс присутності, званий "Presence pulse" (далі я буду використовувати термін PRESENCE). Цей же імпульс пристрій завжди видає в лінію, якщо виявить сигнал RESET.

5. Поява в шині 1-Wire імпульсу PRESENCE після видачі RESET однозначно свідчить про наявності хоча б одного підключеного пристрою.

6. Обмін інформації ведеться так званими тайм слотами: один тайм - слот служить для обміну одним бітом інформації.

7. Дані передаються побайтово, біт за бітом, починаючи з молодшого біта. Достовірність переданих / прийнятих даних (перевірка відсутності спотворень) гарантується шляхом підрахунку циклічної контрольної суми.

На рисунку 1.7 показана діаграма сигналів Reset та PRESENCE, з яких завжди починається обмін даними. Видача імпульсу PRESENCE в процесі обміну служить для дострокового завершення процедури обміну інформацією.

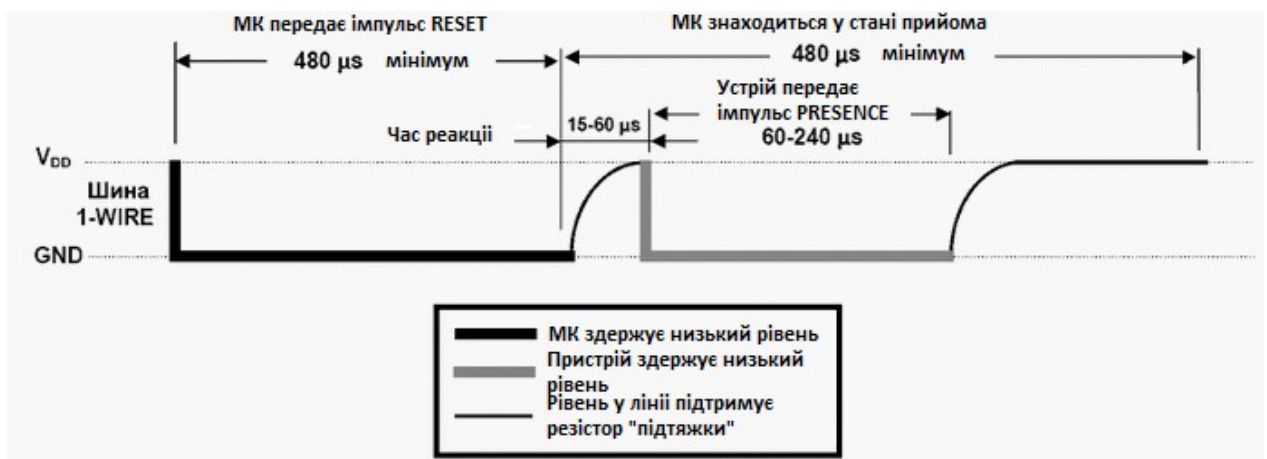


Рисунок 1.7 – Часова діаграма сигналів передачі даних по шині 1-wire

Імпульс RESET формує ведучий МК, переводячи в низький логічний рівень шину 1-Wire і утримуючи її в цьому стані мінімум 480 мікросекунд. Потім МК повинен "відпустити" шину. Через деякий час, залежне від ємкості лінії і опору підтягуючого резистора, в лінії встановиться високий логічний рівень. Протокол 1-Wire обмежує цей час "релаксації" діапазоном від 15 до 60 мікросекунд, що і є визначальним для вибору підтягуючого резистора (як правило, ємкість лінії ми міняти істотно не можемо, а саме вона робить істотний вплив на час повернення лінії до високого рівня).

Виявивши імпульс RESET, ведена пристрій приводить свої внутрішні вузли в початковий стан і формує у відповідь імпульс PRESENCE, як впливає з рисунка - не пізніше 60 мікросекунд після завершення імпульсу RESET. Для цього пристрій переводить в низький рівень лінію DQ і утримує її в цьому стані від 60 до 240 мікросекунд. Конкретний час утримання залежить від багатьох параметрів, але завжди знаходиться в заданому діапазоні. Після цього пристрій так само "відпускає" шину.

Але після завершення імпульсу PRESENCE влаштування дається ще деякий час для завершення внутрішніх процедур ініціалізації, таким чином, МК повинен приступити до будь-якого обміну з пристроєм не раніше, ніж через 480 мікросекунд після завершення імпульсу RESET.

