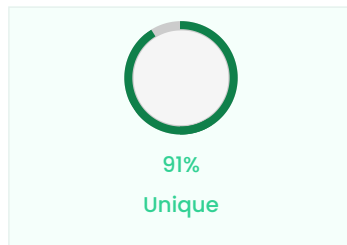
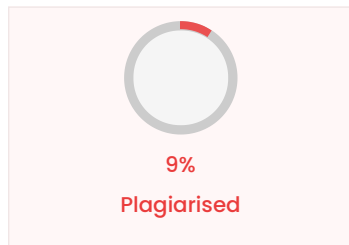


Plagiarism Scan Report



Words Statistics

Words	8070
Characters	60717

Exclude URL: None

Content Checked For Plagiarism

6 МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ПрАТ «ПРИВАТНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД «ЗАПОРІЗЬКИЙ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ» Кафедра комп'ютерної інженерії ДО ЗАХИСТУ ДОПУЩЕНИЙ Зав. кафедри _____ Переверзев А.В. ДИПЛОМНА РОБОТА ТЕМА: «Розробка системи моніторингу стану комп'ютера» Е _____ Є. О. Сілін Керівник викладач _____ К. С. Суха Запоріжжя 2022 Пр НАВЧАЛЬНИЙ ЗАКЛАД «ЗАПОРІЗЬКИЙ ІНСТИТУТ ЕКОНОМІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ» Кафед ЗАТВЕРДЖУЮ Зав. кафедрою д.т.н., проф. А.В. Переверзев _____ 13.03.2022 р. З А В Д А Н Н МОЛОДШОГО СПЕЦІАЛІСТА Студенту гр. КІ-118к9, спеціальності 123 «Комп'ютерна інженерія» Сілін Євген Олекс системи моніторингу стану комп'ютера затверджена наказом по інституту 09.2-17 від 12 березня 2022 р. 2. Термін роботи: 19 червня 2022 р. 3. Перелік питань, що підлягають розробці: 1.Вивчити літературу, присвячену темі розр області та виявлення основних параметрів а майбутньої системи. 3. Огляд існуючих аналогів з виявленням переваг власного проекту. . 4. Вибір та обґрунтування програмно-апаратного рішення. 5.Виконати монтаж компоне оптимальні режими роботи пристрою 6.Запрограмувати пристрій та відлагодити його 7. Оформити результати р видачі завдання: 13 березня 2022 р. Керівник випускної роботи _____ К. С. Суха (підпис) Завда _____ Є.О. Сілін (підпис студента) РЕФЕРАТ Пояснювальна записка складається з: 71 сторінок, 3 джерела. Об'єкт дослідження – «Системи моніторингу стану комп'ютера на платформі Arduino». Мета роботи зручного спостереження за характеристиками комп'ютера. Методи дослідження – проектування, моделювання та п системи. Обрано апаратну частину проекту, що складається з плати Arduino Nano, одного LCD дисплею 2004, дат комп'ютерного кулера, RGB стрічки, MOSFET транзисторів, резисторів, макетної плати та molex роз'єм для живлення для управління мікроконтроллером. Створено робочий макет «Системи моніторингу апаратної частини ПК на пла бути використано для подальшого застосування на практиці. ARDUINO, ARDUINO IDE, КОМП'ЮТЕР, МІКРОК УТИЛІТА ЗМІСТ СПИСОК УМОВИХ ПІЗНАЧЕНЬ.....6 ВСТУП.....8 ПРЕДМЕТНОЇ ГАЛУЗІ ТА РОЗГЛЯД ПРОЕКТУ «Розробка системи моніторингу стану комп'ютера».....8 моніторингу8 1.2 Обладнання для монітору стану заліза.....8 1.3 Принцип системи.....14 1.4 Висновки за розділом.....14 РОЗДІЛ 2 ВИБІР ТА ОБГРУ АПАРАТНОГО КОМПЛЕКСУ.....15 2.1 Вибір мікроконтролеру та платформи..... середовища розробки та мови програмування.....34 2.2.1 Середовище розробки..... програмування.....39 2.3 Висновок за розділом.....43 РОЗДІЛ 3 МОНІТОРІНГУ.....44 3.1 Проектування апаратного комплексу проекту.....44 3.2 Тест системи.....50 3.3 Висновок за розділом.....52 ВИСНОВОК..... ПОСИЛАНЬ.....54 ДОДАТКИ.....55 СПИСОК УМОВ словосполучення Скорочення Широтно-імпульсна модуляція ШИМ Програмне забезпечення ПЗ Персональний кс напівпровідникові польові транзистори (англ. Metal Oxide Semiconductor Field Effect Transistors) MOSFET Черв Red green blue) RGB Центральний процесор (англ. central processing unit) CPU Графічний процесор (англ. с

GPU Оперативна пам'ять (англ. Random-access memory) RAM Рідкокристалічний дисплей (англ. liquid crystal display) Мікроінтегральна схема (англ. Integrated Circuit) ІС ВСТУП У нас час дуже важлива необхідності ретельного контролю температури заліза. Це пов'язано з тим, що стабільність роботи комп'ютера загалом, як і довговічність безпосередньо залежить від своїх робочої температури. Робоча температура у свою чергу пов'язана з комбінацією властивостей пристроїв: їх енергоспоживанням, тепловиділенням, робочою напругою, частотою. А збільшення частоти тепловиділення. Дипломна робота складається з виведення загальної інформації роботи пристроїв ПК безпосередньо температури та завантаженість елементів ПК. Цей пристрій допоможе вам бачити Рівень завантаження: CPU температуру у корпусі комп'ютера та запобігти поломці комп'ютера або окремих комплектуючих. Для цього ще будуть потрібні навантаження за допомогою RGB стрічки і для запобігання перегріву керування кулерами.

