Детектор Плагиата v. 2215 - Отчёт оригинальности: 18.11.2024 16:24:00

Проанализированный документ: Кійко Анна Олександрівна.docx Лицензия: ВОЛОДИМИР МАТІЄВСЬКИЙ_License2

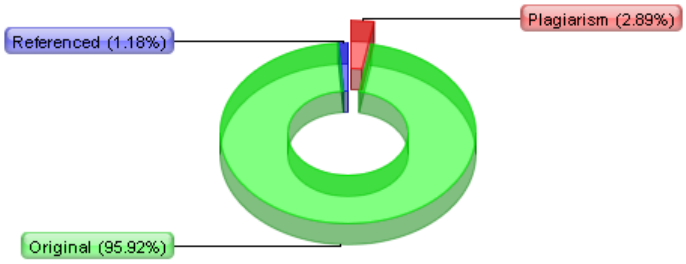
Тип поиска: Поиск переписанного Язык: Uk

Тип проверки: Интернет

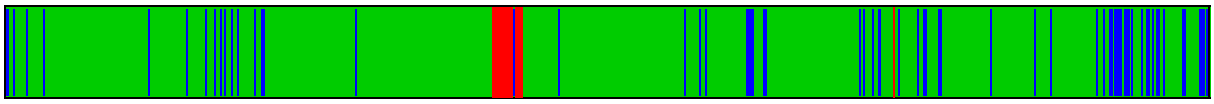
ТЕЕ и кодировка: DocX n/a

Детальный анализ тела документа:

Диаграмма соотношения частей:



Граф распределения зон:



Источники плагиата: 19

3%	508	1.	http://4ua.co.ua/manufacture/qa2ac79b5c43a89421306d37_0.html
2%	253	2.	https://xreferat.com/38/1049-1-nformac-iyno-vim-ryuval-na-sistema-temperaturi.html
1%	206	3.	http://inmad.vntu.edu.ua/portal/static/1D93CE66-A5B3-41E2-A4B4-2A3971C9E5BE.pdf

Детали обработанных ресурсов: 435 - OK / 4 - Ошибок

Важные замечания:

Википедия:	Google Книги:	Сервисы платных работ:	Античит:
[не обнаружено]	[не обнаружено]	[не обнаружено]	**Обнаружено сокрытие!**

Античит-отчет UACE:

1. Статус: Анализатор **Включен** Нормализатор **Включен** сходство символов установлено на **100%**
2. Обнаруженный процент загрязнения UniCode: **12,1%** с лимитом: 4%
3. Процент нераспознанных символов после нормализации: **7,1%**
4. Все подозрительные символы будут отмечены фиолетовым цветом: Abcd...
5. Найдены невидимые символы: 0

Рекомендации по оценке:

Особое внимание следует уделить анализу этого отчета! Предполагается, что этот документ содержит значительное количество символов, чуждых языку документа. Это прямое указание на то, что автор документа использовал специальное программное обеспечение\онлайн-веб-сервис, чтобы эффективно скрыть текст в попытке избежать обнаружения потенциального плагиата. Настоятельно рекомендуется передать это дело на более высокий уровень! В случае сомнений обращайтесь: в службу поддержки Детектора плагиата!

Алфавитная статистика и анализ символов:

Активные ссылки (URL-адреса, извлеченные из документа):

URL не найдены

Исключённые ресурсы:

URL не найдены

Включённые ресурсы:

URL не найдены

Детальный анализ документа:

Міністерство освіти і науки України Державний заклад

Цитирования: **0,03%** id: **1**

«Луганський національний університет імені Тараса Шевченка»

Навчально-науковий інститут математики та інформаційних технологій Кафедра математики та інформатики Кійко Анна Олександрівна ПРОЄКТУВАННЯ СИСТЕМИ КОНТРОЛЮ МІКРОКЛІМАТУ В ОРАНЖЕРЕЇ кваліфікаційна робота здобувача вищої освіти другого (магістерського) рівня освітньої програми

Цитирования: **0,03%** id: **2**

«Комп’ютерні науки та інформаційні технології»

за спеціальністю 122 Комп'ютерні науки Особистий підпис _____ Анна КІЙКО Науковий керівник _____ Владислав КОЗУБ, , доктор філософії, асистент кафедри інформаційних технологій та систем В.о. завідувача кафедри _____ Юрій КОЗУБ, доктор технічних наук, професор кафедри математики та інформатики Полтава – 2025 АНОТАЦІЯ Кійко А. О. Тема: Проєктування системи контролю мікроклімату в оранжереї. Спеціальність: 122

Цитирования: **0,01%** id: **3**

«Комп'ютерні науки».

Установа: ЛНУ імені Тараса Шевченка, 2025р. Магістерська робота містить: 73 с., 35 рис., 12 табл., 55 джерел. Об’єктом дослідження є оранжереї як об’єкт автоматизації та удосконалення систем автоматичного керування. Предмет дослідження – математична модель мікроклімату і система автоматичного контролю параметрів мікроклімату в оранжереї. Мета роботи – розробка автоматизованої системи контролю мікрокліматом оранжереї на основі математичного дослідження мікроклімату. Результати роботи. Проведено аналіз існуючого вигляду пристрою, використовуваного для контролю вологості і температури в оранжереї на основі плати Arduino Mega. Описано необхідні умови для вирощування рослин і представлена математична модель оранжереї при управлінні температурою і вологістю повітря, яка розроблена для застосування в освітніх цілях з метою продемонструвати фізичні явища, що відбуваються в оранжереї і принципи управління мікрокліматом. Описано етапи розробки та робота електронного пристрою автоматизованої системи управління мікрокліматом оранжереї на основі математичного дослідження мікроклімату. Висновки. В результаті розробки у додатку Diptrace була побудована схема друкованої плати, на базі якої було сконструйовано пристрій. Був представлений алгоритм роботи пристрою у вигляді блок-схеми. Ключові слова: МІКРОКОНТРОЛЕР, ТЕМПЕРАТУРА, DS18B20, AM2321, ATMEGA16. ANNOTATION Kiiko Anna Theme: Designing a microclimate control system in a greenhouse. Speciality: 122

Цитирования: **0,01%** id: **4**

"Computer Science".

Institution: Luhansk Taras Shevchenko National University (LTSNU), 2025 year. Master's work of: 74 pages, 35 Fig., 12 Table, 55 source. A research object is greenhouses as an object of automation and improvement of automatic control systems. The subject of research is mathematical model of the microclimate and the system of automatic control of microclimate parameters in the greenhouse. An aim of work is development of an automated greenhouse microclimate control system based on a mathematical study of the microclimate. Job performances. The analysis of the existing appearance of the device used for humidity and temperature control in a greenhouse based on the Arduino Mega board has been carried out. The necessary conditions for growing plants are described, and the mathematical model of the greenhouse is presented in the control of temperature and humidity of air, which is designed for use for educational purposes in order to demonstrate the physical phenomena occurring in the greenhouse and the principles of microclimate management. The stages of development and work of the electronic device of the automated microclimate control system of the greenhouse on the basis of the microclimate math study are described. Conclusions. As a result of the development, the circuit board of the printed circuit board was built in the Diptrace application, on the basis of which the device was designed. The algorithm of the device's operation in the form of a block diagram was presented. Keywords. MICROCONTROLLER, TEMPERATURE, DS18B20, AM2321, ATMEGA16. ЗМІСТ ВСТУП5 РОЗДІЛ І АНАЛІЗ СИСТЕМИ УПРАВЛІННЯ8 1.1. Необхідні умови для вирощування рослин8 1.2. Аналіз сучасних засобів контролю температури і вологості10 1.3. Аналіз системи управління мікрокліматом оранжереї компанії

Цитирования: **0,01%** id: **5**

"Альфа"

на платі Arduino Mega20 1.4. Побудова математичної моделі мікроклімату оранжереї27 1.5. Аналіз динамічних характеристик системи управління мікрокліматом оранжереї33 1.5. Висновки до розділу38 РОЗДІЛ ІІ ПОНЯТТЯ ПРО ТЕМПЕРАТУРУ ТА ВОЛОГІСТЬ ПРИСТРОЇ ДЛЯ ВИМІРЮВАННЯ ТЕМПЕРАТУРИ І ВОЛОГОСТІ40 2.1. Поняття про температуру і про температурні шкали40 2.2. Пристрої для вимірювання температур41 2.3. Пристрої для вимірювання вологості44 2.4. Висновки до розділу54 РОЗДІЛ ІІІ РОЗРОБКА ПРИНЦИПОВОЇ СХЕМИ ПРИСТРОЮ55 3.1. Сімейство AVR55 3.2. Архітектура ATmega1657 3.3. Принцип роботи електронного пристрою59 3.4. Розробка і виготовлення друкованої плати62 3.5. Розробка програмного забезпечення65 3.6. Алгоритм роботи пристрою67 3.7. Висновки до розділу68 ВИСНОВКИ69 СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ70 ДОДАТОК А75 ВСТУП Зростаючий антропогенний вплив на природні об'єкти та зменшення площ природних ландшафтів дедалі більше актуалізує проблему збереження та відновлення ботанічного різноманіття в умовах штучних екосистем. Оранжерейні комплекси можуть стати ефективним інструментом для вирішення цієї проблеми, якщо будуть використовуватися з урахуванням екологічних аспектів та забезпеченням ефективного контролю за мікрокліматом. Так, в Україні в останні роки спостерігається певна тенденція до розвитку оранжерейних комплексів. Це пов'язано з підвищеним інтересом до органічного землеробства та здорового способу життя, а також з бажанням зберегти та відновити біорізноманіття в умовах міського середовища. Застосування сучасних технологій, таких як автоматизовані системи управління мікрокліматом, може значно полегшити процес догляду за рослинами та забезпечити їх ефективний розвиток. Актуальність цієї теми набувається через зростаюче значення тепличних комплексів у сільському господарстві, особливо в умовах міських агломерацій. Однак, для успішного вирощування рослин в оранжереях потрібні сталі, контрольовані умови мікроклімату, що потребує ефективної системи управління мікрокліматом. Такі системи автоматизованого контролю вже зараз широко використовуються в великих ангарах і овочесховищах, теплицях, розраховані на зберігання величезних обсягів продукції. Але поряд з такими величезними сховищами, так само існують і сховища невеликих розмірів, такі як підземні льохи. Такі льохи широко поширені у власників приватних будинків, у яких льох знаходиться безпосередньо в будинку або недалеко від нього. Виходячи з вищесказаного, розробка автоматизованої системи контролю температури і вологості вельми актуальна, адже теплоізоляція вже побудованого льоху часто виявляється дуже трудомісткою і дорогою. Об’єктом дослідження є оранжереї як об’єкт автоматизації та удосконалення систем автоматичного керування. Предмет дослідження – математична модель мікроклімату і система автоматичного контролю параметрів мікроклімату в оранжереї. Мета роботи – розробка автоматизованої системи контролю мікрокліматом оранжереї на основі математичного дослідження мікроклімату. Методи дослідження: техніко-економічний із використанням комп’ютерних технологій, методи моделювання інформаційних процесів, технічний аналіз. Для досягнення поставленої мети необхідно вирішити наступні завдання: розглянути необхідні умови для вирощування рослин; розглянути і проаналізувати способи вимірювання температури і вологості; дослідити математичну модель мікроклімату в оранжереї; розглянути і проаналізувати динамічні характеристики системи управління мікрокліматом оранжереї; розробити принципову схему і сконструювати пристрій; розробити програмний додаток для роботи пристрою. Методи дослідження: техніко-економічний із використанням комп’ютерних технологій, методи моделювання інформаційних процесів, технічний аналіз. Перший розділ містить опис існуючого вигляду пристрою, використовуваного для контролю вологості і температури в оранжереї на основі плати Arduino Mega. Описані необхідні умови для вирощування рослин і представлена математична модель оранжереї при управлінні температурою і вологістю повітря, яка розроблена для застосування в освітніх цілях з метою продемонструвати фізичні явища, що відбуваються в оранжереї і принципи управління мікрокліматом. У другому розділі проаналізувавши способи вимірювання температури було визначено, що рідинні скляні термометри є надійними з високою точністю вимірами. Однак через великого діапазону вимірювання температури, для розроблювального пристрою більш доцільно використовувати датчик DS18B20, в якому за допомогою додаткових обчислень можна досягти такої ж точності, як і в рідинних скляних термометрах. Третій розділ присвячений створенню пристрою. У додатку Diptrace була побудована схема друкованої плати, на базі якої було сконструйовано пристрій. Так само в розділі описана середовище розробки, в якій розроблялося програмний додаток для пристрою. Був представлений алгоритм роботи пристрою у вигляді блок-схеми. РОЗДІЛ І
АНАЛІЗ СИСТЕМИ УПРАВЛІННЯ 1.1. Необхідні умови для вирощування рослин Основні умови, необхідні для росту і розвитку рослин, - тепло, світло, повітря, вода, харчування. Всі ці фактори однаково необхідні і виконують певні функції в житті рослин [1,2]. Життєвий цикл зростання і розвитку ділиться на певні етапи - фази. Умови зовнішнього середовища сильно впливають на ріст і розвиток рослин. Встановлено, що впливом зниженої