Отже, процедура ініціалізації інтерфейсу, з якої починається будь-який обмін даними між пристроями, триває мінімум 960 мікросекунд, складається з передачі від МК сигналу RESET та приймання від пристрою сигналу PRESENCE. Якщо сигнал PRESENCE не виявлений - значить на шині 1-Wire немає готових до обміну пристроїв.

1.6.3 Принципи адресуванні пристроїв та управління ними.

Кожен пристрій 1-Wire володіє унікальним ідентифікаційним 64-бітним номером, програмованим на етапі виробництва мікросхеми. Унікальний - це означає, що фірма-виробник гарантує, що не знайдеться двох мікросхем з однаковим ідентифікаційним номером (принаймні в перебігу декількох десятиків років при існуючих темпах виробництва).

При розгляді протоколу обміну ми будемо виходити з принципу, що на шині 1-Wire є більше одного пристрою. В цьому випадку перед МК встають 2 проблеми: визначення кількості наявних пристроїв і вибір (адресація) однієї конкретної з них для обміну даними. Рішення першої проблеми здійснюється двома шляхами: універсальним і гнучким, але такою, що вимагає доволі складного програмно що реалізується алгоритму, і простим, але з великими обмеженнями. Універсальний алгоритм ми розглядати не будемо. Тож розглянемо більш простий, який полягає в тому, що номери всіх використовуваних у вашій схемі 1-Wire пристроїв ви повинні знати заздалегідь, і просто використовувати їх як константи у вашій програмі. Номери деяких пристроїв нанесені прямо на корпусі мікросхем (наприклад, для пристроїв iButton - всім відомих ключів таблеток), а номери інших можна визначити за допомогою спеціальних програм. iButton - зареєстрована торгова марка, права на яку належать Dallas Semiconductor.

Отже, Припустимо, що ми знаємо номери всіх пристроїв 1-Wire на шині. Як же ведеться робота з ними? Алгоритм тут наступний. МК посилає, як зазвичай, імпульс RESET, і всі наявні пристрої видають PRESENCE. Потім МК посилає в шину команду, яку приймають всі пристрої. Команд визначено декілька загальних для всіх типів 1-Wire пристроїв, а так само можуть бути команди, унікальні для окремих типів. Серед загальних команд нас в першу чергу можуть зацікавити наступні (див. таблицю 1).

Таблиця 1 – Система команд інтерфейсу 1-wire

Команда	Значення байту	Описання
SEARCH ROM	0xF0	Пошук адресов - використовується при універсальному алгоритму зазначення кількості та адресов підключених пристроїв
READ ROM	0x33	Читання адреси пристрою - використовується для зазначення адресу єдиного пристроя на шині
MATCH ROM	0x55	Вибір адреси - використовується для звернення к конкретному адресу пристроя з багатьох під'єднаних
SKIP ROM	0xCC	Ігнорувати адрес - використовується для звернення к єдиному пристрої на шині, при тому адреса пристрою ігнорується (можливо звертатися до невідомого пристрою)

Першу команду непотрібна для поточного використання, друга дозволить визначити адресу що є у пристроїв перед їхню установкою у готовий виріб, а дві останні напевно стануть основними у вашій роботі.

Після того, як МК видасть команду READ ROM, від пристрою надійде 8 байт його власного унікального адресу - МК повинен їх прийняти. Врахуйте, що будь-яка процедура обміну даними з пристроєм повинна бути завершена повністю або перервана посилкою сигналу RESET.

Якщо відправлена команда MATCH ROM, то після неї МК повинен передати так само і 8 байт конкретної адреси пристрою, з яким буде здійснюватися подальший обмін даними. Це рівносильно виставленню адреси на паралельній шині в мікропроцесорних пристроях. Приймавши цю команду, кожен пристрій порівнює переданий адрес з своїм власним. Всі пристрої, адреса яких не співпав, припиняють аналіз та видачу сигналів у лінії 1-Wire, а інші пристрої продовжують роботу. Тепер всі дані, що передаються МК будуть потрапляти тільки до цього "адресованого" пристрою. Те, які саме дані треба

послати в пристрій або отримати від нього після його адресуванні, залежить від конкретного пристрою (наприклад, для згаданого термометра це можуть бути команди запуску вимірювання або зчитування результату, для ключа таблетки не визначено жодних інших команд, Крім основних, а для мікросхем АЦП додаткових команд може бути близько десятка).