РОЗДІЛ 1 АНАЛІЗ ПРОЄКТУ «Розробка системи моніторингу стану комп'ютера» 1.1 Огляд загальних характеристик функцій процесора (CPU) — функціональна частина комп'ютера, що призначена для інтерпретації команд. Рисунок 1.1 – Цей процесор обробляє дані по заданій програмі шляхом виконання арифметичних і логічних операцій; — програмне керування комп'ютера. Архітектура системи команд. З погляду програмістів, під архітектурою процесора мається на увазі певний набір машинних кодів. Більшість сучасних десктопних процесорів відносяться до сімейства x86, або архітектури IA32 (архітектура 32-бітових процесорів Intel). Її основа була закладена компанією Intel в процесорі Pentium. У наступних поколіннях процесорів вона була доповнена і розширена як самою Intel (введені нові набори команд MMX, SSE, SSE2), так і виробниками (набори команд EMMLX, 3DNow! і Extended 3DNow!, розроблені компанією AMD). Ядро. В рамках однієї архітектури процесори можуть досить сильно відрізнятися один від одного. І відмінності ці утілюються в різноманітних програмних наборах певний набір суворо обумовлених характеристик. Найчастіше ці відмінності втілюються в різних частотах системи кешу другого рівня, підтримці тих або інших нових систем команд або технологічних процесах, за якими виготовляються процесори. Зміна ядра в одному і тому ж сімействі процесорів спричиняє за собою заміну процесорного роз'єму (сокет, англ. socket) і питання подальшої сумісності материнських плат. Проте в процесі вдосконалення ядра виробникам доводиться здійснювати зміни, які не можуть претендувати на "власне ім'я". Такі зміни називаються ревізіями (англ. stepping) ядра і, зазвичай, позначаються цифро-буквеними комбінаціями. Проте в нових ревізіях одного і того ж ядра можуть зустрічатися досить помітні нові зміни. Intel ввела підтримку 64-бітової архітектури EM64T в окремі процесори сімейства Pentium 4 саме в процесі ревізії ядра.1.2 Процесор (GPU) — окремий пристрій персонального комп'ютера або ігрової приставки, виконує графічний процесор. Процесори дуже ефективно обробляють і зображують комп'ютерну графіку, завдяки спеціалізованій конвеєрній архітектурі, ефективніші в обробці графічної інформації, ніж типовий центральний процесор. Графічний процесор в основному використовується як прискорювач тривимірної графіки, але в деяких випадках його можна використовувати і для обчислювальних операцій. Особливостями в порівнянні із CPU є: архітектура, максимально націлена на збільшення швидкості обчислень складних графічних об'єктів, обмежений список команд. Рисунок 1.2 – Графічний процесор Оперативна пам'ять (RAM) пам'ять EOM, призначена для зберігання коду та даних програм під час їхнього виконання. У сучасних комп'ютерах переважно представлена динамічною пам'яттю з довільним доступом DRAM. Рисунок 1.3 – Оперативна пам'ять оперативної пам'яті Протилежністю до пам'яті з довільним доступом є пам'ять з послідовним доступом. При організації пам'яті таким чином, що в будь-яку мить можна отримати значення, записане в будь-якій комірці пам'яті, не потрібно прокручувати стрічку, зчитуючи інші елементи. Види пам'яті з довільним доступом: Напівпровідникова статична пам'ять (SRAM) напівпровідникові тригери. Переваги — невелике енергоспоживання, висока швидкодія. Відсутність необхідності періодичного оновлення. Недоліки — малий обсяг, висока вартість. Нині широко використовується як кеш-пам'ять процесорів у комп'ютерах. Динамічна (DRAM) — кожна комірочка є конденсатором на основі переходу КМОП-транзистора. Переваги — низьке енергоспоживання. Недоліки — необхідність періодичного прочитування і перезапису кожної комірки — т.з. «регенерації», і, як наслідок, велике енергоспоживання. Процес регенерації реалізується спеціальним контролером, встановленим на материнській платі центральному процесорі. DRAM зазвичай використовується як оперативна пам'ять комп'ютерів. Феромагнітна пам'ять (FRAM) — є типом пам'яті, в якій перетини яких знаходяться кільця або біакси, виготовлені з феромагнітних матеріалів. Переваги — стійкість до радіації при виключенні живлення; недоліки — мала ємність, велика вага, стирання інформації при кожному читанні. В наслідок цього дискретних компонентів вигляді, не застосовується. Проте до 2003 року з'явилася магніторезистивна оперативна пам'ять (MRAM) інтегральному виконанні. Поєднуючи швидкість SRAM і можливість зберігання інформації при відімкненому живленні, вона заміною типам ROM і RAM. Проте вона приблизно удвічі дорожча за мікросхеми SRAM (при тій же ємності і габаритах).1.4 MSI AFTERBURNER MSI AFTERBURNER — найпопулярніша в світі утиліта для моніторингу та повного контролю над системою. Вона надасть максимально повну інформацію про стан вашої системи, дасть змогу зберігати різноманітні комбінації параметрів, інтегрований бенчмарк та можливість захоплення відеопотоку. Рисунок 1.4 – Інформація про GPU у MSI AFTERBURNER це спрощена назва контролера, що управляє швидкістю руху кулерів та вентиляторів систем охолодження електричних компонентів.

контролери оснащені додатковими корисними функціями, як моніторинг температури систем і підсистем, і конкретної моделі. Як правило, реобаси встановлюються в порт 5.25", але можливе встановлення і в порт 3.5". Існує такого роду – з додатковими виводами USB, аудіовходами та аксесуарами. Рисунок 1.5 – Реобас DEXP CR – розмірами 149x100x42мм. Монтується в стандартний відсік системного блоку 5.25". Їм часто замінюють марний розрахована на керування роботою двох вентиляторів. На фронтальну панель виведено два механічні регуля обертання відповідних вентиляторів. 1.3 Принципова схема функціонування системи Рисунок 1.5 – Принципова схема функціонування системи складається з зчитування даних з комп'ютера та передачу до Arduino та відображаються на дисплеї, а також навантаження передається через колір на RGB стрічці.

1.4 Висновки за предметної області створення системи моніторингу апаратної частини ПК встановлено, що створення цього монітору модифікування свого ПК, для спостереження за станом температури, керуванням та використанням та коригування комп'ютера. та для покращення своїх навичок в програмуванні. Для покращення спостереження за температурою інформація важлива для того, щоб дізнатися при перегріві комплектуючих.

РОЗДІЛ 2 ВИБІР ТА ОБҐРУНТУВАННЯ АПАРАТНОГО КОМПЛЕКСУ

2.1 Вибір мікроконтролера та платформи Nano – одна з найбільш мініатюрних плат аналогом Arduino Uno – так само працює на чіпі ATmega328P (хоча можна ще зустріти варіанти з ATmega168P) фактором. Через своїх габаритних розмірів плата часто використовується в проектах, в яких важлива компактність. Плату живлення Arduino працює через USB (miniUSB або microUSB). В іншому параметри збігаються з Arduino Uno. Рисунок 2.1 – Arduino NANO V3 Рисунок 2.2 – Опис плати Arduino NANO Технічні характеристики Arduino Nano Характеристики Arduino Nano - * живлення 7–12В (рекомендований); - * кількість цифрових пинов – 14, з них 6 можуть використовуватися як аналогових входів; - * максимальний струм цифрового виходу 40 мА; - * флеш пам'ять 16 Кб або 32 Кб, в залежності від чіпа; - * EEPROM 512 байт або 1 Кб; - * частота 16 МГц; - * розміри 19 x 42 мм; - * вага 7 м; Живлення двома способами: 1. Через mini-USB або microUSB при підключенні до комп'ютера; 2. Через зовнішнє джерело живлення 20 В з низьким рівнем пульсацій. Стабілізація зовнішнього джерела виконується за допомогою схеми LM1117IMPX – через кабель від комп'ютера підключення до стабілізатора відбувається через діод Шоттки. Схеми обох типів харчування Arduino Nano. Рисунок 2.3 – Схема Arduino NANO При підключенні двох джерел напруги плата вибирає з найбільшим харчуванням є такі ж обмеження по нарузі і струму на входи і виходи плати. Всі цифрові і аналогові контакти працюють в діапазоні живлення, що виходить за рамки цих значень, напруга буде обмежуватися захисними діодами. У цьому випадку струм контролюється резистором, щоб не вивести контролер з ладу. Найбільше значення впадає або впливає струму не повинно бути більше 200 мА, а загальний струм контактів повинен бути не більше 200 мА. На платі є 4 світлодіоди, які показують стан сигналів TX і RX. На перших двох світлодіод загоряється, коли рівень сигналу низький, і показує, що сигнал TX або RX загоряється при нарузі в 5 В і показує, що підключено харчування. Останній світлодіод – загальний сигнал подається високий сигнал. На даний момент випускається декілька видів Arduino Nano. Є версії 2.X, 3.0., Які відмінно працюють. У версії 2.X. використовується чіп ATmega168 з меншим об'ємом пам'яті (флеш, незалежно від частоти, версія 3.0. працює на чіпі ATmega328. Терморегулятори Arduino Nano. Плата Arduino Nano має позначаються літерою D (цифровий, digital). Контакти використовуються як входи і виходи, у кожного є підтягуючі резистори. Показ контактів і пінів в Arduino NANO Аналогові Піни позначаються літерою A і використовуються як входи. Резистори, вони вимірюють подане на них напруга і повертають значення за допомогою функції analogRead (). Можна побачити значок ~. Такі контакти можуть бути легко використовувати в якості виходів ШИМ. Arduino Nano контактами – це Піни D3, D5, D6, D9, D10, D11. Для роботи з ШИМ висновками використовується функція digitalWrite(). Arduino Nano: - * цифрові входи / виходи: D0–D13; - * аналогові входи / виходи: A0–A7 (10–розрядний АЦП); - * ШИМ: Піни 3, 5, 6, 9, 10, 11, 12, 13 (TX і RX відповідно); - * I2C: SDA – A4, SCL – A5; - * SPI: MOSI – 11, MISO – 12, SCK – 13, SS (10); - Пробіть по пінам UART), D0; * 1 – RX (приймач даних UART), D1. RX і TX можуть використовуватися для зв'язку по послідовному інтерфейсу; - * 3, 29 – скидання; * 4, 29 – земля; * 5 – D2, переривання INT0; * 6 – D3, переривання INT1 / ШИМ / AINO; * 7 – D4, переривання INT2; * 8 – D5, переривання INT3; * 9 – D6, переривання INT4; * 10 – D7, переривання INT5; * 11 – D8, переривання INT6; * 12 – D9, переривання INT7; * 13 – D10, переривання INT8; * 14 – D11, переривання INT9; * 15 – D12, переривання INT10; * 16 – D13, переривання INT11; * 17 – D4, переривання INT2; * 18 – D5, переривання INT3; * 19 – D6, переривання INT4; * 20 – D7, переривання INT5; * 21 – D8, переривання INT6; * 22 – D9, переривання INT7; * 23 – D10, переривання INT8; * 24 – D11, переривання INT9; * 25 – D12, переривання INT10; * 26 – D13, переривання INT11; - * I2C SDA / AIN1. AINO і AIN1 – входи для швидкодіючого аналогового компаратора; * 8 – A5, лічильник T1 / шина I2C; - * D6–D13, з яких D6 (9й), D9 (12й), D10 (13й) і D11 (14й) використовуються як виходи ШИМ. D13 (16й пін) – світлодіод; * D12 – MISO, D13 – SCK використовуються для зв'язку по інтерфейсу SPI; * 18 – AREF, це опорна напруга для АЦП і аналогові входи A0 ... A7. Розрядність АЦП 10 біт. A4 (SDA), A5 (SCL) – використовуються для зв'язку по інтерфейсу I2C; - * D12 – MISO, D13 – SCK використовуються для зв'язку по інтерфейсу SPI; * 18 – AREF, це опорна напруга для АЦП і аналогові входи A0 ... A7. Розрядність АЦП 10 біт. A4 (SDA), A5 (SCL) – використовуються для зв'язку по інтерфейсу I2C; використовується спеціальна бібліотека Wire; Мікроконтролери мають великі функціональні можливості, але у обмежені, по сраувенію з Arduino Mega, число висновків. Тому на етапі складання схеми пристрою слід продумати максимально спростити проект, щоб скоротити число потрібних для підключення контактів. Підключення Arduino Nano до комп'ютера не становить особливих труднощів – воно аналогічно звичайній платі Uno. Єдина проблема при роботі з платою на базі чіпа ATMEGA 168 – в настройках потрібно вибрати спершу плату Nano, а потім поставити драйвер для CH340. Мікросхема CH340 часто використовується в платах Arduino з вбудованим USB-