температури на проростають насіння і прогріванням сухих насіння можна прискорити розвиток рослин і збільшити урожай. На підставі цього наукою розроблені, а практикою широко використовуються спеціальні рекомендації по прогріванню, пророщування, загартовування насіння, тривалість яких і температура різні, а також залежать від культур. Тепло необхідно рослинам в усі періоди їх росту і розвитку. Вимоги до тепла у різних культур неоднакові і залежать від походження, виду, біології, фази розвитку та віку рослини. Насіння теплолюбних культур проростають при температурі вище 10°С. Такі рослини не переносять заморозків і тривалого похолодання. При температурі нижче 10-12 ° С їх ріст і розвиток припиняються, вони слабшають і швидше уражаються грибними і бактеріальними хворобами. При більш низькій температурі вони гинуть. Найбільш сприятлива температура для росту, розвитку і плодоношення теплолюбних культур вище 20 ° С [3,4]. Насіння холодостійких культур проростають при температурі нижче 10°С. Температура 17-20°С найбільш сприятлива для розвитку і плодоношення рослин цієї групи. При зниженні температури зростання холодостійких культур триває, проте, якщо сходи піддаються тривалому впливу низьких температур (2-0 ° С), багато рослин передчасно викидають квітконосний бутон, не утворюючи ні повноцінного врожаю, ні насіння [5,6]. Основне джерело світла - сонце. Тільки на світлі рослини створюють з води і вуглекислого газу повітря складні органічні сполуки. Тривалість освітлення сильно впливає на ріст і розвиток рослин. Вимоги до умов освітлення у рослин не однакові. Для південних рослин довжина світлового дня повинна бути менше 12 годин (це рослини короткого дня); для північних - більше 12 годин (це рослини довгого дня). Вологість не тільки ґрунту, але і повітря необхідна рослині протягом всього його життя. Перш за все, вода разом з теплом пробуджує насіння до життя. Вода (за обсягом) є головною складовою частиною рослин. Вона бере участь у створенні органічних речовин і в розчиненому вигляді розносить їх по рослині [7]. Рослини особливо вимогливі до вологи, що пояснюється значним її змістом у фруктах (від 65 до 97%, залежно від культури), а також великий випаровує поверхнею листя. Зміст вологи в тканинах листа повинно бути не менше 90-95%. При зменшенні її навіть на 10% листя в'януть, робота їх порушується [8,9]. З повітря рослини отримують необхідний їм вуглекислий газ, який є єдиним джерелом вуглецевого харчування. Вміст вуглекислого газу в повітрі мізерно і становить 0,03%. Збагачення повітря вуглекислим газом йде в основному завдяки виділенню його з ґрунту. Для нормального росту і розвитку рослин потрібні різні елементи живлення. Розглянемо приклад вирощування цитрусових рослин. Для того, щоб виростити цитрусовий сад необхідно докласти певних зусиль і запастися терпінням. Догляд за цитрусовими Взимку ставити контейнер біля вікна не рекомендується, адже від скла буде виходити холод, а від батареї - жар, це негативно позначиться на здоров'я рослини. Для вирощування найкраще підходить тепле вологе місце, куди буде надходити достатньо світла. У весняні та літні періоди навпаки рослина необхідно ближче до вікна, що орієнтоване на південний захід або південний схід. Температурний режим У теплий сезон оптимальна температура становить 19-27 градусів, а взимку можна знизити її до 12-17 градусів. Оскільки цитрусові негативно ставляться до температурних коливань, в осінній період краще завчасно занести рослину в тепле приміщення перед тим, як з'являться перші заморозки. Цитруси родом з тропіків, тому про вологість повітря забувати не варто. Якщо повітря недостатньо зволожений, то придбайте спеціальний зволожувач. Освітленість Цитрусовий сад буде мати потребу в гарному освітленні, захищайте його тільки від прямих сонячних променів спекотного літа. Восени і взимку рослинам не вистачатиме природного світла, тому необхідно забезпечити додатково штучне освітлення, без цієї процедури ви не зможете виростити гідний урожай, щоб його світловий день дорівнював 12 годинам. 1.2. Аналіз сучасних засобів контролю температури і вологості На даний час оранжереї широко скрізь розповсюджені, бо враховуючи непостійні погодно-кліматичні умови з великими і різкими перепадами вологості та температури, тому вирощуванням сільськогосподарської продукції на відкритих ґрунтах займатися проблематично. Для уникнення цих проблем часто використовують різні апарати щоб підтримувати кліматичні показники на задовільному рівні. Сучасні оранжереї, як об'єкти управління, характеризуються нехорошою динамікою та непостійністю параметрів, які випливають зі особливостей технологій виробництва. В той час як агротехнічним нормам потрібна висока точність стабілізації температури (+/-1 градус), а також зміни її завчасно в залежності від значення її фотосинтетично опромінення та періоди розвитку флори. Усім цим обставинам потрібні високі вимоги щодо функціонування та технічного вдосконалення обладнання програмно-апаратного забезпечення [37]. Якщо автоматизувати систему управління мікрокліматом в міні оранжереї це дозволить: зекономити від 15 до 25% потрібного тепла, покращить умови праці робочих, підвищить загальну культуру на виробництві, забезпечить чіткі рамки регулювання мікрокліматичними умовами мініоранжереї, точно забезпечить подачу корисних речовин овочам, що забезпечить збільшення їх врожайності. Забезпечуючи високу продуктивність тепличних господарств потрібно підтримувати чимало параметрів у

певних межах або на певному рівні. До головних параметрів належать: обігрів ґрунту, обігрів повітря в середині теплиці, концентрація вмісту CO2, циркуляція повітря по теплиці, освітленість, вологість та вентиляція [37]. На ринку обладнання можна знайти широкий вибір приладів і фірм які займаються автоматизацією цього процесу і кліматичних показників в оранжереї. До них належать: компанія

Цитирования: **0,01%** id: **6**

«ICP DAS»,

ТОВ

Цитирования: **0,01%** id: **7**

«ФІТО»

, компанія

Цитирования: **0,01%** id: **8**

«ОВЕН»,

Цитирования: **0,01%** id: **9**

«EKF»

і т.д. Зробимо коротку характеристику цих обладнань задля забезпечення потрібних параметрів в оранжереї [37]. EKF – є однією з передових компаній в галузі електротехніки. Вони продають у середньому ціновому сегменті та займаєтюся випуском широкого асортименту низьковольтних та високоякісних продукцій. Їхній пристрі для контролю мікрокліматом зображений на рисунку 1.1. Рис.1.1. Вигляд пристрою управління мікрокліматом Цей пристрій призначений, щоб контролювати основні параметри теплиці, гроубокса чи оранжереї і т.п. З його допомогою контролюється температура, освітлення, вентиляція та полив [37]. Пульт для управління мікрокліматом може працювати в автоматичному режимі підтримує задані вентиляційні і температурні режими: - він керує нагрівачами або вентиляторами; - освітлює у кількох режимах; - задає програми для поливальних і повітряних насосів. Головні переваги використовування цієї системи: простота в установці; гнучкість цієї системи; простота налаштування і експлуатації; можливість використовувати резервне живлення задля зберігання налаштування системи якщо відключиться електроживлення; вологозахист. Можна побачити, що пристрій управління має певні переваги: можливість підбору блоку за необхідними налаштуваннями [37]. У EKF також є 8-ми канальний пульт керування мікрокліматом теплиці на основі мікроконтролера DS 1820, його показано на рисунку 1.2. Рис. 1.2. Вигляд пульта управління мікрокліматом Девайс може змінювати вологість і повітря чи ґрунту у мінітеплиці, вмикати і вимикати насос, нагрівати воду, поливати та вентилювати повітря в теплиці. Керувати навантаженням реалізується за допомогою таймера - ввімкнення навантаження в потрібному проміжку часу, і також здійснювати контроль мікроклімату через встановлення температури (термостатний режим). Рис. 1.3. Блок-схема взаємодії з теплицею На рисунку 1.3 показано блок-схеми роботи пристрою керування мікрокліматом теплиці у цілому. До блоку є підключений насос, що качає воду до баку з резервуара для поливу. У баку вода гріється до потрібної умови і в міру необхідності використовується для поливання. Обігрівання реалізоване з допомогою електричних обігрівачів, що так само можна підключати для забезпечення температурного режиму ґрунту і повітря [37]. Щоб було зручно підключення давачів вологості, силових виходів, температури, і так само інтерфейсу RS 232 (COM - порт ПК) у пульті автоматичного керування були передбаченими конкретні клемові роз'єми. Напруга дається зі спеціального роз'єму на адаптер з напругою 9 - 12 В [37]. Технічна специфікація пристрою управління мікрокліматом: кількість процесів управління: 8; режими керування за таймером: ввімкнення навантаження у заданому інтервалі часу; керування навантаженням по деяких днях в тижні, дням в місяці, або за кількома місяцями; режими керування по температурі (режим термостата): керування як нагрівачем; керування як охолоджувачем; убудований таймер із світлом і звуком (підсвічення дисплею): входи для підключення до датчиків; цифровий вхід щоб підключати датчики: 32 датчиків максимум; аналогові входи: 2. енергонезалежний годинник, що працює в реальному часі (та календар зі врахуванням високосних років) до 2100; зберігання всіх налаштовувань у енергетично незалежній пам'яті; подовження коректної роботи програми якщо було тимчасове відключення від інтернету; інші виходи: 8 оптично-ізольованих 7-сторних 2-амперних каскадів із можливим підключенням силових 7-сторов для керування більш потужнішими навантаженнями (більше двох А); 1 віддалене керування термостатом за допомого. COM-порта комп'ютера із допомогою конкретного ПЗ [37]. Кількість налаштувань, що контролюють той модуль позволяє користовуватися ним в міні теплицях. Свого роду його можна називати одним з найкращих. Невелика вартість та простота у використанні є позитивним плюсом. Недоліком є те, що його складно у

програмувати. Програмування модуля на мовах як низького так і, високого рівня, але потрібно врахувати синтаксис певних мов [37]. Для кращої реалізації процесу автоматичного керування компанія

Цитирования: **0,01%** id: **10**

«ICP DAS»

дає пропозицію використання інтелектуального модуля вводу/виводу WISE-7118 Z. Вигляд розумного пристрою автоматизації приведено на рисунку 1.4. Рис. 1.4. Повний вигляд розумного модуля вводу/виводу WISE-7118Z У цього модуля є 6 каналів цифрового виводу та 10 аналогового вводу. Юзер має можливість змінювати канали в різному діапазоні напруги і струму, і так само різні типи термічних пар для вимірювання самого мікроклімату теплиці, наприклад, вологість або температуру [37]. З допомогою цифрових виходів є можливість управляти обігрівачем, люком-вентиляцією, поливним процесом, освітленням та системою випарного охолоджування. Інтелектуальний пульт WISE-7118 Z буде постійно контролювати становище мікроклімату в теплиці і в залежності від змін у параметрах освітленості, ґрунту, повітря використовувати певне керування [37]. Окрім того розумний модуль WISE використовує протокол для передачі даних Modbus TCP Slave. Що позволяє об'єднувати ці системи у єдиний диспетчерський вузол, де використовуючи систему SCADA, користувач має можливість керувати усім процесом та отримувати правильну інформацію про становище в кожній теплиці мікроклімату. На рисунку 1.5 показано блок-схему з’єднання інтелектуального пристрою з обладнанням у теплиці [37]. Рис. 1.5. Блок-схема керування мікрокліматом в теплицях Переваги у використанні WISE: – щоб створити конфігурацію керування у модулі WISE використовується логіка IF-THEN- ELSE. Користувач має можливість користуватися 36 правилами IF-THEN-ELSE. Утворивши правила і завантаживши їх на модуль, вони мають виконувати певні рішення із настанням певної умови. Усі потрібні дії налаштування утворюються у звичайному Web-браузері за кілька хвилин: підтримує Modbus TCP для інтегрування у систему SCADA; WISE-7118 Z використовує технологію POE живлення, яка звільняє його від необхідного прокладування кабельних ліній для отримання живлення. Фірма