Якщо пристрій один на шині - можна прискорити процес взаємодії з ним за допомогою команди SKIP ROM. Повчивши цю команду, пристрій одразу вважає адресу яке співпало, хоча ніякого адреси за цією командою не слідує. Деякі процедури не вимагають прийому від пристрою жодних даних, в цьому випадку команду SKIP ROM можна використовувати для передачі якоїсь інформації відразу усіх пристроїв. Це можна використовувати, наприклад, для одночасного запуску циклу вимірювання температури кількома датчиками термостатами типу DS18S20.

Прийом і передачу байтів завжди починається з молодшого біта. Змінено порядок байтів при передачі і прийомі адреси пристрої так само ведеться від молодшого до старшого. Порядок передачі іншої інформації залежить від конкретного пристрою, тому слід звертатися до документації на застосовувані вами пристрою.

Декілька слів про так зване "паразитне питання" пристроїв 1-Wire. Специфікація цих пристроїв допускає їх живлення від самій лінії даних, тобто мікросхема може отримувати харчування за власною (і єдиною) лінії даних! Не всі пристрої підтримують такий режим харчування, але дуже багато, наприклад, той же термометр термостат DS18S20 або ключ - таблетка DS1990A. Тепер вам повинно бути зрозумілим, чому низький рівень сигналу в шині 1-Wire має такі жорсткі часові рамки - десятки мікросекунд. Це пов'язано з необхідністю забезпечувати харчуванням – ті пристрої, які отримують його від шини. Споживання струму всіх пристроїв 1-Wire таке нікчемне, що ємкості вбудованих конденсаторів достатньо для підтримки в

робочому стані схеми пристрою протягом невеликого інтервалу, коли в шині 1-Wire низький логічний рівень.

Слід пам'ятати, що при цьому способі, можливості пристроїв кілька зменшуються, що проявляється в погіршенні роботи на лінію з великою власної ємкістю (тобто лінію великої протяжності). Тому зв'язок з пристроями, які використовують цей спосіб, можлива на відносно невеликих відстанях.

Крім того, він часто супроводжується додатковими побічними ефектами, які описуються в документації на конкретні вироби.

2 РОЗРОБКА АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Розробка структурної схеми комп'ютерної системи

Структурна схема апаратного забезпечення комп'ютерної системи представлена на рисунку 2.1.

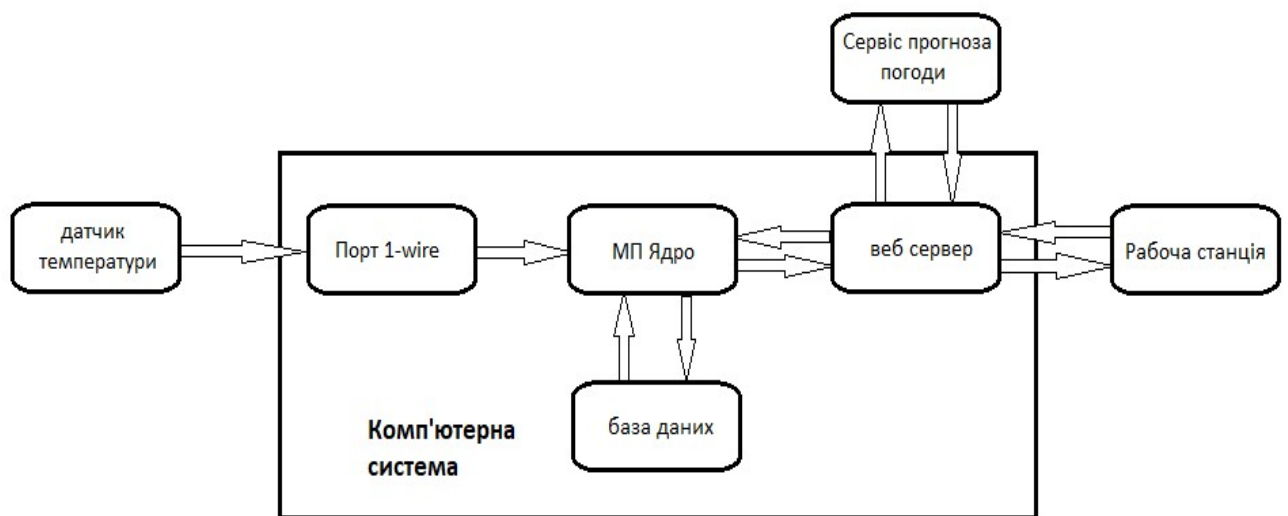


Рисунок 2.1 – Структурна схема апаратного забезпечення комп'ютерної системи

Система працює наступним чином.