Вона дозволяє зменшити витрати на виробництво плат, не впливаючи на її працездатність. За допомогою цього прошивати плати Arduino. Для того, щоб почати працювати з цією мікросхемою, потрібно встановити драйвер виконується в кілька етапів: * завантаження архіву з драйвером для потрібної операційної системи. Для Windows, і драйвери можна за посиланням в нашій статті про USB UART; * розпакування архіву; * пошук файлу SETUP.EXE, з'явиться вікно, в якому потрібно натиснути кнопку Install. Установка драйвера почнеться, після чого можна починати роботу.

Inter-Integrated Circuit) – послідовна асиметрична шина для зв'язку між інтегральними схемами всередині. Використовує дві двонаправлені лінії зв'язку (SDA і SCL), застосовується для з'єднання низько швидкісних периферійних пристроїв з процесорами і мікроконтроллерами. Даний модуль дозволяє зменшити кількість використовуваних висновків кобінетного з'єднання, потрібно тільки 2 виведення (SDA і SCL). Рисунок 2.5 – модуль I2C Te: Підтримка дисплеїв: iLCD і 16x02/20x04; * Додатково: ірегулювання контрастності; * Напруга харчування 5В; * 154мм x 19мм x 15мм. Загальні відомості інтерфейсного модуля I2C: Оскільки кількість контактів на контролерах Arduino використанні різних датчиків і модулів вони закінчуються, з'являється необхідність в їх економії, для цих випадків його допомогою можна реалізувати передачу по двом контактам (SDA і SCL). Рисунок 2.6 – Група контактів в I2C модулі, побудований він на мікросхемі PCF8574T. Резистори R8 (4.7кОм) і R9 (4.7кОм) необхідні для підтяжки ліній підключення двох і більше пристроїв по шині I2C необхідно використовувати підтяжку тільки на одному пристрої перемикач (за схемою видно що лінії A0, A1, A2 підтягнуті до харчування через резистори R4, R5, R6), необхідні пристрої, всього їх 8 варіантів. Зміна адресації дає нам можливість підключення до восьми пристроїв по шині I2C варіанти адрес показані на малюнку (за замовчуванням адреса пристрою 0x27). Так само модуль оснащений допомогою можна змінити контрастність LCD дисплея. Рисунок 2.7 – Лінії A0, A1, A2 Для з'єднання на модулі розташовані наступні групи: – SCL: лінія тактирування (Serial Clock); – SDA: лінія даних (Serial Data); – VCC: «+» харчування; Друга група: – VSS: «-» харчування; – VDD: «+» харчування; – VO: Висновок управління контрастності; – RW: Читання/запис (режим запису при з'єднанні з землею); – E: Enable (стрибає по спаду); – DB0–DB3: Старші біти інтерфейсу; – A: «+» харчування підсвічування; – K: «-» харчування. Третя група: (за замовчуванням встановлена перемикач) – VCC: – A від LCD: Рисунок 2.8 – Інтерфейсний модуль I2C має кілька переваг: * для роботи потрібно всього 2 лінії – SDA (лінія даних) і SCL (лінія синхронізації); * підключення пристроїв; * зменшення часу розробки; * для управління всім набором пристроїв потрібно тільки один мікроконтролер підключаються мікросхем до однієї шини обмежується тільки граничною ємністю; * високий ступінь збереження фільтра переважної сплески, вбудованого в схеми; * проста процедура діагностики виникають збоїв, швидко налаштувати шина вже інтегрована в саму Arduino, тому не потрібно розробляти додатково шинний інтерфейс; Недоліки: * існує ліній – 400 пФ; * важке програмування контролера I2C, якщо на шині є кілька різних пристроїв; * при великій кількості труднощі локалізації збою, якщо одне з них помилково встановлює стан низького рівня; Рідкокристалічний дисплей LCD 2004 є хорошим вибором для виведення рядків символів в різних проектах. Він коштує недорого, є різні модифікації підсвічування, ви можете легко завантажити готові бібліотеки для скетчів Arduino. Короткий опис пинів LCD Давай розглянемо LCD 2004 уважніше: Рисунок 2.9 – Короткий опис пинів LCD 2004 Кожен з виводів має своє призначення: 1. Земля; 2. Установка контрастності монітора; 4. Команда, дані; 5. Записування та читання даних; 6. Enable; 7–14. Лінії даних; 15. Мінус підсвічування. Технічні характеристики дисплея: * символний тип відображення, є можливість завантаження підсвітка; * контролер HD44780; * напруга живлення 5В; * формат 20x4 символів; * діапазон робочих температур в діапазоні температур зберігання від –30С до +80 С; * кут огляду 180 градусів; Схема підключення LCD до плати Arduino приєднання монітора безпосередньо до мікроконтролера Arduino без I2C виглядає наступним чином. Рисунок 2.10 – Підключення монітора Arduino без I2C Через велику кількість підключаються контактів може не вистачити місця для підключення елементів. Використання I2C зменшує кількість проводів до 4, а зайнятих пинів до 2. Підключення до Arduino. Для Arduino UNO R3 x 1 шт; – LCD-дисплей 1602A (2 x 16, 5V, Синій) x 1 шт; – інтерфейсний модуль I2C, IIC, TWI для LCD x 1 шт; – кабель USB 2.0 AB x 1 шт; – кабель FM (Female – Male) x 1 шт; – кабель USB 2.0 AB x 1 шт; Підключення : Насамперед, припаюємо модуль I2C до плати Arduino необхідно підключити дисплей до Arduino UNO. Для цього скористаємося проводками DuPont, підключення здійснюється за допомогою Другу частину розробки буде зібрано на макетній платі. Макетна плата – універсальна друкована плата для прототипів електронних пристроїв. Макетні плати поділяються на два типи: для монтажу за допомогою пайки та безпайкового макетних плат: Універсальні мають виключно металізовані отвори, які розробник повинен з'єднувати перемикачами намічені можливі місця під мікросхеми, по всій платі проведені шини живлення. Спеціалізовані – для пристроїв певної моделі. На таких платах є як розведені стандартні ланцюги, так і матриця отворів і доріжок для нестандартних мікроконтролерів стандартними ланцюгами будуть: * посадкове місце для мікросхеми; * живлення; * «земля»; * катоди внутрішньосхемного програмування. Для диплому було обрано порожню двосторонню макетну палату 5x7см. Тимчасово Рисунок 2.11 – Макетна плата PCB-5x7 Для того щоб знати температуру всередині корпусу використовується датчик