Цитирования: **0,01%** id: **11**

«ФІТО»

дає пропозицію систем керування мікрокліматом у теплицях серії FC. А зокрема повний вигляд цього мікрокліматичного комп’ютера від тієї фірми зображено на рисунку 1.6 [37]. Рис. 1.6. Повний вигляд мікрокліматичного комп’ютера Це є оптимальним рішенням для будь-якого типу теплиць різних за розмірами як скляних, так і плівкових. Для керування головними параметрами теплиці, її обладнують, так сказати, виконуючими системами: системою вентиляції, обігрівання, освітленості, системою додавання CO2. Строге виконання головних параметрів клімату - це запорука високого врожаю та стійкості флори до захворювань. Але час не стоїть на місці, а також і технології вирощування, тому на нинішній день найкращі агрономи та технологи приділяють велику кількість уваги до розширеного набору показників мікроклімату, що включає у себе вологість листа, температуру листа, розподілення температури у повітрі по вертикальному зрізу теплиць, зміна швидкості руху в повітрі. Компанією

Цитирования: **0,01%** id: **12**

«ФІТО»

є розроблені клімат-комп'ютери, що підтримують не лише головні показники клімату, але й позволяють змінювати згадану розширену кількість показників [37]. Архітектура клімат-комп'ютера позволяє повністю у автоматичному режимі управляти типами контролюючих систем теплиць з строгим дотриманням певного агрономного режиму. Підвищуючи ціни на енергоносії змушують не тільки забезпечувати підтримання мікроклімату, але і думати про ефективне використання ресурсів, хоч то ввімкнення системи опалення або освітлення, чи подача CO2 чи постійне відкриття/закриття фрамуг. У зв'язку з чим функціонал можливостей систем керування позволяють утворювати

Цитирования: **0,01%** id: **13**

«стратегію керування»,

де агроном має можливість від залежності фази росту рослини і /або економічної доцільності вибирати пріоритет економії енергетичних витрат чи як найбільшого дотримування технології [37]. Принцип роботи мікрокліматичного комп’ютера. Ядро системи це промисловий контролер керування, розроблений спеціалістами фірми

Цитирования: **0,01%** id: **14**

«ФІТО»

конкретно для усіх видів теплиць. Завдяки сучасній елементній базі з японських і

американських комплектуючих, у контролерів існує великий показник безперебійності та хорошої роботи. Окрім контролера, система керування кліматом включає у себе підсистему встановлених всередині теплиці вимірювальних датчиків [37]. Якщо є необхідність система автоматично інтегруватися з котельні. Для того існує конкретний модуль, котрий через інтерфейс FIDUFACE здійснює передачу даних до котельні задля керування вироблення тепла, або CO2 чи електроенергії. Спостерігати за процесом клімату і також вносити поправки до завдань у зручній формі можна за допомогою звичайного комп'ютера [37]. Так само є доступна функція віддаленого керування системою через мережу інтернет. Функціональні можливості: вимірювання параметрів клімату в декількох зонах; повний автоматичний контроль систем опалення, вентиляції зашторювання, CO2, освячування; створення оптимальної

Цитирования: **0,01%** id: **15**

"стратегії управління"

; інтеграція в систему управління котельні (FIDUFACE); зручний інтерфейс; функція економії енергетичних ресурсів; віддалений моніторинг і аналіз з ПК [38]. Таким чином проаналізувавши всі вище перераховані засоби для підтримання мікроклімату в оранжереї, можна зазначити, що вибір на сьогоднішній день є досить різноманітним і користувач може вільно вибирати параметри, які повинні його задовольняти. Всі засоби мають у дуже великий потенціал [38]. Пульт управління мікрокліматом компанії

Цитирования: **0,01%** id: **16**

«EKF»

варто застосувати у невеликих теплиць. Хоча пульт може виконувати такі дії як: контроль освітленості, полив, регулювання температури у теплиці і забезпечення її вентиляцію, але кількість параметрів можна сказати є середньою в порівнянні з його аналогами [38]. Ще одне творіння цієї фірми - це блок керування мікрокліматом на мікроконтролері DS1820. Серед переваг, як і у попередника, можна віднести малі розміри, простота в експлуатації, регулювання різними параметрами в залежності від підключених датчиків. Кількість підключених датчиків сягає 32, що при правильному їх підборі можливо повність автоматизувати процес регулювання мікроклімату на одному модулі. Вісім виходів дозволяють підключити пристрої для управління більш потужними навантаженнями. Можливість підключення до комп'ютера дає змогу користувачу відразу, в реальному часі, відслідковувати всі параметри і швидко реагувати у разі необхідності. Блок можна використовувати і для малих, і для великих теплиць [38]. Найбільш досконалим у своєму роді, з нашої точки зору, є інтелектуальний модуль WISE- 7118Z. Не зважаючи на його малі розміри, він може контролювати достатню кількість параметрів, але з додатковими можливостями. А саме: постійний контроль за мікрокліматом у теплиці, можливість використання режиму передачі Modbus TCP, що дозволяє збирати дані про стан у теплиці і відображати це на моніторі користувача. Це досить корисно, коли контроль ведеться за кількома теплицями одночасно. Хоча кількість датчиків, які можна під'єднати до контролера тільки 10, але з використанням комутаторів, таких як RSM-408 і NS-205PSE можна збільшити кількість модулів підключення. Дані подаються на сервер і обробляються у SCADA системі. Передбачено також логічне управління подіями, що додає мікроконтролеру додаткові переваги [38]. Останнім серед розглянутих прикладів був кліматичний комп'ютер фірми

Цитирования: **0,01%** id: **17**

«ФИТО».

Не зважаючи на його великі розміри, він є дуже потужним у плані параметрів, які контролюються. Він добре підходить як для великих так і для малих теплиць, контролює основні параметри, має додаткові можливості контролю температури листка, вологості листка, контролює розподіл температури по теплиці і т.д. Кліматичний комп'ютер є економічно вигідним, а його архітектура зводить до мінімуму процент втручання людини у роботу системи [38]. З нашої точки зору, закордонні виробники випускають кращу і більш надійну продукцію, а тому, як наслідок, її ціна і якісь є вищою ніж інших вітчизняних виробників. Якщо кошторис не обмежується, то більш раціональним буде застосування кліматичного комп'ютера фірми

Цитирования: **0,01%** id: **18**

«ФИТО»,

або пристрою на інтелектуальному модулі WISE-7118Z. Ці два представника призначені як для великих, так і для середніх теплиць, а тому в повній мірі задовольнять всі основні вимоги до автоматизації мікроклімату в теплиці [38]. В свою чергу пульт управління мікрокліматом компанії

« ЕКА » Цитирования: **0,01%** id: **19**

і блок керування мікрокліматом на мікроконтролері DS1820 краще застосовувати для малих теплиць і теплиць середнього розміру. 1.3. Аналіз системи управління мікрокліматом оранжереї компанії "Альфа" на платі Arduino Mega" Існують різні способи зберігання рослин і цитрусових плодів: в закритих або відкритих контейнерах. Вибір способу вирощування та зберігання залежить від виду оранжерей або теплиць, стану закладаються паростків і насіння, строків реалізації продукту і від оснащеності господарств технічними засобами, але в будь-якому варіанті в умовах промислового і приватного зберігання застосовуються регулюючі системи мікроклімату. Автоматизовані системи забезпечують відведення продуктів обміну речовин, придушення підвищеної активності дихання, проростання і гниття, запобігання рослин від замерзання, а також збереження цінних органічних речовин, включаючи воду. Для цих цілей була розроблена система управління мікрокліматом в оранжереях (див. Рис. 1.7). Дана система є частиною реального присадибного господарства. З її допомогою процес вирощування рослин став частково автоматизованим, що не вимагає постійної присутності людини. Конкретний екземпляр даної системи відпрацьовується на каркасно-скляній оранжереї, довжиною 6 метрів, шириною 3 метри, висотою 2 метри. В оранжереї є одні двері і 2 кватирки, проведені електрика та водогін. Нагрівання води відбувається в ємності об'ємом 70 літрів. Тиск в ємності складає близько двох атмосфер. В оранжереї вирощується близько 35 рослин [10] Система має наступний вигляд: Рис. 1.7. Схема системи управління мікрокліматом в оранжереї Центральне місце в системі займає плата Arduino Mega (рис. 1.8): Рис. 1.8. Arduino Mega Arduino є повністю відкритою платформою, що складається з плати і середовища розробки, в якій реалізована перероблена версія мови Processing / Wiring. Використовувана апаратна платформа побудована на мікроконтролері ATmega1280. У даній системі задіяні 8 цифрових входів / виходів (всього на платформі їх 54) і 10 аналогових (всього їх 16). Плата отримує харчування від зовнішнього джерела живлення. Плата має наступні характеристики: • робоча напруга: 5В; • рекомендована вхідна напруга: 7-12 В; • гранична вхідна напруга: 6-20 В; • 54 цифрових портів введення / виводу; • 16 аналогових входів; • струм споживання на одному виводі: до 40 мА; • струм споживання виведення 3.3В: 50 мА; • пам'ять Flash Memory: 128 KB, з яких 4KB використовуються загрузчиком; • ОЗУ: 8 KB; • незалежна пам'ять: 4 KB; • тактова частота: 16 МГц; • розмір: 75x54x15 мм; • вага: 45 г; До Arduino Mega підключені необхідні датчики і модулі. Включення / вимикання поливу залежить від ряду параметрів: • вологість ґрунту; •температура води; •час доби. У даній системі задіяно 4 датчика вологості ґрунту (див. Рис. 1.9 - 1.10). Для вимірювання вологості ґрунту використовується саморобний датчик, що представляє собою два цвяха і резистор. Принцип дії заснований на залежності електричного опору ґрунту від її вологості. Цвяхи, введені в ґрунт на деякій відстані один від одного, виступають в якості щупів, між якими перевіряється опір. За підсумковим аналоговому сигналу можна судити про ступінь вологості. Схема датчика представлена на рисунку: Рис. 1.9. Датчик вологості ґрунту Для вимірювання температури води використовується LM335Z - аналоговий термодатчик термостабілітрон: Рис. 1.10. Аналоговий термодатчик LM335Z Використовуваний датчик має такі характеристики: • діапазон: -40 ... + 100; • точність: 1 ° С; • залежність: 10мВ / оС. Для підключення датчика до плати потрібен резистор, опором 2.2 кОм. Ставлячи струм через датчик в діапазоні від 0.45 мА до 5 мА (резистором R1), отримуємо напругу на датчику, яке в десятках мВ представляє абсолютну температуру в градусах Кельвіна. Схема підключення представлена на рисунку 1.11: Рис. 1.11. Схема підключення термодатчика Для того, щоб полив включався тільки в темний час доби, використовуються 2 датчики світла Light Sensor-BH1750 (див.рис. 1.12): Рис. 1.12. Датчик світла Light Sensor-BH1750 Даний датчик служить для вимірювання освітленості в межах від 1 до 65535 люкс. Він має наступні характеристики: • напруга живлення: 3-5в; • дозвіл: 16 біт; • габарити: 19x14x3 мм; • похибка: ± 20%. Підключення датчика здійснюється наступним чином, як показано на рисунку 1.13: Рис. 1.13. Підключення датчика світла Light Sensor-BH1750 Коли отримані з датчиків показники задовольняють певним умовам (вони розрізняються для кожного виду рослин), включається полив. Для регулювання поливу використовується електромагнітний клапан. Він підключається до плати за допомогою реле (див. Рис.1.14). А саме використовується релейний модуль для Arduino проектів Relay Module 2 DFR0017. Він використовує високоякісне реле Omron G5LA. Стан виходу реле відображається за допомогою світлодіода. Цей модуль управляється за допомогою цифрового порту вводу-виводу. Час перемикання контакту становить 10 мс. Як і датчики для вимірювання температури і вологості грунту, релейний модуль підключається в керуючої електроніці через три дроти: Рис. 1.14. Призначення контактів роз'єму релейного модуля Крім поливу дана система контролює і температуру повітря в теплиці. Для одночасного вимірювання температури і вологості повітря використовується датчик DHT11 Temperature Humidity