Фактично в системі наявні декілька каналів передачі та обробки даних. Перший канал пов'язаний з зчитуванням даних з датчика температури та занесенням даних до бази даних з визначеним дискретом за часом. Другий канал, пов'язаний з отриманням інформації стосовно поточного прогнозу температури атмосфери з онлайн-ресурсу з використанням відповідного API. І третій канал, пов'язаний з формуванням запиту до серверної частини з

робочої станції, при якому з бази даних отримується поточне значення температури, поєднується з даними з онлайн-ресурсу та розміщується на веб-сторінці. Таким чином, клієнт в своєму браузері отримує з сервера згенеровану сторінку з даними про поточну температуру з датчика та прогнозовану з інтернет-ресурсу.

Для зменшення кількості провідників що пов'язують датчик температури з комп'ютерною системою, використовується інтерфейс 1-wire. Що дозволяє отримувати компактне підключення датчика на віддалених дистанціях від комп'ютерної системи з використанням негабаритного інтерфейсного кабелю. Мікропроцесорне ядро відповідно до відпрацьовування вбудованого інтервального таймеру, здійснює циклічне опитування інтерфейсу 1-wire до якого підключений датчик. Отримане цифрове значення температури додається до бази даних, яка розташована локально на комп'ютерній системі. Відповідно, на цієї же системі розташовується система управління базами даних, яка, в свою чергу, поєднана даними авторизації з односторінковим сайтом-інформером.

Отримання даних про температурний режим з інтернет-ресурсу здійснюється за запитом, який надходить від клієнта до веб-сервера. В цьому випадку, здійснюється отримання даних з віддаленого сайту з використанням API. Крім цього, веб-сервер формує запит до серверу бази даних на отримання з таблиці температури даних для відображення. Ці данні використовуються для генерації сторінки сайту яка і надається клієнтові.

2.2 Розробка принципової схеми апаратного забезпечення комп'ютерної системи

В якості датчика температури використовується цифровий датчик DS18B20. Типовий варіант підключення цього датчику до шини 1-wire представлений на рисунку 2.2.

Резистор номіналом 4.7кОм використовується для підтягування лінії передачі даних до високого рівня, що відпоідає напрузі живелння 3.3В. В цьожму випадку, розмах цифрового сигналу при передачі даних буде мати максимальне значення та забезпечувати максимальну перешкодостійкість та інваріантність до незначних просаджень напруги, що може бути наслідком імпульсного споживання струму. В якості входу отримання даних в послідовному форматі використовується цифровий інтерфейсний вивід GPIO4 одноплатного комп'ютера RaspberryPI.

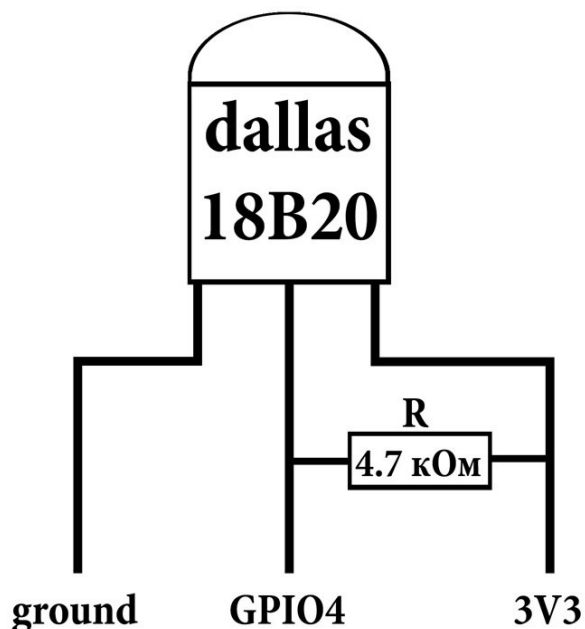


Рисунок 2.2 – Схема підключення цифрового датчику DS18B20 до інтерфейсу 1-wire

Схема під'єднання датчику до комп'ютерної системи наведена на рисунку 2.3.