DS18B20 – це датчик температури, який має роздільну здатність перетворення від 9 до 12 розрядів. Тривожний сигнал якісно контролювати температуру рідини. Більшість параметрів контролю задаються самостійно користувачем. Вони можуть бути перенастроєні у майбутньому. Датчик ds18b20 використовує протокол 1-Wire інтерфейсу для обміну даними.

Датчик ds18b20 Датчик може вимірювати температуру в широкому діапазоні, від -55 до +125 градусів за Цельсом. Найчастіше становить максимум півградуса. Вищеописані характеристики роблять датчики популярним для використання в контролі, моніторингу температурних змін у будинках, а також у вузлах обладнання. Короткий огляд характеристик датчика.

Власнику потрібен невеликий резистор, що підтягує, адже всі агрегати підключаються до загальної шини. Утворюється паразитним харчуванням. Яким варіантом харчування користуватись вирішувати саме власнику.

Для зручності позначення температури та для декору використовуватиметься RGB світлодіодна стрічка RGB (Red, Green, Blue – червоний, зелений, синій) – це світлодіодна стрічка, яка змінює колір світіння під напругою. Змінюючи яскравість світіння кожного кристала, можна отримати будь-який колір видимого спектра.[12] Для підключення RGB стрічки та датчика температури використовуватиметься MOSFET транзистор IRF3205PBF. Рисунок 2.15 – MOSFET транзистор IRF3205PBF

англійських словосполучень: Metal-Oxide-Semiconductor (метал – оксид – напівпровідник) та Field-Effect Transistor (полівисхідний електричний транзистор). Тому MOSFET – це нічим іншим, як звичайний МОП-транзистор. Суть роботи полягає в можливості управління струмом, що протікає через нього, за допомогою електричного поля (напруги). Цим відрізняє MOSFET транзисторів біполярного типу, де керування великим вихідним струмом здійснюється за допомогою малого вхідного струму.

Для підключення використовуються резистор 10 кОм та 100 Ом. Рисунок 2.17 – пасивний елемент електричного кола. Також його називають "опір", завдяки здатності обмежувати струм, створюючи на собі падіння напруги.

Резистори використовуються практично у всіх електричних схемах. Найчастіше їх використовують для поділу напруги та керування силою струму.[9] Для переміщення між етапами на дисплеї використовуються кнопки B3F-1070 Характеристики: група OFF-(ON), максимальний струм (Imax), mA 50, тип монтажу THT, тип контактної групи (США) SPST. Рисунок 2.18 – кнопка B3F-1070

Для підключення вентиляторів використовуються гвинтовий клемник DG306-5.0-02P-12-00AH Рисунок 2.19 – клемник DG306-5.0-02P-12-00AH

Для живлення цієї системи вмонтовується кабель Molex 12B Рисунок 2.20 – Molex кабель 2.2

Вибір середовища програмування Не менш важливим питанням що до вирішення є використання програмних засобів для проектування відповідного набору команд та прошивання платформи. Тому розглянемо основні програмні компоненти нижче.

Стандартна середовище розробки Arduino IDE використовується для роботи всіх видів Arduino з комп'ютером. Щоб встановити Arduino IDE, спочатку завантажити Arduino IDE з офіційного сайту і встановити її. Зручніше завантажувати Windows Installer, який розробки буде встановлена на постійному робочому комп'ютері. Якщо викачаний архів, то його потрібно розпакувати в Arduino.exe. Як тільки середовище встановлено, потрібно її запустити. Для цього потрібно підключити до комп'ютера плату через USB. Потім перейти в меню Пуск >> Панель управління >> Диспетчер пристроїв, знайти там Порти COM і встановити плату і вказано номер порту, до якого підключається плата. Після цього потрібно запустити Arduino IDE Інструменти >> Порт, і вказати порт, до якого приєднана Arduino. У мене Інструменти >> Плати потрібно вибрати мій даному випадку Arduino Nano. Якщо у вас плата Nano версії 2.0, то вам потрібно також вибрати варіант процесора. Важливо пам'ятати, що якщо до комп'ютера буде підключатися інша плата, настройки знову потрібно буде поміняти.

2.2.1 Середовище розробки Arduino – це електронний конструктор і зручна платформа швидкої розробки електронних пристроїв професіоналів. Платформа користується величезною популярністю в усьому світі завдяки зручності і простоті модифікації відкритої архітектури і програмного коду. Пристрій програмується через USB без використання програматорів. Arduino вийти за рамки віртуального світу в фізичний і взаємодіяти з ним. Пристрої на базі Arduino можуть отримувати інформацію середовища за допомогою різних датчиків, а також можуть управляти різними виконавчими пристроями. Arduino програмується за допомогою мови Arduino (заснований на мові Wiring) і середовища розробки Arduino (Processing). Проекти пристроїв, засновані на Arduino, можуть працювати самостійно, або ж взаємодіяти з комп'ютером (наприклад, Flash, Processing, MaxMSP). Плати можуть бути зібрані користувачем самостійно або купити розробки Arduino складається з вбудованого текстового редактора програмного коду, області повідомлень, вікна налаштувань і панелі інструментів з кнопками часто використовуваних команд і декількох меню. Для завантаження програм і зв'язу підключається до апаратної частини Arduino. Програма, написана в середовищі Arduino, називається скетч. Скетч