Sensor (див. Рис. 1.15). Рис. 1.15. DHT11 Temperature Humidity Sensor Він підключається до керуючої електроніки через три дроти: живлення (VCC), земля (GND) і сигнальний. На платі крім датчика розташований мікроконтролер, в пам'яті якого записані калібрувальні поправки для датчиків. Сигнал з пристрою передається по шині у цифровому вигляді. Це дозволяє передавати дані на відстань до 20 м. Даний датчик має такі характеристики: • напруга живлення: 5 В; • діапазон температур: 0-50 ° С, похибка ± 2 ° С; • вологість: 20-90%, похибка ± 5%. Для регулювання температури повітря в теплиці використовується два режими: пасивне і активне провітрювання. Пасивне провітрювання є відкриття / закриття кватирок, а активне - включення / вимикання вентилятора. Відкриття кватирок проводиться за допомогою двох (по одному на кватирку) сервоприводів Futaba T306 MG995 (див. Рис. 1.16): Рис. 1.16. Сервопривід Futaba T306 MG995 Використовуваний сервопривід має наступні характеристики: • швидкість роботи: 0.17 с / 60 градусів (4,8 В без навантаження); • момент: 13 кг-см при 4,8 В; • момент: 15 кг-см при 6 В; • робоча напруга: 4,8 - 7.2 В; • довжина дроту: 300 мм; • розміри: 40мм х 19мм х 43 мм; • вага: 55 м Підключення вентилятора проводиться таким же способом, як і підключення клапана (через релейний модуль). Отримані з датчиків дані записуються на карту пам'яті SD (див. Рис. 1.17). Надалі вони обробляються, аналізуються і на їх основі будуються графіки різних показань. Для цього використовується модуль SD-карт DFRobot: Рис. 1.17. Модуль SD-карт Підключення вентилятора проводиться таким же способом, як і підключення клапана (через релейний модуль). Він містить роз'єм для стандартних карт пам'яті SD, що дозволяє додати накопичувач для запису і зчитування даних в будь-який проект. Він має наступні характеристики: • роз'єм для стандартних SD карт і через перехідник MicroSD карт; • містить фіксатор карти пам'яті; • підтримує читання і запис; • може використовуватися з іншими микроконтроллерами; • напруга живлення: 5 В; • розмір: 36 х 30 х 5 мм; • вага: 7 гр. У даній системі управління враховані вимоги технології вирощування та поливу, логіка алгоритмів взаємопов'язана і оптимізована. Функціонал простий і не викликає труднощів при його освоєнні. Протокол управління легко запам'ятовується, не створює труднощів при виборі режимів, читанні інформації та управлінні системою. 1.4. Побудова математичної моделі мікроклімату оранжереї Відомо, що типові рішення по управлінню режимами роботи об'єктів різної природи засновані на використанні досить простих моделей об'єктів, параметри яких абстрактні і не відображають фізичні закономірності процесів [11]. При використанні математичних моделей такого типу вдається успішно вибрати структуру і (зазвичай) закон управління, але не вдається обґрунтовано визначити реалістичні показники якості управління. Крім того, в зв'язку з абстрактним характером параметрів спрощених моделей не вдається намітити шляхи поліпшення автоматизованого технологічного процесу не тільки за рахунок налаштування параметрів системи управління, а й за рахунок зміни характеристик самого об'єкта. У зв'язку з цим представляють інтерес роботи, в яких для синтезу системи управління і для аналізу її якості з вибором напрямків вдосконалення автоматизованого технологічного комплексу використовуються різні моделі: більш прості і універсальні - для задач синтезу і більш складні, що відображають фізичні основи роботи об'єкта - для задач аналізу і вдосконалення об'єкта спільно з системою управління [12]. Однією з важливих стадій створення системи управління температурно-вологісним режимом оранжереї є розробка моделей об'єкта, що відбивають що відбуваються в ній процеси з позицій вирішення завдань двох класів синтезу алгоритмів, з одного боку, і аналізу якості управління, з іншого боку. Якщо вимогою до моделей для другого класу задач є адекватність, то при розробці моделей для першого класу задач повинні бути враховані не тільки вимоги адекватності, а й досягнутий рівень наукової підтримки задач синтезу алгоритмів. Відповідно до цієї класифікації розділимо існуючі моделі мікроклімату на дві групи: • Принципові моделі, орієнтовані на рішення задач аналізу властивостей об'єкта і якості систем управління. У моделях даної групи фізичні явища описуються диференціальними рівняннями (зазвичай в просторі станів). Параметри в моделях цієї групи мають фізичну інтерпретацію. • Моделі

Цитирования: **0,01%** id: **20**

"чорного ящика",

орієнтовані на рішення задач синтезу алгоритмів керування. Залежно від конкретного завдання синтезу використовуються або статичні моделі (регресивні, поліноміальні, засновані на використанні нейронних мереж, нечітких множин), або динамічні моделі (зазвичай - у формі диференціальних рівнянь, коефіцієнти яких визначаються за експериментальними даними методами ідентифікації; явний зв'язок цих параметрів з фізичними і конструктивними параметрами оранжереї відсутня). В якості основи для розробки моделей обох класів в даній роботі використана принципова модель мікроклімату. Модель в безперервному часу розроблена на основі наступних спрощують припущень: 1) Модель інтерпретує оранжерею як заданий обсяг повітря, обмежений

стінами, дахом і підставою (землею). Просторовий розподіл змінних, що описують мікроклімат, не враховується. Дане припущення виправдовується при постійній конвекції повітря і примусової вентиляції. 2) Зміна біомаси рослин в процесі їх розвитку розглядається мною як зовнішній фактор, не пов'язаний з показниками мікроклімату. Це припущення виправдовується тим, що система управління підтримує показники мікроклімату згідно з вимогами до технології вирощування культур. Біомаса рослин в роботі є постійним значенням. 3) Зміни параметрів моделі в часі відбуваються настільки повільно, що при описі циклу динамічних процесів в системі управління їх можна вважати постійними. Відповідно до цих припущеннями модель мікроклімату отримана як детермінована і зосереджена, виходячи з співвідношень масового і теплового балансу з наближеним урахуванням стадій біологічного розвитку рослин (форма обліку - зміна параметрів моделі в часі) [13,14]. У моделі виділяються дві підсистеми: повітря оранжереї і навколоґрунтовий шар з рослинами і ґрунтом. При описі підсистем використовуються наступні змінні: температура повітря в оранжереї, температура рослин оранжереї (температури навколоґрунтового шару), вологість (відносна або абсолютна) повітря оранжереї [15]. Рівняння масового балансу води в оранжереї має вигляд: ,(1.1) де ρ - щільність повітря оранжереї, (кг / м3); V - об'єм повітря оранжереї, (м3); X (t) - абсолютна вологість в оранжереї, (кгвода / кгвоздух); t - час, (с); F (t) - витрата води в повітряному потоці, які пройшли через віконні щілини (кгвода / с); C_sat (t) - коефіцієнт насичення повітря; E (t) - швидкість сумарного випаровування води рослинами (кгвода / с); fog (t) - витрата води системою туману (кгвода / с). Співвідношення (1.1) показує, що зміна кількості води у внутрішньому повітрі оранжереї залежить від наступних складових: • кількості води, що надходить із зовнішнім повітрям через віконні щілини - F (t); • кількість води, що випаровується рослинами в процесі свого зростання - E (t); • кількості води розпилюється системою туману - fog (t) [16]. Рівняння теплового балансу енергії, що впливає на зміну температури внутрішнього повітря оранжереї має вигляд: , (1.2) де ρ - щільність повітря оранжереї, (кг / м3); V - об'єм повітря оранжереї, (м3); Cv - теплоємність повітря, [Дж / (кг · ° С)]; T (t) - температура повітря всередині оранжереї, (° С); t - час, (с); Qs (t) - сонячна енергія, що передається повітрю оранжереї, (Вт); Qcc (t) - енергія обміну при провідності і конвекції, (Вт); Qp (t) - обмін енергією з рослинами, (Вт); C_sat (t) - коефіцієнт насичення повітря; Qu (t) - втрати енергії за рахунок сумарного випаровування води рослинами, (Вт); Qt (t) - втрати енергії за рахунок розпилення води системою туману, (Вт); Qv (t) - енергія обміну повітряної вентиляції, (Вт); W (t) - енергія системи обігріву, (Вт). Співвідношення (1.2) показує, що зміна температури внутрішнього повітря оранжереї залежить від наступних складових: • сонячної енергії, що нагріває внутрішнє повітря оранжереї - Qs (t); • температури повітря навколишнього середовища, що надходить в оранжерею через систему вентиляційних фрамуг і щілини - Qv (t); • температури повітря навколишнього середовища, що впливає на обмін енергією через стіни шляхом провідності і конвекції - Qcc (t); • сумарного випаровування води рослинами, що призводить до охолодження повітря оранжереї -Qu (t); • енергії системи обігріву - W (t); • втрат енергії від розпилення води системою туману - Qt (t); • теплообміну з рослинами - Qp (t). Рівняння теплового балансу енергії, що впливає на зміну температури рослин оранжереї, має вигляд: , (1.3) де Soutside - площа поверхні оранжереї, (м2); Cp - теплоємність рослин, [Дж / (° С · м2)]; Tp (t) - температура рослин всередині оранжереї, (° С); t - час, (с); Qur (t) - енергія, засвоюється рослинами протягом дня, (Вт); Qp (t) - обмін енергією з рослинами, (Вт); Qg (t) - втрати енергії через грунт, (Вт). Співвідношення (1.3) показує (1.4), що зміна температури рослин оранжереї залежить від наступних складових: • енергії, засвоюваній рослинами - Qur (t); - втрат енергії через грунт - Qg (t); • теплообміну повітря з рослинами - Qp (t). Таким чином, система рівнянь (1.4), наближено описує мікроклімат оранжереї, має вигляд: , (1.4) Модель (1.4) є наближеною. Не враховується просторовий розподіл температури і вологості по площі оранжереї; розподіл цих змінних по висоті враховується тільки агрегованих; вплив температурно-вологісного режиму на показники розвитку рослин зведено до детермінованої залежності коефіцієнтів моделі від часу. Але, виходячи з мети використання моделі (для аналізу і синтезу алгоритмів керування, а не для вибору технологічно доцільних режимів зміни температури і вологості), представлена на рисунку 1.18 модель прийнятна [17]. Рис. 1.18. Математична модель температурно-вологісного режиму оранжереї 1.5. Аналіз динамічних характеристик системи управління мікрокліматом оранжереї Дослідження математичних моделей оранжереї як об'єкта управління в останні десятиліття істотно просунулися. Найбільш прості моделі, представлені у вигляді апериодических ланок першого або другого порядку з запізненням. Такі моделі є досить грубими і наближене описують поведінку об'єкта управління при регулюванні одного параметра, наприклад температури [18]. Існують моделі більш складної структури, наприклад, для управління двома і більше параметрами. Незаперечною перевагою таких способів завдання математичних моделей є наочність моделювання та можливість застосування класичних методів синтезу регуляторів систем