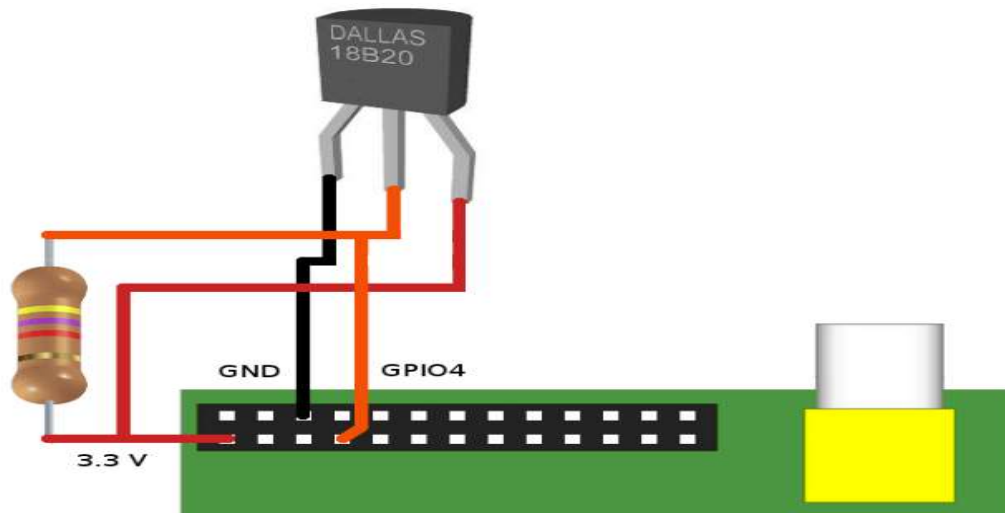


Рисунок 2.3 – Схема підключення цифрового датчику DS18B20 до комп'ютерної системи

Використання цифрових датчиків температури на Raspberry реалізовано з використанням стандартних бібліотек підтримки комунікаційної периферійного інтерфейсу GPIO.

Для реалізації програмної підтримки отримання даних з датчику треба завантажити два модулі ядра. Модулі завантажуються командами:

```
sudo modprobe w1-gpio
```

```
sudo modprobe w1-therm
```

Перша команда активує протокол модуля 1-wire на GPIO4. Друга загрузає модуль для того щоб зчитати температуру з самої шини. Для того, щоб модулі автоматично завантажувалися після перезавантаження або вимкнення Raspberry Pi, дві вищенаведені команди треба додати у файл /etc/modules. Зробити це можна наступним чином:

```
nano /etc/modules
```

У вікні редактору цього файлу додаємо у кінці журналу

W1 GPIO

W1 therm

Таким чином читання поточних температури зводиться до читання спеціального файлу. Цей файл розташований в каталозі `/sys/bus/w1/devices`. Даний каталог містить один або кілька підкаталогів. Один, який завжди присутній, називається `w1_bus_master1`. Але нас він не цікавить. Необхідні каталоги що мають ім'я вигляду `28-*****`. Для кожного підключеного 1-Wire датчика температури створюється свій такий каталог. Може потрібно декілька секунд для їх появи. Назву кожного каталогу - це унікальний серійний номер датчика `ds18b20`. Якщо ви підключили понад одного датчика по шині, вам доведеться з'ясувати який каталог з серійним номером якого датчику належить. Зробити це можна шляхом прогрівання одного датчика, зчитування температури з усіх датчиків і перевірки температура якого саме змінюється. Або можна підключати датчики по черзі і записувати, який каталог якому датчику відповідає.

Обмін інформацією ведеться тайм-слотами. один тайм-слот служить для обміну одним бітом інформації.

При обміні інформацією ведучий ініціює кожний зв'язок на бітному рівні. Передача кожного біта, незалежно від напрямку (передача чи прийом), повинна бути ініційована ведучим. Шина даних за замовчуванням підтягується до "одиниці", тому для початку як прийому, так і для передачі ведучий опускає лінію на деякий час в "нуль".

Розглянемо 3 основних команд для зв'язку по шині 1-Wire: "Запис 1", "Запис 0", "Читання".

На рисунках 2.4-2.6 червоним виділено управління лінією від ведучого, синім - управління лінією від веденого, чорним - звільнена лінія (за допомогою підтягування шина автоматично переходить в "одиницю").

Ведучий встановлює низький рівень протягом 1 ... 15 мкс. Після цього, протягом решти часового слоту він звільняє шину. Розглянемо сигнал-Запис-1 на рисунку 2.4



Рисунок 2.4 Сигнал Запис-1

Ведучий встановлює низький рівень протягом 1 ... 15 мкс. Після цього, протягом решти тимчасового слоту він звільняє шину.