редакторі, що має інструменти вирізки / вставки, пошуку заміни тексту. Під час збереження і експорту проєкту з'являються пояснення, також можуть відображатися виникли помилки. Рисунок 2.21 – Інтерфейс Arduino IDE В основні функціональні елементи: 1) меню програми; 2) панель швидкого доступу до найбільш важливих функцій; 3) коду програми); 4) панель повідомлень і статусу програми. Меню програми дозволяє здійснювати управління проєктом нового проєкту, збереження поточного, роздрукувати на принтері вихідний код. Рисунок 2.22 – Меню «Скетч» Цікавим елементом меню «Інструменти», яке включає до себе функції автоматичного форматування коду, архівування проєкту послідовного порту (USB в Arduino розглядається як звичайний послідовний порт). Рисунок 2.23 – Меню «Інструменти» є можливість вибору відповідної плати, тобто вашої системи Arduino підключення знаходяться всі офіційні версії Arduino. У меню «Інструменти» ви також можете встановити порт, до якого підключено Arduino IDE сам визначає порт, але іноді потрібно вручну встановити номер порту в налаштуваннях. Для нормального використання використовується панель швидкого доступу, яка оснащена найбільш важливими кнопками. Це рішення, що полегшує нам прямий доступ до практично всіх необхідних параметрів при написанні і тестуванні програми. Рисунок 2.24 – Вони дозволяють (зліва направо): * компілювати програму; * завантажити програму в мікроконтролер (перед компілюється); * почати роботу над новим проєктом; * відкрити існуючий проєкт; * зберегти проєкт на диск; * вкрити порт. Всі опції, розташовані на панелі швидкого доступу, продубльовані в меню програми. Останнім елементом меню і статусу. Видима там інформація дозволяє користувачеві знайти помилки в програмному коді і отримати підтвердження компіляції і завантаження програми в мікроконтролер. Arduino IDE в порівнянні з її аналогами такими як WinAVR, AVR для новачків, які тільки починають розбиратися в програмуванні та схемотехніці. В самій IDE є дуже багато різних вершин AVR LibC також має розумний інтерфейс і значно спрощує створення/завантаження. Платформа Arduino що ви хочете зробити зі своєю схемою, а не займатися технічними тонкощами використання самого мікропроцесора можна сказати, що Arduino IDE – це простий програмний пакет, який дозволяє запрограмувати будь-яку відому плату послідовним портом і легко управляти проєктами.[6] 2.2.2 Вибір мови програмування На сьогоднішній час для мікроконтролера AVR є Assembler та GCC C/C++. Однак при використанні IDE доречно використовувати мову C та GCC C, що інтегровано до відповідного середовища. Мова програмування пристроїв Arduino заснований на C бібліотекою AVR Libc і дозволяє використовувати будь-які її функції. Лістинг 2.1 – Необхідні блоки скетча void setup() на початку програми ініціалізації*/ /* Ця програма виконується 1 раз і призначена для приведення системи у вихідний стан записувати функції ініціалізації портів і периферійних пристроїв*/ /* закриваюча дужка в кінці програми ініціалізації відкриваюча дужка на початку основного циклу*/ /* по суті нескінченний цикл, це основний цикл програми, все це основна частина(крім початкової ініціалізації)*/ /* закриваюча дужка в кінці основного циклу*/ Функція – структура має ім'я і містить в собі деяку послідовність дій. Більшість записів в програмі повинні закінчуватися ";", виключення – цикли, вибору, умов, а також опис прототипів функцій. На початку програми, перед функцією setup(), зазвичай визначення. Після включення живлення плати першою виконується функція setup(). Вона виконується тільки одразу після запуску плати в режимі роботи портів: порти, до яких підключені різні датчики, встановлюються як входи, а порти з виходом. Код, написаний в функції loop() починає виконуватися після виконання функції setup(), і виконується в постійно знову. У цій функції виконується основна робота: різні обчислення, отримання значень датчиків, виведення значень на екран Monitor – безкоштовна утиліта з відкритим вихідним кодом, яка надає централізований інтерфейс, де можна легко отримати інформацію про продуктивності обладнання, температурні датчики, споживання напруги, навантаження і тактові частоти процесора в більшості мікросхем, якими обладнуються сучасні популярні плати. Рисунок 2.25 Утиліта Open Hardware Monitor – повністю безкоштовний програмний пакет, призначений для того, щоб користувачі могли отримати інформацію про свої операційної системи в режимі реального часу. Мало того, що це може бути корисно для розробників, такий софт хоче збільшити операційний здатність своєї системи або отримати доступ до додаткових рішень щодо усунення неполадок. Open Hardware Monitor – відобразити всі найбільш важливі фізичні показники операційної системи. Наприклад, температура процесора, напруга, що споживаються всіма жорсткими дисками, Швидкість будь-яких вентиляторів і т. Д. Ці фактори можуть допомогти підвищити ефективність жорсткого диска і збільшити загальний продуктивність «в один клік» робить використання програми максимально простим. Утиліту також можна налаштувати на запуск в фоновому режимі. Переваги: * може запускатися під час запуску системи; * сумісність з більшістю контролерів плат, в тому числі Winbond, ITE і Fintek; * дає користувачеві можливість відображати контрольовані значення три головному вікні інтерфейсу, за допомогою персоналізованого настроюється гаджета для робочого столу і в системі відображення температури процесора процесорів Intel і AMD ; * моніторинг датчиків відеокарт ATI і Nvidia; * зворотний SMART (самоконтроль, аналіз і звітність) зв'язок практично для всіх комп'ютерних жорстких дисків; * виведення напруги процесора Vcore і батареї; * відображає температуру системи і процесора в градусах Цельсія і в Фаренгейтах; * обертає верхнього, нижнього і заднього вентиляторів, а також джерела живлення. Недоліки: 1. Середньому користувачеві