управління параметрами мікроклімату. У свою чергу інший підхід до завдання математичних моделей оранжереї базується на описі цієї складної динамічної системи, де такі параметри як температура і вологість пов'язані нелінійними законами термодинаміки, за допомогою моделі в просторі станів такої форми: x f (t,x,u,v), (1.5) де f позначає нелінійну функцію; t - час; x - змінні стану, такі як температура повітря в теплиці, його вологість, концентрація діоксиду вуглецю; u - керуючі впливи, такі як обігрів повітря калориферами або іншими пристроями, природна або примусова вентиляція, системи освіти туману, а також системи збагачення CO2; v - зовнішні впливи, наприклад, температура і вологість зовнішнього повітря, сонячна радіація, вітрове вплив та ін. Модель в просторі станів більш точно описує процеси, що відбуваються в оранжереї, враховуючи взаємозв'язок змінних стану, зовнішніх збурень і сигналів управління, проте в свою чергу суттєво ускладнюється моделювання внаслідок нелінійності моделі. Це дослідження має на меті дослідження математичної моделі оранжереї при управлінні температурою і вологістю повітря, що базується на законах збереження енергії і маси. А також розробку Simulink-моделі для застосування в освітніх цілях для демонстрації фізичних явищ, що відбуваються в оранжереї і принципів управління мікрокліматом. dTвнутрt dt= 1ρCвVmQнt+Snt-λQm(t)-υвtVm+kт.огорρCвVmTвнутрt-Tзовнt,(1.6) dφвнутр(t)dt=1Vв Qmt+1VвESnt,φвнутр(t)-υвtVmφвнутрt-φзовнt,(1.7) ESnt,φвнутр(t)=Sn(t)λ-φвнутрt,(1.8) де Tвнутр, Tзовн - температура повітря всередині і ззовні оранжереї відповідно (oC); φвнутр, φзовн - відносна вологість повітря всередині і зовні оранжереї відповідно (%); kт.огор - коефіцієнт теплопередачі матеріалу огорожі оранжереї (Вт / К); V - повний геометричний об`єм оранжереї (м3); Vт, Vв - об`єми оранжереї що обігріваються та зволожуються відповідно (м3). Зазвичай складають не більше 60% -70% загального обсягу оранжереї; ρ - щільність повітря (1,2 кг / м3); Св - питома теплоємність повітря (1,005 кДж • кг-1 • K-1); Qн - потужність обігрівачів повітря оранжереї (Вт); Qт - продуктивність системи туманоутворення (гвода / с); Sп (t) - поглинене теплицею сонячне випромінювання (Вт); λ - питома теплота пароутворення (2256 кДж / кг); υв(t) - повітрообмін, який забезпечувався б системою вентиляції оранжереї (м3 / с); E Sп (t), внутр (t) - евапотранспірації рослин, в функції від поглинання сонячного випромінювання і вологості повітря в оранжереї (гвода / с); , – масштабні коефіцієнти, що приймаються за постійні. Для спрощення моделі зазвичай вважається, що виділені рослинами випаровування більшою мірою залежать від сонячної радіації та меншою від вологості повітря, тому в рівнянні (3) нехтують доданком φвнутр(t). Підставляючи (3) в (1), після перетворень отримаємо dTвнутрt dt= 1CвQнmaxQнt+Snt-λ`Qm(t)-υвtVm+kт.огорρCвVmTвнутрt-Tзовн(t),(1.9) dφвнутр(t)dt=1V`Qmt+`Snt-υвtTυφвнутрt-φзовнt,(1.10) де, Qн=QнQнmax, Qm=QmQmmax, υв=υвυвmax – нормалізованні управляючі змінні; λ`=λQmmax,V`=VmQmmax, `=Vвλ – нормалізовані коефіцієнти; Tv=VвυвmaxT - час, необхідний для повної заміни повітря у зволоженій частині оранжереї. Matlab-Simulink є інтегрованим середовищем для моделювання, аналізу та симуляції динамічних систем. У Simulink блоки групуються за їх призначенням та функціональністю [6]. Основні групи блоків в Simulink: Sources (Джерела) - це блоки, які входять до системи і постачають вхідні сигнали. Sinks (Приймачі) - ці блоки є виходами системи і приймають вихідні сигнали. Linear Algebra (Лінійна алгебра) - ці блоки дозволяють виконувати матричні операції та розв'язувати системи лінійних рівнянь. Nonlinear (Нелінійні) - ці блоки дозволяють моделювати нелінійні елементи, такі як гістерезисні елементи, сатурацію та інші. Discrete (Дискретні) - ці блоки дозволяють моделювати системи з дискретним часом. Continuous (Неперервні) - ці блоки дозволяють моделювати системи з неперервним часом. Signal Attributes (Характеристики сигналу) - ці блоки дозволяють встановлювати параметри сигналів, такі як амплітуда, частота та інші. Modeling Constructs (Конструкції моделювання) - ці блоки дозволяють створювати складні моделі та виконувати різноманітні операції над блоками. Math Operations (Математичні операції) - ці блоки дозволяють виконувати математичні операції над сигналами. User-Defined Functions (Користувацькі функції) - ці блоки дозволяють використовувати користувацькі функції для обробки сигналів. Крім цих основних груп, у Simulink є багато інших груп блоків, які дозволяють моделювати різноманітні системи та операції. На базі системи рівнянь (1.9) - (1.10) в даному дослідженні була розроблена блочно-імітаційна модель в середовищі Simulink програмного комплексу Matlab, представлена на рисунку 1.19 [19]. Рис. 1.19. Математична модель оранжереї в середовищі Matlab-Simulink На схемі виділені три групи блоків. До групи 1 входять блоки відповідно керуючим впливам. Група 2 об'єднує блоки впливів, що обурюють. У третю групу об'єднані блоки констант-параметрів конкретної оранжереї. Решта блоки які призначені для виконання математичних операцій та для візуалізації результатів моделювання. У цьому прикладі була промоделювати оранжерея на 100 м2 висотою 2,5 м, з параметрами, представленими в таблиці 1. Всі параметри наведені до одного квадратного метра площі оранжереї. Встановлені в блоках інтегрування початкові значення вологості і температури: 15 oC, і 50%. Таблиця 1.1

Параметри моделі оранжереї Найменування Значення Cв -324,67 хв•Вт/оС kт.огор 29,81Вт/оС Tv 3 хв λ` 460 Вт α` 0.0033 г/(м3•хв•Вт) V` 13,3 г/(м3•хв) Моделювання проводилося в середовищі Matlab / Simulink на базі операційної системи Windows 10. Для інтегрування використовувався метод Рунге-Кутта 4-го порядку з фіксованим кроком 1 хвилина, діапазон інтегрування 1 доба або 1440 хвилин. При моделюванні всі параметри математичної моделі оранжереї вважалися постійними, зовнішні обурення моделювалися підсистемами (див. Рис. 1.14), в які входять в якості основних генеруючих ланок: постійна, гармонійна і випадкова складові. В даному випадку, представлений блок завдання температури навколишнього середовища у вигляді такої залежності. Tвнешн(t) Tср T sin(t) (t) 25 10sin(0,0044t) (t), (1.11) де Tср = 25 ос середня температура; T = 10 оС - амплітуда коливань температури протягом доби; ω = 2π / 1440 частота гармонійної складової; δ (t) - випадкова складова (білий шум). Рис. 1.20. Модель завдання обуреного впливу Подібним чином моделювалися коливання вологості, а також сонячного випромінювання протягом доби. Результати моделювання представлені на рисунку 1.21. Рис. 1.21. Результати моделювання температури і вологості повітря всередині оранжереї 1.5. Висновки до розділу Перший розділ містить опис існуючого вигляду пристрою, використовуваного для контролю вологості і температури в оранжереї на основі плати Arduino Mega. Математична модель мікроклімату оранжереї може бути корисним інструментом для дослідження та прогнозування різних аспектів фізичного середовища, таких як температура, вологість, освітленість та інші. Одна з можливих математичних моделей мікроклімату оранжереї може бути базовою фізичною моделлю теплообміну та вологопереносу між рослинами, повітрям, ґрунтом та іншими компонентами системи. Модель може включати різні параметри, такі як теплова ємність, теплопровідність, теплові потоки, вологопровідність, вологовміст, вологові потоки та інші. Для побудови математичної моделі можна використовувати експериментальні дані з різних датчиків, що вимірюють різні параметри мікроклімату, такі як температура, вологість, освітленість, рівень CO2 та інші. Дані можуть бути зібрані протягом тривалого періоду часу, щоб врахувати різні сезонні та денні зміни. Після зібрання даних можна використати різні математичні методи, такі як лінійна регресія, нейронні мережі, генетичні алгоритми та інші, для побудови моделі. Після цього модель може бути перевірена за допомогою інших експериментів та даних. Отримана математична модель може бути використана для прогнозування мікроклімату оранжереї та планування оптимальних умов для рослинного росту. Вона також може бути використана для оптимізації системи управління мікрокліматом та для зменшення витрат на енергію та ресурси. РОЗДІЛ II
ПОНЯТТЯ ПРО ТЕМПЕРАТУРУ ТА ВОЛОГІСТЬ
ПРИСТРОЇ ДЛЯ ВИМІРЮВАННЯ ТЕМПЕРАТУРИ І ВОЛОГОСТІ 2.

Обнаружен Плагиат: **1,65%** http://4ua.co.ua/manufacture/qa2ac7... + 4 ресурсов! id: **21**

1. Поняття про температуру і про температурні шкали Температурою називають величину, що характеризує тепловий стан тіла. Згідно кінетичної теорії температуру визначають як міру кінетичної енергії поступального руху молекул. Звідси температурою називають умовну статистичну величину, прямо пропорційну середньої кінетичної енергії молекул тіла [22]. Усі пропоновані температурні шкали будувалися (за рідкісним винятком) однаковим шляхом: двом (щонайменше) постійним точкам присвоювалися певні числові значення. Передбачалося, що видиме термометричні властивість використовуваного в термометрі речовини лінійно пов'язано з температурою t: ,(2.1) де k - коефіцієнт пропорційності; E - термометрична властивість; D - постійна. Беручи для двох постійних точок певні значення температур, можна обчислити постійні k, D і на цій основі побудувати температурну шкалу. При зміні температури коефіцієнт k змінюється, причому він різний для різних термометричні речовин. Тому термометри, побудовані на базі різних термометричні речовин з рівномірною градусної шкалою, давали при температурах, що відрізняються від температур постійних точок, різні свідчення. Останні особливо були помітні при високих (температури кипіння води) і дуже низьких температурах. Термодинамічна шкала тотожна зі шкалою ідеального газу, побудованої на залежності тиску ідеального газу від температури. Закони зміни тиску від температури для реальних газів відхиляються від ідеальних, але поправки на відхилення реальних газів невеликі і можуть бути встановлені з високим ступенем точності. Тому, спостерігаючи за розширенням реальних газів і вводячи поправки, можна оцінити температуру по термодинамічної шкалою. На початку XX століття широко застосовувалися шкали Цельсія і Реомюра, а в наукових роботах - також шкали Кельвіна і воднева. Перерахунки з однієї шкали на іншу створювали великі труднощі і приводили до ряду непорозумінь. Тому в 1933 році було прийнято рішення про введення Міжнародної температурної шкали (МТШ). Досвід застосування МТШ показав необхідність внесення до неї низки уточнень і доповнень, щоб по можливості максимально наблизити її до термодинамічної шкалою. Тому МТШ була переглянута і приведена у відповідність зі

станом знань того часу. У 1960 році було затверджено нове

Цитирования: **0,06%** id: **22**

"Положення про міжнародній практичній температурній шкалі 1948 року. Редакція 1960 р".

Обнаружен Плагиат: **0,72%** http://4ua.co.ua/manufacture/qa2ac7... + 4 ресурсов! id: **23**

2.2. Пристрої для вимірювання температур Температуру вимірюють за допомогою пристроїв, що використовують різні термометричні властивості рідин, газів і твердих тіл. Існують десятки різних пристроїв, що використовуються в промисловості, при наукових дослідженнях, для спеціальних цілей. У таблиці 2.1 наведені найбільш поширені пристрої для вимірювання температури і практичні межі їх застосування. Ми ж розглянемо кілька пристроїв з даної таблиці [23]. Таблиця 2.1 Пристрої вимірювання температури Термометрична властивість Найменування пристрою Межі тривалого застосування, 0С0С Нижня Верхня Теплове розширення Рідинні скляні термометри –190 600 Зміна тиску Манометричні термометри –160 60 Зміна електричного опору Електричні термометри опору. –200 500 Напівпровідникові термометри опору –90 180 Термоелектричні ефекти Термоелектричні термометри (термопари) стандартизовані. –50 1600 Термоелектричні термометри (термопари) спеціальні 1300 2500 Теплове випромінювання Оптичні пірометри. 700 6000 Радіаційні пірометри. 20 3000 Фотоелектричні пірометри. 600 4000