Сигнал "Запис 0" рисунок 2.5. Ведучий формує низький рівень протягом не менше 60 мкс, але не довше 120 мкс.



Рисунок 2.5 Сигнал Запис-0

Сигнал "Читання" рисунок 2.6. Ведучий встановлює низький рівень протягом 1 ... 15 мкс. Після цього ведений, якщо хоче передати 0, утримує шину в низькому стані до 60 мкс; якщо ж ведений хоче передати 1, то він просто звільняє лінію. Ведучий зазвичай сканує стан шини після закінчення 15 мкс після встановлення низького рівня на шині.



Рисунок 2.6 Сигнал Читання

Так, ведений утримує лінію до землі, якщо хоче передати "0", і просто відпускає лінію, якщо хоче передати "1".

Основні проблеми, які виникають при реалізації читання-запису - це проблеми з часом, тобто "невитримування", або навпаки, "перевитримування" тимчасових затримок при читанні лінії. Виникають ці проблеми через те, що часто не роблять поправку на затримку виконання команд мов програмування високого рівня. Особливо це стосується різних "додаткових" функцій.

Сигнал "скидання/присутність". Тут тимчасові інтервали імпульсів відрізняються. Ведучий встановлює низький рівень протягом 8 тимчасових слотів (480 мкс), а потім звільняє шину. Даний тривалий період низького стану називається сигналом "скидання".

Якщо на шині присутній ведений, то він повинен протягом 60 мкс після звільнення ведучим шини встановити низький рівень тривалістю не менше 60 мкс. Даний відгук носить назву "присутність". Якщо такий сигнал не виявляється, то ведучий повинен думати, що немає підключених пристроїв до шини і подальший зв'язок неможливий.

Дана послідовність сигналів завжди починає будь-який обмін інформацією між пристроями.

Крім цього, потрібно враховувати, що будь-який ведений пристрій після підключення живлення відразу ж видає сигнал присутності.

Сигнал "скидання" дозволяє ведучому достроково завершити обмін інформацією - наприклад, якщо сенсор температури передає нам всю свою пам'ять, а нам потрібні тільки перші два байти, які містять значення температури, то після отримання цих двох байт мікросхема просто може опустити лінію в нуль на потрібну тривалість часу - сенсор зрозуміє, що більше нічого пересилати не треба.

При реалізації зазвичай необхідно запрограмувати мікроконтролер чи мікрокомп'ютер як ведучий пристрій, тому мікрокомп'ютеру варто генерувати сигнали тривалості трохи більшої необхідного мінімуму, а відповіді від ведених пристроїв чекати за найгіршими прогнозами - тоді взаємодія по протоколу буде оптимальною.

При цьому ведучому треба не забувати періодично перевіряти стан лінії даних - те, що вона повертається в підтягнутий стан з плином часу - а то може виникнути ситуація, наприклад, що який-небудь ведений зламався і закорочує, наприклад, лінію в нуль – і, в принципі, протокол не перевіряє дану проблему сам, хоча помилка може і не виникнути.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Програмне забезпечення контролю температури.

Підключаємося до mysql:

```
mysql -u root -p
```

вказуємо пароль, який вписувався при установці mysql.

Створюємо базу з ім'ям sensor_db:

```
mysql > Create Database sensor_db Character Set utf8 collate utf8_unicode_ci;
```

Створюємо користувача бази з логіном pi і паролем raspberry0, який буде мати повні права на базу sensor_db:

```
mysql > Grant All PRIVILEGES on sensor_db.* to pi@localhost identified by "raspberry0";
```

```
mysql > Grant All PRIVILEGES on sensor_db.* to "Pi" @'%' identified by "raspberry0";
```

```
mysql > Flush PRIVILEGES;
```

```
mysql > exit;
```

Підключаємося до mysql правами користувача pi і паролем raspberry0:

```
mysql -u pi -p
```

Enter password: вписуємо пароль для користувача-адміністратора Raspberry PI

Робимо базу sensor_db поточною:

```
mysql > use sensor_db;
```

Створюємо таблицю sensors001:

```
mysql > Create Table sensors001 (Place varchar (50), sensdata varchar (30), modelsens varchar (30), pname varchar (30), datetime datetime, other varchar (50));
```

В вищенаведеному SQL-скрипті:

pcname - у цей полі будемо заносити ім'я ПК, до якого приєднано датчик.