бути складно зрозуміти, що означають різні статистичні дані. Тому для ефективного аналізу результатів зві забезпеченням, може знадобитися допомога професіонала. 2. Оскільки це бета-версія, софт може іноді видає Висновок за розділом Для реалізації цього проекту було обрано: * Arduino NANO 1шт; * I2C модуль; * LCD дисплей mini; * датчик температури Ds18b20; * комп'ютерний кулер; * RGB стрічки; * MOSFET транзисторів; * резисторів роз'єм. Для створення прошивки програмного коду була обрана програма Arduino IDE 1.8.0 та утиліта Open Hardware Monitor. РЕАЛІЗАЦІЯ СИСТЕМИ МОНІТОРІНГУ 3.1 Проектування апаратного комплексу проекту Збірку Монітору стану за що треба припаяти ніжки до I2C модуля, а потім цей модуль припаяти до LCD дисплею (якщо він не припадає підключити Arduino в до I2C проводами Рисунок 3.1 Підключення Arduino к I2C Рисунок 3.2 Ініціалізація схе прошивки треба підключити Arduino до комп'ютера кабелем USB micro. Рисунок 3.3 Подання живлення до Arduino модифікувати проект. Було додано RGB стрічку, кулера для комп'ютера та датчик температури. Перший MOSFET керує ШІМ сигналу. Три інших MOSFET управляють кольорами RGB стрічки. Кожен MOSFET відповідає за окремий колір. Складання компонентів було проведено на друкованій платі. Рисунок 3.4 – Повна схема приладу Рисунок 3.5 – Після підключення Arduino до кабелю живлення ми підключаємо кабель молекс до блоку живлення, для підключення буде виводити інформацію на LCD екрані з утиліти open hardware monitor. Рисунок 3.6 – Інтерфейс monitor. На першому екрані виводиться температура та завантаженість CPU, GPU, GPUmem, RAMuse. На другому швидкість вентилятора, температура датчика, температура материнської плати, жорсткого диска та час з моменту потрібно зайти в утиліту open hardware monitor і вибрати порт. Рисунок 3.7 – Вибір порту За замовчуванням залежить від температури. Джерелом температури можна вибрати CPU, GPU, максимум між процесором та відеокартою. Рисунок 3.8 – Вибір джерела температури Регулювання температури та швидкості відбувається за лінійним законом. Функція температури та швидкості Також є функція ручного управління вентилятора і самостійно регулювати швидкість обертання – Ручне управління кулером Колір світлодіодної стрічки за промовчанням від вибраного джерела температури. стрічка світиться синім кольором, потім при підвищенні температури вона світиться зеленим, жовтим і в самому кінці червоним. Також ручне керування світлом можна поставити і на світлодіодну стрічку і вибрати будь-який колір. Рисунок 3.11 Також нижче можна керувати яскравістю світлодіодної стрічки. Рисунок 3.12 – Керування яскравістю Ще нижче буде оновлення графіків від 1 до 10 секунд. Рисунок 3.13 – Налаштування інтервалу оновлення графіків 3.2 Тестування Для живлення та роботоздатності проекту потрібно підключити Arduino до комп'ютера та для додаткового живлення Для запуску треба увімкнути трансляцію даних в утиліті open hardware monitor. Рисунок 3.14 – Увімкнення підключення до живлення, компілювання Arduino та трансляція даних проект виглядає так. Рисунок 3.15 Висновок за розділом Використовуючи обрані у другому розділі компоненти, був створений Монітору стану підключаємого функціоналу створено програмний код прошивки плати. Опис диплому: * Виведення основних параметрів на дисплей LCD; * Температура: CPU, GPU, материнська плата, найгарячіший HDD; * Рівень завантаження: CPU, GPU; * Графіки зміни вищезазначених параметрів за часом; * Температура зовнішніх датчиків (DS18B20); * Поточний історичний графік швидкості вентиляторів; * Управління великою кількістю 12 вольтових 2, 3, 4 провідних вентиляторів; * Автоматичне керування температурою; * Ручне керування швидкістю з інтерфейсу програми; * Управління RGB світлодіодною стрічкою; * Пропорційно температурі (синій – зелений – жовтий – червоний) ; * Ручне керування кольором і яскравістю світлодіодної стрічки. ВИСНОВКИ Здійснивши огляд предметної області створення системи моніторингу апаратної частини ПК створення цього монітору є гарним рішенням для модифікування свого ПК, для спостереження за станом температури своїх навичок в програмуванні. Ця інформація важлива для того, щоб дізнатися при перегріві комплектуючих. Для цього було обрано: * Arduino NANO 1шт; * I2C модуль; * LCD дисплей 2004; * Дата кабель USB mini; * датчик температури DS18B20; * кулер; * RGB стрічки; * MOSFET транзисторів; * резисторів; * макетна плата; * молекс роз'єм. Для створення прошивки коду була обрана програма Arduino IDE 1.8.0 та утиліта Open Hardware Monitor. Використовуючи обрані у другому розділі компоненти, був створений Монітору стану заліза ПК. За допомогою підключаємого функціоналу створено програмний код прошивки плати. Підключення Arduino NANO. [Електронний ресурс] / Режим доступу www. URL: <https://arduinomaster.ru/platy-arduino> 23.05.2022 2. Огляд інтерфейсного модуля, I2C. [Електронний ресурс] / Режим доступу www. URL: <http://interfeysnogo-modulya-i2c/> 23.05.2022 3. Підключення дисплея LCD 1602 до arduino по i2c. [Електронний ресурс] / Режим доступу www. URL: <https://arduinomaster.ru/datchiki-arduino/lcd-i2c-arduino-display-ekran/> 25.05.2022 4. Підключення дисплея LCD 1602 до arduino по i2c. [Електронний ресурс] / Режим доступу www. URL: <http://arduino.ru/Hardware/ArduinoBoardNano> – 25.05.2022 5. Підключення дисплея LCD 1602 до arduino по i2c. [Електронний ресурс] / Режим доступу www. URL: <https://robo02.ru/2018/03/22/arduino-display-connect-a-6/> 6. Офіційний Arduino [Електронний ресурс] / Режим доступу www. URL: <https://www.arduino.cc/> – 26.05.2022 7. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 8. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 9. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 10. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 11. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 12. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 13. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 14. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 15. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 16. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 17. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 18. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 19. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 20. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 21. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 22. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 23. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 24. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 25. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 26. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 27. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 28. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 29. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 30. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 31. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 32. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 33. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 34. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 35. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 36. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 37. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 38. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 39. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 40. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 41. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 42. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 43. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 44. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 45. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 46. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 47. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 48. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 49. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 50. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 51. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 52. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 53. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 54. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 55. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 56. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 57. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 58. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 59. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 60. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 61. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 62. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 63. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 64. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 65. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 66. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 67. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 68. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 69. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 70. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 71. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 72. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 73. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 74. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 75. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 76. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 77. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 78. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 79. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 80. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 81. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 82. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 83. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 84. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 85. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 86. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 87. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 88. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 89. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 90. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 91. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 92. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 93. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 94. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 95. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 96. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 97. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 98. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 99. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022 100. Arduino IDE 1.8.0 [Електронний ресурс] / Режим доступу <https://www.arduino.cc/en/software> – 26.05.2022


```

timeoutTick(); // проверка таймаута if (tm.Hour < 7) { lcd.noDisplay(); lcd.noBacklight(); } else { lcd.display
----- ОСНОВНОЙ ЦИКЛ -----
(irrecv.decode(&results)) { switch (results.value) { case 16736925: display_mode++; reDraw_flag = 1
display_mode = 0; break; case 16754775: display_mode--; reDraw_flag = 1; if (display_mode < 0) disp
irrecv.resume(); } } void buttonsTick() { btn1_sig = !digitalRead(BTN1); btn2_sig = !digitalRead(BTN2); if (
reDraw_flag = 1; if (display_mode == 0) display_mode = 1; else if (display_mode == 1) display_mode = 0
btn1_flag = 1; } if (!btn1_sig && btn1_flag) { btn1_flag = 0; } if (btn2_sig && !btn2_flag) { reDraw_flag = 1
display_mode = 3; else if (display_mode == 3) display_mode = 2; else display_mode = 2; btn2_flag = 1; } if
{ btn2_flag = 0; } } void getTemperature() { if (updateTemp_flag) { sensors.requestTempC(Thermometer1);
sensors.getTempC(Thermometer1); temp1 = sensors.getTempC(Thermometer1); updateTemp_flag = 0; }
PCdata[16]; if (PCdata[13] == 1) // если стоит галочка Manual Color LEDcolor = PCdata[15]; // цвет равен устан
// если нет LEDcolor = map(mainTemp, PCdata[11], PCdata[10], 0, 1000); LEDcolor = constrain
(COLOR_ALGORITHM) { // алгоритм цвета 1 // синий убавляется, зелёный прибавляется // зелёный убавляется
(LEDcolor <= 500) { k = map(LEDcolor, 0, 500, 0, 255); R = 0; G = k; B = 255 - k; } if (LEDcolor > 500) { k = m
255); R = k; G = 255 - k; B = 0; } } else { // алгоритм цвета 2 // синий максимум, плавно прибавляется зелёный //
убавляется синий // зелёный максимум, плавно прибавляется красный // красный максимум, плавно убавляетс
250) { k = map(LEDcolor, 0, 250, 0, 255); R = 0; G = k; B = 255; } if (LEDcolor > 250 && LEDcolor <= 500) { k = r
255); R = 0; G = 255; B = 255 - k; } if (LEDcolor > 500 && LEDcolor <= 750) { k = map(LEDcolor, 500, 750, 0, 255
(LEDcolor > 750 && LEDcolor <= 1000) { k = map(LEDcolor, 750, 1000, 0, 255); R = 255; G = 255 - k; B = 0; } } Rf =
100); Bf = (b * B / 100); analogWrite(R_PIN, Rf); analogWrite(G_PIN, Gf); analogWrite(B_PIN, Bf); } void dutyC
== 1) // если стоит галочка ManualFAN duty = PCdata[14]; // скважность равна установленной ползунком
(PCdata[18]) { case 0: mainTemp = PCdata[0]; // взять опорную температуру как CPU break; case 1: mainTemp =
опорную температуру как GPU break; case 2: mainTemp = max(PCdata[0], PCdata[1]); // взять опорную темпе
GPU break; case 3: mainTemp = temp1; break; case 4: mainTemp = temp2; break; } duty = map(mainTemp,
PCdata[9], PCdata[8]); duty = constrain(duty, PCdata[9], PCdata[8]); } if (!timeOut_flag) duty = ERRO
соединение, поставить вентиляторы на ERROR_DUTY } void parsing() { while (Serial.available() > 0) { char
(aChar != 'E') { inData[index] = aChar; index++; inData[index] = '\0'; } else { char *p = inData; char *str; in
while ((str = strtok_r(p, ";", &p)) != NULL) { string_convert = str; PCdata[index] = string_convert.toInt();
updateDisplay_flag = 1; updateTemp_flag = 1; tm.Second = PCdata[21]; tm.Hour = PCdata[19]; tm.Minute =
PCdata[22]; tm.Month = PCdata[23]; tm.Year = PCdata[24] - 1970; setTime(makeTime(tm)); } timeout = mi
void updateDisplay() { if (updateDisplay_flag) { if (reDraw_flag) { lcd.clear(); switch (display_mode) {
break; case 1: draw_labels_12(); break; case 2: draw_labels_21(); break; case 3: draw_labels_22(); break; }
(display_mode) { case 0: draw_stats_11(); break; case 1: draw_stats_12(); break; case 2: draw_stats_
draw_stats_22(); break; case 50: debug(); break; } updateDisplay_flag = 0; } } void draw_stats_11() {
lcd.print(PCdata[0]); lcd.write(223); lcd.setCursor(13, 0); lcd.print(PCdata[4]); if (PCdata[4] < 10) perc = "%
perc = "%"; else perc = ""; lcd.print(perc); lcd.setCursor(4, 1); lcd.print(PCdata[1]); lcd.write(223);
lcd.print(PCdata[5]); if (PCdata[5] < 10) perc = "% "; else if (PCdata[5] < 100) perc = "%"; else perc = ""; lcd.pri
i++) { byte line = ceil(PCdata[lines[i]] / 16); lcd.setCursor(7, i); if (line == 0) lcd.printByte(1) else lcd.printBY
n++) { if (n < line) lcd.printByte(4); if (n >= line) lcd.printByte(2); } if (line == 6) lcd.printByte(4) else
draw_stats_12() { lcd.setCursor(13, 0); lcd.print(PCdata[7]); if (PCdata[7] < 10) perc = "% "; else if (PCdata
perc = ""; lcd.print(perc); lcd.setCursor(13, 1); lcd.print(PCdata[6]); if (PCdata[6] < 10) perc = "% "; else if (PC
else perc = ""; lcd.print(perc); for (int i = 0; i < 2; i++) { byte line = ceil(PCdata[lines[i + 2]] / 16); lcd.set
lcd.printByte(1) else lcd.printByte(4); for (int n = 1; n < 5; n++) { if (n < line) lcd.printByte(4); if (n >= line) lcd.p
lcd.printByte(4) else lcd.printByte(3); } } void draw_stats_21() { lcd.setCursor(13, 0); lcd.print(duty); if ((duty
((duty) < 100) perc = "%"; else perc = ""; lcd.print(perc); lcd.setCursor(3, 1); lcd.print(temp1); lcd.write(
lcd.print(temp2); lcd.write(223); byte line = ceil(duty / 16); lcd.setCursor(6, 0); if (line == 0) lcd.printByte(1) els
n = 1; n < 5; n++) { if (n < line) lcd.printByte(4); if (n >= line) lcd.printByte(2); } if (line == 6) lcd.printByte(4) e
draw_stats_22() { lcd.setCursor(2, 0); lcd.print(PCdata[2]); lcd.write(223); lcd.setCursor(10, 0); lcd.print(P
char lcd_time[10]; sprintf(lcd_time, sizeof(lcd_time), "%02d:%02d", PCdata[19], PCdata[20]); char lcd_dc
sizeof(lcd_date), "%02d/%02d/%04d", PCdata[22], PCdata[23], PCdata[24]); lcd.setCursor(0, 1); lcd.print(lcd_
lcd.print(lcd_date); } void draw_labels_11() { lcd.createChar(0, degree); lcd.createChar(1, left_empt
center_empty); lcd.createChar(3, right_empty); lcd.createChar(4, row8); lcd.setCursor(0, 0); lcd.print("C