Кольорові пірометри 1400 2800 Рідинні скляні термометри Термометри скляні рідинні застосовуються для вимірювання температур в області від -200 до +750 ° С. Незважаючи на те, що крім скляних рідинних термометрів є ряд інших приладів для вимірювання температур, які відповідають в великій мірі вимогам сучасної техніки контролю технологічних процесів, все ж скляні термометри набули великого поширення як в лабораторній, так і в промисловій практиці внаслідок простоти звернення, досить високою точності вимірювання і низької вартості. Принцип дії скляних рідинних термометрів заснований на тепловому розширенні термометричної рідини, укладеної в термометрі. При цьому, очевидно, показання рідинного термометра залежать не тільки від зміни обсягу термометричної рідини, але також і від зміни обсягу скляного резервуара, в якому знаходиться ця рідина. Таким чином, спостерігається (видиме) зміна обсягу рідини применшити на розмір, відповідно рівний збільшенню обсягу резервуара (і частково капіляра) [24]. Для заповнення рідинних термометрів застосовують ртуть, толуол, етиловий спирт, гас, петролейний ефір, пентан і т.д. З рідинних термометрів ртутний набув найбільшого поширення. Вони мають ряд переваг, завдяки суттєвим перевагам ртуті, що не змочує скла, порівняно легко виходить в хімічно чистому вигляді і при нормальному атмосферному тиску залишається рідкою в широкому інтервалі температур (від -38,87 до + 356,58 ° С) [25] . Термометр ТС -7А Термометр ТС-7А є рідинний скляний термометр (див. Рис. 2.1). Рис. 2.1. Термометр ТС–7А Термометр ТС-7А широко застосовують для вимірювання температури в складських приміщеннях, парниках, теплицях, оранжереях. Сам термометр скляний рідинний НЕ ртутний в оправі з полімерів з вкладеною шкальної пластиною з паперу. Термометр повного занурення. При вимірюванні температури повинен повністю перебувати в вимірюваної середовищі. При перегрів без руйнування термометр може витримати температуру вище верхньої межі за шкалою не більше 20 ° С (табл. 2.2) [26]. Принцип дії термометра заснований на тепловому зміні обсягу термометричної рідини, в залежності від температури вимірюваного середовища. Термометр складається з капілярної трубки з резервуаром, заповненим термометричної рідиною. Капілярна трубка захищена скляною оболонкою, всередину якої вкладена шкала, що служить для відліку вимірюваної температури. Термометр закріплений на підставі. Таблиця 2.2 Метрологічні і технічні характеристики Марка Діапазон вимірювання температури, ºС Ціна ділення шкали, ºС Довжина термометра, мм Діаметр, мм Термом. рідина ТС-7А -10...+60 1 170 11х7,7 (ellipse) толуол 2.3. Пристрої для вимірювання вологості Вологість повітря - показник вмісту води в фізичних тілах або середовищах. Різні культури по-різному ставляться до вологості повітря, і оптимум в різні фази їх розвитку також неоднакові. Однак при оптимальній вологості повинні дотримуватися оптимальні температури. В іншому випадку порушується нормальний ріст і розвиток рослин: при низькій вологості і високій температурі повітря посилюється процес транспірації і може виникнути дефіцит вологості, створюються сприятливі умови для розвитку хвороб і шкідників [27]. При підвищенні температури на 5 ° відносна вологість падає з 76 до 54%. При високій вологості та низькій температурі транспірація знижується, в листі порушується нормальний хід фізіологічних процесів і процес запилення, стимулюється розвиток різної мікрофлори, рослини хворіють білою гниллю, борошнистою росою, бурою плямистістю і ін. На огороджувальних поверхнях (склі, плівці та ін.) Утворюється

Цитирования: **0,01%** id: **24**

"крапель"

- продукт конденсату, яка також сприяє захворюванню рослин і погіршує освітленість приміщення. Вологість ґрунту. Значний вплив на процеси водообміну в рослинах надає вологість ґрунту. У різні періоди життя рослин вона також неоднакова. Найбільша потреба рослин у волозі в період проростання насіння і в розсадний період (до 90-95% НВ), а також в фазу плодоутворення і плодоношення [28]. При високих температурах і освітленості спостерігається потреба рослин в більш високій вологості ґрунту. Так, в спекотні місяці зимово-весняного обороту (з квітня по липень в сьомий світловий зоні), т. Е. У нас, вологість ґрунту необхідно підтримувати в межах: до плодоношення - 75-80%, в період плодоношення - 80-85% . У прохолодний період осінньо-зимового обороту вологість ґрунту можна знижувати до 65% НВ. При нестачі води в ґрунті порушується нормальний ріст і розвиток рослин, при надлишку вологи в ґрунті рослини страждають (і навіть гинуть) через кисневе голодування кореневої системи; розвивається коренева гниль. Тому вологість ґрунту повинна узгоджуватися з умовами температури, освітленості і потребою рослин на певному етапі розвитку. Якщо раніше прилади для вимірювання вологості застосовували тільки з професійною метою, наприклад, в наукових лабораторіях, то сьогодні будь-яка людина, яка піклується про мікроклімат, може використовувати гігрометр. Технічний прогрес йде ще далі, пристрій вдосконалювалося, з'явилися моделі, за допомогою яких можна визначати вміст вологи в твердих тілах [29,30]. Більш детальна класифікація представлена в таблиці 2.3: Таблиця 2.3 Вимірювальні прилади вологості повітря Назва Опис Призначення Вимірювач вологості твердого середовища Використовується для перевірки таких матеріалів як бетон, цемент, деревина, картон, папір і т.ін. Один прилад може бути розрахований на роботу тільки з одним видом матеріалу, наприклад, тільки деревина, або поєднувати в собі відразу кілька можливостей. Замір може проводитися шляхом введення спеціальних гострих зондів в структуру об'єкта або безконтактним способом без порушення зовнішнього вигляду виробу. Контроль якості будівельних матеріалів на виробництві, при закупівлі, зберіганні, перевірці якісних характеристик бетонних конструкцій, готової продукції з деревини, паперу тощо. Термогігрометр Має сенсор, який знімає показники та температури, і вологість повітря. Працює, як і попередній, бесконтактним способом. Вимірювання параметрів мікроклімату в житлових, виробничих приміщеннях, офісах, місцях утримання тварин, теплицях, на складах і т.ін. Вимірювач вологості повітря Оснащений зондом або датчиком, що реагує на скупчення найдрібніших водяних частинок в повітрі. Перевірка умов зберігання книг і предметів мистецтва в музеях, бібліотеках, контроль процесу сушіння продукції на виробництві. Універсальний гігрометр Призначений для вимірювання вологості повітря, твердих середовищ, а також температури і точки роси. Може застосовуватися в різних сферах діяльності: при перевірці умов зберігання паперової продукції, продовольчих товарів, зерна, а також у побуті. Термогігрометр HT-86 (див. Рис. 2.2) забезпечує швидкі і точні вимірювання параметрів навколишнього середовища, включаючи вологість і температуру повітря, температуру точки роси і температуру вологого термометра, має наступні характеристики (табл. 2.4) і особливості [31]. Рис. 2.2. Термогігрометр HT-86 Особливості Термогігрометри HT-86: • Мікропроцесорний прилад; • Висока швидкодія, висока точність; • Одночасно виводиться на екран два параметра - вологість і температура; • Функції утримання даних, утримання максимального і мінімального значення; • Автоматичне вимикання, підсвічування. Таблиця 2.4 Технічні характеристики Вимірювання вологості Діапазон: 0~100% відносної вологості Точність: ±3% RH (при 25°C, 20% ~80 % RH), ±3.5% RH (при інших умовах) Роздільна здатність: 0.01% RH Вимірювання температури Діапазон: -20°C ~ 100°C (-22°F~199°F) Точність: ±0.5 °C/0.9 °F (при 25°C) ±0.8 °C ±1.5 °F (в другому діапазоні) Разрешение: 0.01 °C/0.01 °F Температура точки роси -20°C ~ 100°C (-22 °F ~ 199 °F) Температура вологого термометра (0~80°C /32-176 °F) Відгук 10 сдля вимірювання вологості повітря Живлення 9В батарейка NEDA 1604, IEC 6F22 Умови експлуатації 32 °F до 104 °F (0 °C до 40°C); 80% RH Умови зберігання 32°F до 104°F(0 °C до 40°C); 80% RH Габарити 175 мм*58 мм*35мм Вага 143 г Датчик температури і вологості AM2321 В даний датчик включає в себе датчик вологості і датчик температури (див. Рис. 2.3-2.4). Діапазон робочих напруг від 2.6 до 5.5V. Розміри датчика 11.3x7.8 висота 4 мм. Крок висновків 1.27мм. Рис. 2.3. Датчик вологості AM2321 Рис. 2.4. Відносна вологість максимальної похибки AM2321 Частота шини до 100 Кб / с. Діапазон вимірюваної температури від -40 до +80. Може працювати в двох режимах передачі даних I2C і 1-Wire. Для того що б перемкнути в режим I2C потрібно подати харчування і на шині SCL мінімум на 500мс встановити логічну одиницю. Для перемикання в 1-Wire режим після подачі живлення на шині SCL мінімум на 500мс встановлюємо логічний нуль. Адреса датчика 0xB8 (10111000 r / w) [32]. Температура і вологість комбінованого датчика AM2321. Цифровий датчик температури і вологості, характеристики якого наведені в таблиці 2.5, являє собою цифровий відкалібрований вихідний сигнал. Датчик складається з ємнісного елемента вологості і інтегрованого приладу для вимірювання температури високої точності,

з'єднаний з мікропроцесором високої продуктивності (табл. 2.5). Таблиця 2.5 AM2321 відносна вологість Параметр Стан Мін Тип Макс Од.вим Роздільна здатність 0.1 %RH Асортимент 0 99.9 %RH Точність 25℃ ±3 %RH Стабільність ±0.1 %RH Взаємозамінність Повністю взаємозамінні Час відгуку 1/e(63%) 5 S Повільність ±0.3 %RH Дрейф Типові значення 0.5 %RH/рік Зв'язки з використанням шини, є два режими зв'язку стандартний інтерфейс однієї шини I2C. Ультра-малий розмір, низьке енергоспоживання, передачі сигналу на відстані до 20 метрів, що робить всі види додатків і навіть найвибагливіші додатки кращим вибором. Користувач може безпосередньо використовувати зв'язок з I2 C шини без додаткової проводки, яка проста у використанні [33]. Два режиму зв'язку використовуються як вологість, температура з CRC контрольної сумою температурної компенсації на виході. Користувачам не потрібно обчислювати вторинний цифровий вихід, і немає необхідності в температурної компенсації вологості. Два режиму зв'язку можуть вільно перемикатися, користувач може вільно вибирати, простий у використанні або широкий спектр застосування. Таблиця 2.6 Таблиця продуктивності Параметр Стан Мін Тип Макс Од.вим Роздільна здатність 0,1 Точність 16 bit Стабільність ±0,5 Взаємозамінність -40 80 Час відгуку ±0.2 Повільність Дрейф =0,1 S Електричні характеристики: високе споживання енергії, низький рівень, вхідної та вихідної напруги, в залежності від джерела живлення. У таблиці 2.7 докладно описуються електричні характеристики AM2321, а на рисунку 2.5 - максимальна похибка датчика температури. Таблиця 2.7 AM2321 DC Характеристики Параметр стан мін тип макс блок Напруга живлення 2.6 5 5.5 v Споживана потужність Сплячий режим, Міра, В средньому 8 10 500 200 μA Вихідна напруга низького рівня CC 0 250 MV Висока вихідна напруга R p 25 kΩ 90% 100% VDD Вхідна напруга низького рівня Зниження 0 30% VDD Висока вхідна напруга Підйом 70% 100% VDD R p u 6 VDD = 5V VIN = VSS 30 45 60 kΩ Вихідний струм Open Tri-State (Off) 10 8 20 MA μA Період вибірки 2.0 s Рис. 2.5. Максимальна похибка датчика температури Вхід послідовного тактового сигналу (SCL). SCL використовується для передачі інформації про лінії зв'язку вибору і синхронізації I2S (табл.2.8). Коли живлення утримується на низькому рівні, в той час як SCL залишається низьким навіть 500 мс, це означає, що користувач вибирає один режим передачі даних по шині (рис. 2.6), в іншому випадку - IC-зв'язок. Після вибору режиму живлення, режим зв'язку з датчиком залишається незмінним; Якщо ви хочете змінити зв'язок, увімкніть живлення, натисніть на операцію вибору, щоб встановити зв'язок, при виборі IC-зв'язку між мікропроцесором і AM2321, SCL для синхронної зв'язку. Таблиця 2.8 Призначення контактів Pin Name Description 1 VDD Джерело струму (2,6 -5.5V) 2 SDA Послідовні дані, двонаправлений порт v3 GND Земля 4 SCL Послідовний вхід синхронізації порту (один бас на землю) Рис. 2.6. AM2321 шина I2C. Типова схема застосування та тимчасова діаграма Якщо потрібно отримати результати, використовуючи датчик, необхідно дотримуватися характеристик зазначених в табліце2.9 для даного датчика. Таблиця 2.9 Характеристики інтерфейсу датчика C Symbol Parameters Standard C mode Unit min Max SCL clock frequency 100 kHz T w (SCLL) SCL Clock low time 4.7 μs T w (SCLH) SCL Clock high time 4.0 T s u (SDA) SDA Settling time 250 T h (SDA) SDA Data hold time [1] T r (SDA) SDA and SCL Rise time 1000 ns T r (SCL) T f (SDA) SDA and SCL Fall Time 300 T f (SCL) T h (STA) Start condition hold time 4.0 μs T s u (STA) Repeated Start condition setup time 4.7 T s u (STO) Stop condition setup time 4.0 μs T w (STO:STA) Stop to Start condition time (bus idle) 4.7 μs C b Capacitive load for each bus 400 P F Область даних включає в себе необхідні датчиком відомості про те, яку інформацію або які дії необхідно виконати. Ця інформація може являти собою дані (такі як: температура, вологість, інформація про сенсорному пристрої, дані для запису користувачем і т. Д.), Адреса посилання. Наприклад, код функції 03 повідомляє хосту через значення регістра, що повертаються датчиком дані (включаючи початкову адресу регістру, що підлягає читання, і регістр довжини зчитування) містять довжину даних і вміст реєстрового регістра даних. Датчик використовує власний комунікаційний протокол шини IC-Mod, який використовується головною командою зв'язку (код функції 03), яка може зчитувати його регістр даних, таблицю регістра даних (табл. 2.10). У регістрі даних датчика зберігаються значення температури і вологості, а також відповідна інформація про сенсорному пристрої та інших відповідних сигналах. Кожен регістр даних являє собою один байт (8 біт) двійкових даних. Коли дані зчитуються до 10 регістрів датчика (більше довжини зчитування) датчик повертає відповідний код помилки. Таблиця 2.10 Регістр даних Register information Address Register information Address Register information Address Register informatio n Address High humidity 0x00 Model High 0x08 Users register a high 0x10 Retention 0x18 Low humidity 0x01 Model Low 0x09 Users register a low 0x11 Retention 0x19 High temperature 0x02 The version number 0x0A Users register 2 high 0x12 Retention 0x1A Low temperature 0x03 Device ID (24-31) Bit 0x0B Users register 2 low 0x13 Retention 0x1B Retention 0x04 Device ID (24-31) Bit 0x0C Retention 0x14 Retention 0x1C Retention 0x05 Device ID (24-31) Bit 0x0D Retention 0x15 Retention 0x1D Retention 0x06 Device ID (24-31) Bit 0x0E Retention 0x16 Retention 0x1E Retention 0x07 Status Register 0x0F Retention 0x17 Retention 0x1F Формат виведення температури Температурне дозвіл є 16-