Максимальна довжина = 30;

datetime - поле для дати і часу спрацьовування датчиків;

modelsens - поле для моделі датчика. Максимальна довжина = 30;

place - поле для зазначення місця розташування датчика (кімната / поверсі / гараж і т. Д.). Максимальна довжина = 50;

sensdata – данні про температуру, що отримуються з відповідного датчика. Максимальна довжина = 30;

other - додаткове поле для пояснень.

Дивимося створене:

```
mysql > Show y;
```

```
--
```

```
Tables_in_sensor_db
```

```
--
```

```
sensors001
```

```
--
```

```
1 row in set (0.01 sec)
```

і ще дивимося:

```
mysql > describe sensors001;
```

```
-----
```

```
Field Type Null Key default Extra
```

```
-----
```

```
Place varchar (50) YES Null
```

```
sensdata varchar (30) YES Null
```

```
modelsens varchar (30) YES Null
```

```
pcname varchar (30) YES Null
```

```
datetime datetime YES Null
```

other varchar (50) YES Null

6 rows in set (0.01 sec)

Створимо ще одну колонку, який буде автоматично генеруватися і буде
PRIMARY KEY:

```
mysql > Alter Table sensors001 Add ID int not Null auto_increment First, Add  
Primary Key (ID);
```

```
mysql > Set Names utf8 collate utf8_unicode_ci;
```

```
mysql > describe sensors001;
```

Field	Type	Null	Key	default	Extra
-------	------	------	-----	---------	-------

ID	int (11)	No	PRI		auto_increment
----	----------	----	-----	--	----------------

Place	varchar (50)	YES			Null
-------	--------------	-----	--	--	------

sensdata	varchar (30)	YES			Null
----------	--------------	-----	--	--	------

modelsens	varchar (30)	YES			Null
-----------	--------------	-----	--	--	------

pcname	varchar (30)	YES			Null
--------	--------------	-----	--	--	------

datetime	datetime	YES			Null
----------	----------	-----	--	--	------

other	varchar (50)	YES			Null
-------	--------------	-----	--	--	------

7 rows in set (0.01 sec)

Встановимо bc:

```
apt - get install bc
```

Створимо скрипт:

```
mkdir -p /script/gpio/
```

Змінюємо права доступу до скрипта:

```
chmod -R 0777 ./script/gpio/
```

Преходимо до редагування вмісту файлу скрипту:

nano /script/gpio/sensors001.sh

Напишемо туди:

```
*****
```

```
#!/bin/bash
# Вказуємо свій сенсор
sens01="28-03155178d4ff"
# знімаємо температуру з датчика 28 03155178d4ff

#TemperatureFull=$(cat /sys/bus/w1/devices/28-03155178d4ff/w1_slave |
grep "t=" | cut -f 2 -d "=")
TemperatureFull=$(cat /sys/bus/w1/devices/$sens01/w1_slave | grep "t=" |
cut -f 2 -d "=")

#TemperatureC=$(echo "scale=1; $TemperatureFull*0.001" | bc -l); echo
$TemperatureC | cut -c 1-4
TemperatureC=$(echo "scale=1; $TemperatureFull*0.001" | bc -l);
TempC=$(echo $TemperatureC | cut -c 1-4);
#echo $TempC

#TemperatureC=$(echo "scale=0; $TemperatureFull*0.001" | bc -l); echo
${TemperatureC%.*}

mysql -D sensor_db -u pi -praspberry0 << EOF
USE sensor_db;
SET NAMES utf8 COLLATE utf8_unicode_ci;
INSERT INTO sensors001 VALUES ('','кімната №1',$TempC,'dallas
18B20','raspi001',NOW());

EOF
#echo "$TempC занесено в базу"
```

```
*****
```

Змінюємо права файлу для можливості його виконання:

chmod +x /script/gpio/sensors001.sh

Після запуску скрипту дані з датчика записуються в базу. Подивимося, як вони там виглядають. Виконуємо в консолі, не в клієнті mysql (між - р і паролем raspberry0 пробілу бути не повинно!):

```
mysql - d sensor_db -u pi -p raspberry0 -e "select * from sensors001";
```

3.2 Програмне забезпечення веб-сервера

Веб-сторінка для даного проекту була зроблена на базі React JS.

Поперше зробимо компоненти.