```

```
lcd.print("GPU:"); } void draw_labels_12() { lcd.createChar(0, degree); lcd.createChar(1, left_center_empty); lcd.createChar(3, right_empty); lcd.createChar(4, row8); lcd.setCursor(0, 0); lcd.print("GP 1); lcd.print("RAMuse:"); } void draw_labels_21() { lcd.createChar(0, degree); lcd.createChar(1, left_center_empty); lcd.createChar(3, right_empty); lcd.createChar(4, row8); lcd.setCursor(0, 0); lcd.print("FA lcd.print("T1: "); lcd.setCursor(8, 1); lcd.print("T2:"); } void draw_labels_22() { lcd.createChar(0, degree); lcd.c lcd.createChar(2, center_empty); lcd.createChar(3, right_empty); lcd.createChar(4, row8); lcd.setCu lcd.setCursor(6, 0); lcd.print("HDD:"); } void timeoutTick() { if (millis() - timeout > 5000) { if (timeOut. lcd.setCursor(2, 0); lcd.print("DISCONNECTED"); timeOut_flag = 0; reDraw_flag = 1; } char lcd_time_i sizeof(lcd_time_i), "%02d:%02d",hour(),minute()); char lcd_date_i[12]; sprintf(lcd_date_ "%02d/%02d/%04d",day(),month(),year()); lcd.setCursor(0, 1); lcd.print(lcd_time_i); lcd.setCursor(6, 1); lcd.p show_logo() { lcd.createChar(0, logo0); lcd.createChar(1, logo1); lcd.createChar(2, logo2); lcd.createChar(3, logo4); lcd.createChar(5, logo5); lcd.setCursor(0, 0); lcd.printByte(0); lcd.printByte(1); lcd.printByte lcd.printByte(3); lcd.printByte(4); lcd.printByte(5); lcd.setCursor(9, 0); lcd.print("PC"); lcd.setCursor(5, 1); lcd.p debug() { lcd.clear(); lcd.setCursor(0, 0); for (int j = 0; j < 5; j++) { lcd.print(PCdata[j]); lcd.print(" "); } lcd.setC 10; j++) { lcd.print(PCdata[j]); lcd.print(" "); } lcd.setCursor(0, 2); for (int j = 10; j < 15; j++) { lcd.print(P lcd.setCursor(0, 3); for (int j = 15; j < 18; j++) { lcd.print(PCdata[j]); lcd.print(" "); } }
```

Source

Запорізький інститут економіки та інформаційних ...Підрозділи - Запорізь...

0.27%

Запорізький інститут економіки та інформаційних ...Підрозділи - Запорізький інститут економіки та ...

<https://www.zieit.edu.ua/>

процесори. Нерідко зміна ядра в одному і тому ж сімействі процесорів сп...

0.27%

процесори. Нерідко зміна ядра в одному і тому ж сімействі процесорів спричиняє за собою заміну процесорного роз'єму (сокет, англ. socket), з чого витікають ...

http://xemttc.at.ua/Metod_zbs/akredit_2020/KM/metod_lr_arkh.pk.pdf

Центральний процесор - Вікіпедія - NiNa.Az

0.27%

Центральний процесор - Вікіпедія - NiNa.Az

<https://www.wiki.uk-ua.nina.az/%D0%A6%D0%9F.html>

Графічний процесор (англ. graphics processing unit, GPU) — окремий п...

0.27%

Графічний процесор (англ. graphics processing unit, GPU) — окремий пристрій персонального комп'ютера або ігрової приставки, виконує графічний рендеринг. Сучасні графічні процесори дуже ефективно ...

https://uk.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84%D1%96%D1%87%D0%B%D0%B8%D0%B9_%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D0%BE%D1%80/

Оперативна пам'ять (RAM) – Google SitesВиди внутрішньої пам'яті – Внут...

0.27%

Оперативна пам'ять (RAM) – Google SitesВиди внутрішньої пам'яті – Внутрішня пам'ять '1
– Google Sites

<https://sites.google.com/site/2randomaccessmemory8>

by ЮО Худякова · 2020 — Найбільше значення впадає або впливає стру...