бітовим (табл. 2.11), старший біт (Біт15) дорівнює 1 і вказує на негативну температуру, якщо старший біт (Біт15) дорівнює 0, вказує на позитивну температуру; температура на додаток до найбільш значимого біту (Bit14 ~ Біт0) вказує на значення рядка датчика температури. Таблиця 2.11 Регістр стану Status register bits Bit7 Bit6 Bit5 Bit4 Bit3 Bit2 Bit1 Bit0 Features Retention Retention Retention Retention Retention Retention Retention Retention Регістр стану, Біт7-Біт0 біт, тимчасово захищені. 2.4. Висновки до розділу Проаналізувавши способи вимірювання температури видно, що рідинні скляні термометри є надійними з високою точністю вимірами. Однак через великий діапазон вимірювання температури, для розроблювального пристрою більш доцільно використовувати датчик DS18B20, в якому за допомогою додаткових обчислень можна досягти такої ж точності, як і в рідинних скляних термометрах. Так само інтеграція датчика DS18S20 в пристрій є легшою, ніж інтеграція рідинних скляних термометрів. Проаналізувавши існуючі датчики вологості видно, що термогігрометр є надійним з високою точністю, але датчик AM2321 легше інтегрувати в пристрій. РОЗДІЛ III
РОЗРОБКА ПРИНЦИПОВОЇ СХЕМИ ПРИСТРОЮ 3.1. Сімейство AVR Фірма Atmel зіграла в електронній промисловості роль, в деякому роді порівнянну з роллю Apple у виробництві ПК: вона не перебуває на перших ролях, але завжди відрізнялася передовими інноваціями. Зараз на ринку 8-розрядних МК вона має близько 6%. Фірму заснував спочатку 1980-х Джордж Перлегос, відомий тим, що керував в Intel напрямком незалежній пам'яті: при його безпосередній участі були розроблені перші типи електрично стираних кристалів EEPROM. У 80-ті роки Atmel спробувала зайняти провідне місце на цьому швидко зростаючому ринку, і залишається до сих пір одним з провідних виробників flash- і EEPROM-чіпів для установки в схеми різних пристроїв (наприклад, нерідко можна зустріти Flash BIOS на чіпах Atmel) [34]. В кінці 1980-х Atmel в числі інших виявилася втягнута в патентні розбірки з енергонезалежною пам'яттю, ініційовані Intel, з яких вийшла навіть з деяким прибутком, опинившись власником ліцензії на виробництво x51-сумісних МК, в якою області і понині є однією з провідних. У 1993 році Atmel стала ініціатором перевороту в галузі, почавши виробництво МК AT89C51 з вбудованою flash-пам'яттю програм, що виявилося значно зручніше одноразових

Цитирования: **0,01%** id: **25**

"пропалює"

кристалів або УФ-стираних, що вимагають тривалої

Цитирования: **0,01%** id: **26**

"ручний"

процедури очищення. У 1995 році два норвезьких студента з м Тронхейма, Альф Боген і Вегард Воллей, висунули ідею 8-розрядного RISC-ядра, яку запропонували керівництву Atmel. Ядро AVR названо за іменами розробників (Alf - Vergard - RISC). У AVR-ядрі типові інструкції виконуються за один такт, причому в забезпечення цього є елементарний конвеєр з одночасним виконанням інструкції і вибіркою наступної (тому команди розгалуження виконуються за два такту - конвеєр вважає, що умова розгалуження не виконається). Первісне ядро складалося всього з 32 тисяч транзисторів. У роботі над ядром взяла участь згадувана фірма IAR System, розробник С-компіляторів. Можливо, тому головною особливістю AVR став рішучий розрив з характерними для дотеперішніх архітектур з інструкціями, що виконувалися через спеціальний регістр-акумулятор. Більшість команд може безпосередньо оперувати з регістрами загального призначення, не потребуючи в зверненні до ОЗУ. Тому структура асемблерних програм для AVR стала підозріло нагадувати програми на мові високого рівня, де оператори працюють не з осередками пам'яті і регістрами, а з абстрактними змінними і константами. Мало того, в ряді молодших моделей не можна навіть безпосередньо працювати зі стеком - для програми він при такій кількості регістрів-змінних виявляється просто непотрібним, і використовується лише апаратно при виклику підпрограм [35]. Зробити крок до переходу від асемблера на С при такій архітектурі значно простіше: фактично залишалося лише упакувати те, що знаменитий програміст Дейкстра назвав

Цитирования: **0,03%** id: **27**

"локшиною умовних і безумовних переходів",

в звичні цикли з передумовою / післяумовою або оператори вибору. Саме з цих причин архітектура AVR вважається найбільш пристосованою до програмування на С. Причому різні способи адресації, що мають таке важливе значення в архітектурі x51, тут розкидані по зовсім різним командам, і програміст взагалі може не вивчати відповідний розділ інструкції. Кількість команд в архітектурі AVR досить велике, 120-130 штук, але протиріччя з RISC-концепцією тут немає - 30-40% команд є лише псевдоніми, введені для зручності програміста. Причому інструкцій ділення і множення, в повній відповідності з RISC-

концепцією, канонічний AVR не передбачає: лише пізніше для більш

Цитирования: **0,01%** id: **28**

"просунутих"

моделей з'явилася можливість апаратного множення. Іншим великим плюсом AVR став вперше введений послідовний інтерфейс занесення програми в пам'ять кристала, завдяки чому МК стало можна програмувати прямо в готовій схемі, без будь-яких спеціальних програматорів: досить програмного рішення на ПК, що з'єднується через LPT зі схемою п'ятьма проводами. Через подібного переваги архітектура AVR швидко стала популярною. Пізніше таким інтерфейсом були змушені обзавестися і інші великі сімейства МК. До недоліків AVR можна частково віднести те, що електрично висновки портів за рівнями сигналів є КМОП-сумісними: це підвищує стійкість (поріг розрізнення низького-високого рівня лежить в районі середини харчування, а не ближче до нуля, як у x51), але викликає ряд проблем сумісності з різними стандартними інтерфейсами. Не дуже логічно і побудова деяких команд в області операцій з бітами. Але в цілому архітектура AVR, безсумнівно, більш прогресивна і якби в цій галузі моди змінювалися так само швидко, як це відбувається в області ПК, то AVR мала б усі шанси на першість. 3.2. Архітектура ATmega16 В роботі використовується ATmega16 - високопродуктивний 8-розрядний AVR-мікроконтролер з низьким енергоспоживанням (рис. 3.1). Мікроконтролер виготовлений по КМОП-технології, яка в поєднанні з RICS архітектурою дозволяє досягти найкращого співвідношення показників швидкодію / енергоспоживання. Основні переваги Atmega16: • мікроконтролер здатний обробляти 131 потужну інструкцію, більшість з яких виконуються за один машинний цикл; • 32 8-розрядних регістра загального призначення і регістри управління вбудованою периферією; • повністю статична робота; • продуктивність до 16 мільйонів операцій в секунду при тактовій частоті 16 МГц; • вбудоване множити пристрій виконує множення за 2 машинних циклу; • 16 Кбайт внутрісистемного програмованої флеш-пам'яті; • пам'ять даних (ОЗУ) - 1 Кбайт; • пам'ять даних (EEPROM) - 512 байт, розрахована на 100000 циклів читання / запису; • 32 програмовані лінії введення / виводу (рис. 3.2). Рис. 3.1. Блок-схема ATmega16 Рис. 3.2. Розташування виходів ATmega16 3.3. Принцип роботи електронного пристрою Відлік часу проводиться всередині мікроконтролера за допомогою асинхронного таймера - Т2. Програмно сформований будильник подає сигнал кожні дві години. Відображення часу і подача інших візуальних сигналів проводиться за допомогою світлодіодної матриці зібраної з трьох блоків ТС15-11. Керуючий сигнал формується в мікроконтролері і подається на блок світлодіодних матриць через сигнальний шлейф і двійково-десятковий дешифратор на мікросхемі 74HC595. У звичайному режимі на екрані відображається час в форматі годинник-хвилини. З інтервалом в одну хвилину відбувається запуск рядка, що біжить. Біжучим рядком відображається день тижня, дата, рік, температура, і вологість в оранжереї. Та, що біжить рядок настроюється, тобто можна включити / вимкнути відображення будь-якого з елементів. При виключенні відображення всіх елементів рядка, що біжить, вона не запускається зовсім, і годинник постійно відображають лише час. Харчування будильника можливо різними способами: блок живлення на 5В, батареї або акумулятори, або іоністра. Також є можливість установки резервного джерела живлення, для підтримки працездатності системи при пропажі живлення від основного джерела. Для контролю температури і вологості в приміщенні використовуються датчики AM2321. Може працювати в двох режимах передачі даних I2C і 1-Wire. У нашому випадку використовується режим 1-wire. В даному режимі передача даних здійснюється по черзі - спочатку мікроконтролер надсилає запит датчику, а потім датчик відправляє відповідь. У разі використання декількох датчиків, передача даних на мікроконтролер виконується по черзі. Завдяки наявності унікального 64-бітного послідовного ідентифікаційного коду, розташованого у внутрішній ROM-пам'яті і призначеної для багатоточкових систем, є можливість визначити з якого конкретно датчика приходить інформація. Знаючи розташування датчика, ми можемо більш точно контролювати температуру в різних частинах підконтрольного об'єкта, що є важливою складовою при роботі з деякими рослинами. Виходячи з даної інформації, побудована структурно-функціональна схема пристрою (див. Рис. 3.3). Рис. 3.3. Синтез структурно-функціональної схеми пристрою Призначення кнопок будильника: 1. Кнопка 1 - умовна назва

Цитирования: **0,01%** id: **29**

"ОК".

2. Кнопка 2 - умовна назва

Цитирования: **0,01%** id: **30**

"КРОК".

Управління будильником здійснюється двома кнопками. Їх кількість можна було довести до

8шт, не додаючи жодного компонента, крім самих кнопок, але для зручності

Цитирования: **0,01%** id: **31**

"скористалися"

всього двома. Кнопки умовно названі

Цитирования: **0,01%** id: **32**

"ОК"

і

Цитирования: **0,01%** id: **33**

"КРОК".