Form.js:

```
import React from "react";
```

```
class Form extends React.Component{
```

```
  render(){
```

```
    return(
```

```
      <form onSubmit={this.props.weatherMethod}>
```

```
        <input name="city" placeholder="Город"/>
```

```
        <button>Получить погоду</button>
```

```
      </form>
```

```
    );
```

```
  }
```

```
}
```

```
export default Form;
```

info.js:

```
import React from "react";
```

```
class Info extends React.Component {
```

```
  render() {
```

```
    return(
```

```
      <div>
```

```
        <h2>Погодное приложение</h2>
```

```
      </div>
```

```
    );
```

```
  }
```

```
}
```

```
export default Info;
```

weather.js:

```
import React from "react";
```

```
class Weather extends React.Component {
```

```
render(){  
  
  return(  
  
    <div>  
      <p>Страна: {this.props.country}</p>  
      <p>Город: {this.props.city}</p>  
      <p>Температура по сайту: {this.props.temp}</p>  
      <p>Температура с датчика : {this.props.sendsdata}</p>  
  
    </div>  
  
  );  
}  
}
```

export default Weather;

App.js:

```
import React from "react";  
import Info from "./components/info";  
import Form from "./components/form";  
import Weather from "./components/weather";  
  
const API_KEY = "7a002d5d28388e84c97374bbca9389a9"  
const connection = require("mysql");
```

```
const connection = connection.createConnection({
  host: "localhost",
  user: "pi",
  password: "raspberry0",
  database: "sensors001"
});
```

```
connection.connect(err =>{
  If(err){
    Return err;
  }
})
```

```
class App extends React.Component{
```

```
  state = {
    temp: undefined,
    city: undefined,
    country: undefined,
    sensdata: undefined,
    error: undefined
  }
```

```
  gettingWeather = async (e) => {
    e.preventDefault();
    const city = e.target.elements.city.value;
```



```
    const api_url = await
fetch(`https://api.openweathermap.org/data/2.5/weather?q=${city}&appid=${API_KEY}&units=metric`);
    const data = await api_url.json();
    const temp2 = connection.query(SELECT_ALL_FROM_SENSDATA);
    console.log(data);

    this.setState(
      {
        temp: data.main.temp,
        sensdata: temp2,
        city: data.name,
        country: data.sys.country,
        error: ""
      }
    );
  }
}
```

```
render(){
  return(
    <div>
      <Info />
      <Form weatherMethod={this.gettingWeather}/>
      <Weather
```

```
    temp={this.state.temp}
    city={this.state.city}
    country={this.state.country}
    error={this.state.error}
  />
</div>
);
}
}

export default App;
```

Далі потрібно зробити «build» проекту. Для цього в консолі потрібно прописати:

- якщо використовується npm, то виконується команда `npm run build`
- якщо yarn , то – `yarn build`.

ВИСНОВКИ

В ході виконання дипломної роботи було розроблено інформаційну комп'ютерну систему моніторингу температури навколишнього середовища на основі даних прогнозу погоди з сайту openweathermap.org, а також шляхом прямого вимірювання температури за допомогою цифрового датчика температури з інтерфейсом 1-wire.

Платформою для реалізації web-сервера, а також контролера з'явлення датчика було використано одноплатний комп'ютер Raspberry Pi з Linux-подібною операційною системою Raspbian (на основі Debian).

Комп'ютерна система здійснює моніторинг поточної температури з датчика, та у випадку запиту до web-серверу, генерує сторінку з даними про температуру з локальної бази даних та даних з онлайн-сервісу погоди.

Розроблена система може використовуватися як окремий пристрій, з виділеною IP-адресою, а також буди елементом вбудованої комп'ютерної системи, яка окрім розробленого функціоналу, дозволяє виконувати інші функції.

ПЕРЕЛІК ЛІТЕРАТУРИ

1. Raspberry Pi - <https://academicfox.com/ru/raspberry-pi-ds18b20-2/>
2. Що таке температура - <https://bagan.ru/articles/294940>.
3. Raspberry Pi. Сборник рецептов. Решение программных и аппаратных задач
4. The Official Raspberry Pi Projects Book volume 2
5. Raspberry Pi For Kids For Dummies
6. Измерение температуры поверхности твердых тел:
<https://eknigi.org/apparatura/84834-izmerenie-temperaturey-poverxnosti-tverdyx-tel.html>
7. Методы измерения температуры-
https://eknigi.org/estestvennye_nauki/128693-metody-izmereniya-temperaturey-chast-1.html
8. Работа сетей 1-wire-
https://elin.ru/files/pdf/1-Wire/app148ru_p.pdf