0.55%

by ЮО Худякова · 2020 — Найбільше значення впадає або впливає струму не повинно перевищувати значення 40. мА, а загальний струм контактів повинен бути не більше 200 мА.

https://essuir.sumdu.edu.ua/bitstream/123456789/78906/1/khudiakova_bac_rob.pdf

by МБ Гуменицький · 2019 — 18 – AREF, це опорна напруга для АЦП мікрок...

0.82%

by МБ Гуменицький · 2019 — 18 – AREF, це опорна напруга для АЦП мікроконтролера. • 19 – 26: аналогові входи А0 ... А7. Розрядність АЦП 10 біт. А4. (SDA), А5 (SCL) – використовуються ...

http://lib.pnu.edu.ua:8080/bitstream/123456789/10776/1/%D0%9C%D0%B0%D0%B3%D1%96%D1%81%D1%82%D0%B5%D1%80%D1%81%D1%8C%D0%BA%D0%B0_%D0%93%D1%83%D0%BC%D0%B5%D0%BD%D0%B8%D1%86%D1%8C%D0%BA%D0%B8%D0%B9.pdf

Програмування ки індикаторів i2c для Ардуіно ... – totrdlo.ru

0.27%

Програмування ки індикаторів i2c для Ардуіно ... – totrdlo.ru

<https://totrdlo.ru/uk/programmirovanie-zhki-indikatorov-i2c-dlya-arduino-podklyuchenie-lcd-ekrana.html>

by ОА Погребенний · 2019 — Світлодіодна підсвітка;. ◦ Символьний тип ві...

0.27%

by ОА Погребенний · 2019 — Світлодіодна підсвітка;. ◦ Символьний тип відображення, є можливість завантаження символів;. ◦ Формат 16x2 символів;. ◦ Контролер HD44780;.

https://ela.kpi.ua/bitstream/123456789/30682/1/Pogrebennyi_magistr.pdf

Підключення дисплея 1 602 по i2c до arduino. LCD I2C ...

0.27%

Підключення дисплея 1 602 по i2c до arduino. LCD I2C ...

<https://hddrecover.ru/different/podklyuchenie-displeya-1602-po-i2c-k-arduino-lcd-i2c-modul-podklyuchenie-k-arduino/>

Лабораторні Роботи Основи Робототехніки | PDF – Scribd

0.27%

Лабораторні Роботи Основи Робототехніки | PDF – Scribd

<https://www.scribd.com/document/394822796/%D0%9B%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D1%96-%D0%A0%D0%BE%D0%B1%D0%BE%D1%82%D0%B8-%D0%9E%D1%81%D0%BD%D0%BE%D0%B2%D0%B8-%D0%A0%D0%BE%D0%B1%D0%BE%D1%82%D0%BE%D1%82%D0%B5%D1%85%D0%BD%D1%96%D0%BA%D0%B8>

Бібліотека AVR для роботи з шиною I2C та з годинником ...

0.27%

Бібліотека AVR для роботи з шиною I2C та з годинником ...

<https://shongames.ru/uk/internet/biblioteka-avr-dlya-raboty-s-shinoy-i2c-i-s-chasami-realnogo-vremeni-pcf8583/>

Обзор модуль пламени для Arduino – RobotChipПодключение матрично...

0.27%

Обзор модуль пламени для Arduino – RobotChipПодключение матричной мембранной клавиатуры к Arduino

<https://robotchip.ru/obzor-modul-plameni-dlya-arduino/>

Підключення RGB стрічки та варіанти її ... – zachetpizza.ru

0.27%

Підключення RGB стрічки та варіанти її ... – zachetpizza.ru

<https://zachetpizza.ru/uk/analytics/podklyuchenie-rgb-lenty-i-varianty-e-ispolzovaniya-rgb-svetodiody.html>

by АВ Сагун · 2021 — Проекти пристроїв, засновані на Arduino, можуть пр...

0.55%

by АВ Сагун · 2021 — Проекти пристроїв, засновані на Arduino, можуть працювати самостійно, або ж взаємодіяти з програмним забезпеченням на комп'ютері (наприклад, такими ...

https://ela.kpi.ua/bitstream/123456789/46647/1/RPMODPM_posibnyk.pdf

by МЮ Тягунова · 2021 — Середовище розробки Arduino складається з вб...

0.27%

by МЮ Тягунова · 2021 — Середовище розробки Arduino складається з вбудованого текстового редактора програмного коду, області повідомлень, вікна виведення тексту (консолі), панелі ...

http://eir.zntu.edu.ua/bitstream/123456789/7569/1/MV_Tiahunova.pdf

Завантажити програму для програмування Ардуіно ...

0.27%

Завантажити програму для програмування Ардуіно ...

<https://newtravelers.ru/uk/poleznoe/skachat-programmu-dlya-programmirovaniya-arduino-podklyuchenie-platy.html>

by ВІ Рева · 2019 — Код, написаний в функції loop() починає виконуватися...

0.27%

by ВІ Рева · 2019 — Код, написаний в функції loop() починає виконуватися після виконання функції setup(), і виконується в нескінченному циклі знову і знову. У цій функції ...

<https://eir.zp.edu.ua/bitstream/123456789/4519/1/M07054.pdf>

Г. Шилдт С++: руководство для начинающих, 2-е издание.: Пер. с англ. — ...

0.27%

Г. Шилдт С++: руководство для начинающих, 2-е издание.: Пер. с англ. — М.: Издательский дом «Вильямс», 2005. — 672 с.: ил. — Парал. тит. англ. — 3 000.

http://eds.kpi.ua/wp-content/uploads/2017/12/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4%D0%B8%D1%87%D0%BA%D0%B0_%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F-2017.pdf

PCdisplay/PCdisplay_v_1.4_1602.ino at master - GitHub

0.55%

PCdisplay/PCdisplay_v_1.4_1602.ino at master - GitHub

https://github.com/AlexGyver/PCdisplay/blob/master/firmware/PCdisplay_v_1.4_1602/PCdisplay_v_1.4_1602.ino

PCdisplay_v_1_3_1602.ino · GitHub

0.27%

<https://gist.github.com/aklyk/4cd21b52dfaf0733ca592ae692024588>

PCdisplay_v_1.3_1602 · GitHubPCdisplay/PCdisplay_v1.6.ino at master ...

0.55%

PCdisplay_v_1.3_1602 · GitHubPCdisplay/PCdisplay_v1.6.ino at master - GitHub

<https://gist.github.com/1271/ae3838b4e9382713992bab8ab5af5775>

Apr 07, 2011 · What do the following loops print? Work out the answer by t...

0.82%

Apr 07, 2011 · What do the following loops print? Work out the answer by tracing the code, not by using the computer. a. `int s = 1; for (int n = 1; n <= 5; n++) { s = s + n; System.out.print(s + " "); }`

<https://www.chegg.com/homework-help/questions-and-answers/following-loops-print-work-answer-tracing-code-using-computer--int-s-1-int-n-1-n-q5250676/>

Question: Question 8 10 pts for (int j 10; j< 15; j++) cout<

0.27%

Question: Question 8 10 pts for (int j 10; j< 15; j++) cout<

<https://www.chegg.com/homework-help/questions-and-answers/question->

AddRolesEventArgs, Components C# (CSharp) Code Examples ...Hero, C...

0.27%

AddRolesEventArgs, Components C# (CSharp) Code Examples ...Hero, Components.Sprite.Role C# (CSharp) Code Examples ...

<https://csharp.hotexamples.com/examples/Components/AddRolesEventArgs/-/php-addroleseventargs-class-examples.html>



[Home](#)

[Blog](#)

[Testimonials](#)

[About Us](#)

[Privacy](#)

Copyright © 2022 Plagiarism Detector. All right reserved