Кнопкою

Цитирования: **0,01%** id: **34**

"КРОК"

як правило відбувається перехід до наступного пункту меню, а кнопкою

Цитирования: **0,01%** id: **35**

"ОК"

зміна параметрів поточного меню. Сигнал спрацював будильника також вимикається кнопками

Цитирования: **0,01%** id: **36**

"ОК"

або

Цитирования: **0,01%** id: **37**

"КРОК".

Натискання будь-якої кнопки під час сигналу будильника відключає його.

Цитирования: **0,01%** id: **38**

"Серцем"

будильника обрана AtMega16A, через її доступності і дешевизни. Схема була спрощена, тому все що можна, було покладено на контролер. В результаті вдалося обійтися всього двома мікросхемами, контролером і регістром з потужними виходами TPIC6B595. У разі не доступності TPIC6B595, його можна замінити на 74HC595 + ULN2803. Відлік часу проводиться за допомогою асинхронного таймера - T2. Хід годин триває і при пропажі живлення. В цей час велика частина схеми знеструмлена, і тільки контролер отримує харчування від батарейки, акумулятора, або від іоністори. Струм споживання будильника в черговому режимі становить 15 мка. При харчуванні від іоністори на 1Ф, будильник пропрацював чотири доби. Цього цілком достатньо для підтримки ходу під час перебоїв харчування. Якщо застосувати батарейку CR2032, заряду повинно вистачити на 1,5 року. Наявність напруги контролер приймає через висновок PB.3. Напруга живлення через дільник R2-R3 подається на висновок PB.3, і в нормальному стані дорівнює приблизно 1,5 В. Якщо зовнішня напруга впаде нижче 4,1 вольта, то напруга на виводі PB.3 стане менше 1,23 вольта, при цьому згенерує переривання від компаратора, і в обробнику цієї переривання вимикаються всі

Цитирования: **0,01%** id: **39**

"зайві"

вузли контролера, і сам контролер йде в режим очікування. В цьому режимі продовжує працювати тільки відлічує час таймер T2. При появі зовнішнього живлення, напруга на PB.3 знову підніметься вище 1,23 В, контролер

Цитирования: **0,01%** id: **40**

"побачивши"

це, переведе всі вузли в робочий стан. Якщо замість іоністори, буде використовуватися батарейка CR2032, то її потрібно підключити через діод (переважно діод Шоткі). Анод діода підключається до + батарейки, а катод до катода VD1. 9 будильників розділені на 3 одноразових і 6 багаторазових. При включенні будильників 1-3, вони спрацьовують тільки один раз. Для того щоб вони спрацювали ще раз, їх потрібно повторно включати вручну. А будильники 4-9 багаторазові, тобто вони будуть спрацьовувати щодня, у встановлений час. Крім того ці будильники можна налаштувати на роботу тільки в певні дні тижня. Це зручно, наприклад, якщо не хочете, щоб будильник перевірив температуру і вологість тільки у вихідні. Або, наприклад, Вам потрібно перевіряти температуру і вологість в будні

дні в 7-00, а в четвер о 8-00, будильник не потрібен. Тоді налаштовуємо один багаторазовий на 7-00 в понеділок-середу і п'ятницю, а другий на 8-00 в четвер. Крім того всі будильники мають настройку тривалості сигналу, і якщо мало сигналу протягом 1 хвилини, то можна збільшити його на час від 1 до 15 хв. Корекція ходу проводиться один раз на добу, в 00-00. Якщо годинник поспішають, наприклад, на 5 сек на добу, то в 00-00-00 час встановиться в 23-59-55, якщо ж годинник відстає на 5 сек, то в 00-00-00 час встановиться в 00-00- 05. Крок корекції - 0,1 сек. Максимальна корекція - 59,9 сек / добу. З справним кварцом більше навряд чи знадобитися. Корекція здійснюється і в черговому режимі при живленні від батареї. 3.4. Розробка і виготовлення друкованої плати Для розробки друкованої плати було вибрано додаток DipTrace версії 2.3.1.0. DipTrace - це система автоматизованого проектування друкованих плат (PCB). У пакет включено чотири програми: • Schematic - розробка принципових схем; • DipTrace - створення плат, ручна і автоматична трасування; • ComEdit - редактор корпусів; • SchemEdit - редактор компонентів. Програма володіє найбільш зручним серед САПР для проектування друкованих плат призначеним для користувача інтерфейсом. Вся робота супроводжується підсвічуванням редагованих і залежних від них елементів, що дозволяють наочно оцінювати ситуацію. Логічна структура принципової схеми або плати формується відразу при побудові і зміна одного елемента відбивається на елементах, залежних від нього [37]. У редакторах корпусів і компонентів застосована пошарова система малювання, можливе створення компонентів і корпусів на основі готових шаблонів. Допускається створення корпусів з перемінними параметрами (кількість ніжок, відстань між ними), які потім можна буде змінювати прямо в PCB. Схемотехніка підтримує створення багатолистових принципових схем (див. Рис. 3.4), перехід з одного аркуша на інший здійснюється за допомогою загальних шин, які з'єднані міжсторінкових переходами. Створена принципова схема перетвориться в друковану плату вибором відповідного пункту меню або натисканням комбінації клавіш в схемотехніці. Перекладена плата являє собою корпусу компонентів з логічними зв'язками 38,39]. Рис 3.4. Schematic. Принципова схема пристрою При ручному трасуванні виконується підсвічування всіх висновків мережі і автоматичне видалення вже розведених зв'язків, при редагуванні розведених трас можливе переміщення фрагментів провідників з дотриманням кутів 90 і 45 градусів або вільне, зміна ширини, переклад з одного шару в інший і т.п. Автотрасування здійснюється за допомогою вбудованого сіткового оптимізаційного трасувальника - SimpleRouter. Закінчений варіант трасування плати представлений на рисунках 3.5 і 3.6. Рис. 3.5. PCBLayout. Верхня сторона плати розроблювального пристрою Рис. 3.6. PCBLayout. Нижня сторона плати розроблювального пристрою Висновок принципових схем і друкованих плат на принтер виконується в довільному масштабі, для друкованої плати вказуються об'єкти, принципової схеми - листи. Можлива підстановка рамки і штампа (штампів), для рамки може задаватися розмітка зон. Створення штампів здійснюється прямо в програмі через спеціальний редактор, який працює за принципом електронних таблиць. Спеціальна функція експорту обведення в DXF дозволяє виробляти плати методом фрезерування (необхідний конвертер DXF в G-код). До складу DipTrace включено велику кількість бібліотек корпусів і компонентів. Стандартні бібліотеки містять більше 40 тис. Компонентів різних фірм-виробників. Створене пристрій виглядає так, як показано на рисунку 3.7 Рис. 3.7. Плата розроблювального пристрою 3.5. Розробка програмного забезпечення Для написання програми роботи мікроконтролера була вибрана мова програмування С і середовище розробки MikroC PRO [40,41]. MikroC PRO for AVR - потужний інструмент розробки програм для AVR мікроконтролерів. Він сконструйований, щоб забезпечити програмісту найменш трудомісткі рішення по створенню додатків для вбудованих систем. MikroC PRO забезпечує успішне поєднання розвиненої середовища програмування (IDE), відповідного ANSI компілятора, широкого набору бібліотек для апаратних засобів, всебічної документації та великої кількості, готових до вживання прикладів [42,43]. MikroC дозволяє швидко розробити і впровадити складні програми: Текст програми вводиться за допомогою вбудованого редактора вихідного коду (за допомогою в коді і параметрах, контекстної підсвічуванням, автокорреціей, кодовими шаблонами і т.п.); Використання додаються бібліотек mikroC істотно підвищує швидкість розробки: збір, зберігання, індикація, перетворення і обмін даними. Підтримуються мікроконтролери фірми Atmel сімейства AVR [44,45]; Провідник по коду (Code Explorer) дозволяє оперативно контролювати структуру програми, змінні і функції проекту; Після обробки створюються прокоментував і читабельний файл на асемблері і стандартний HEX файл, сумісний з усіма программаторами; Вбудований відладчик дозволяє перевіряти хід і логіку виконання програми; Після компіляції надається повна статистика використання пам'яті, асемблерний лістинг, дерево викликів функцій; Забезпечено достатню кількість прикладів, які можна розширювати і використовувати як складові частини проектів, що розробляються. Якщо потрібно, можна копіювати їх в проекти цілком, для того вони і включені до складу компілятора [46]. В даний час mikroC PRO для AVR включає наступні бібліотеки і приклади:

• ADC бібліотека; • CANSPI бібліотека; • Compact Flash бібліотека; • EEPROM бібліотека; • Флеш-пам'ять бібліотека; • Графічний ЖК-бібліотека; • Клавіатура бібліотека; • РК-бібліотека; • Манчестер кодекс бібліотека; • Multi Media Card бібліотека; • OneWire бібліотека; • Port Expander бібліотека; • PS / 2 Бібліотеки; • PWM 16 розрядна бібліотека; • RS-485 бібліотека; • Програмне забезпечення I2C бібліотека; • Програмне забезпечення SPI бібліотека; • Software UART бібліотека; • Sound Library; • SPI бібліотека; • SPI Ethernet бібліотека; • SPI Графічний ЖК-бібліотека; • SPI LCD бібліотека; • SPI LCD8 бібліотека; • SPI T6963C Графічний ЖК-бібліотека; • T6963C Графічний ЖК-бібліотека; • TWI бібліотека; • UART і USB бібліотеки; 3.6. Алгоритм роботи пристрою Після включення пристрою відбувається запуск таймера, який кожні півсекунди перевіряє, підключено або відключено пристрій, а також виконується збільшення лічильників часу. Після цього порівнюється поточний час з часом будильника. У разі, якщо воно збігається, подається звуковий сигнал. Далі слід натиснути будь-яку кнопку. Якщо час не збігається з часом будильника, кожні 2 години виконується перевірка температури і вологості. Якщо була натиснута одна з кнопок (S1 або S2), то активується підпрограма налаштування пристрою. Після отримання даних з датчика виконується висновок даних на світлодіодні матриці. Блок-схема алгоритму роботи пристрою приведена на рисунку 3.8. Рис. 3.8. Алгоритм роботи пристрою 3.7. Висновки до розділу У додатку Diptrace була побудована схема друкованої плати, на базі якої було сконструйовано пристрій. Також в розділі описана середовище розробки, в якій розроблялося програмне додаток для пристрою. Представлений алгоритм роботи пристрою у вигляді блок-схеми. ВИСНОВКИ У магістерській роботі було розглянуто розробку системи контролю температури і вологості повітря на основі математичного дослідження мікроклімату. Перший розділ містить опис існуючого вигляду пристрою, використовуваного для контролю вологості і температури в оранжереї на основі плати Arduino Mega. Так само в першому розділі описані необхідні умови для вирощування рослин і представлена математична модель оранжереї при управлінні температурою і вологістю повітря, яка розроблена для застосування в освітніх цілях з метою продемонструвати фізичні явища, що відбуваються в оранжереї і принципи управління мікрокліматом. У другому розділі проаналізувавши способи вимірювання температури було визначено, що рідинні скляні термометри є надійними з високою точністю вимірами. Однак через великого діапазону вимірювання температури, для розроблювального пристрою більш доцільно використовувати датчик DS18B20, в якому за допомогою додаткових обчислень можна досягти такої ж точності, як і в рідинних скляних термометрах. Так само інтеграція датчика DS18S20 в пристрій є легшою ніж інтеграція рідинних скляних термометрів. Проаналізувавши існуючі датчики вологості видно, що термогігрометр є надійним з високою точністю, але датчик AM2321 легше інтегрувати в пристрій. Третій розділ присвячений створенню пристрою. У додатку Diptrace була побудована схема друкованої плати, на базі якої було сконструйовано пристрій. Так само в розділі описана середовище розробки, в якій розроблялося програмне додаток для пристрою. Був представлений алгоритм роботи пристрою у вигляді блок-схеми. СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ Статистична інформація. Сільське господарство. [Електронний ресурс]. Режим доступу: URL: http://www.vn.ukrstat.gov.ua/index.php/statisticalinformation/ 228/2013--1995-2010.html (дата звернення: 13.07.20). Гросуляк П.І. Структура та моделі роботи системи управління мікрокліматом мінітеплиці / Гросуляк П.І. , Цьома Б.І. // Тези доповідей Науково-практичної конференція молодих вчених і студентів

Цитирования: **0,03%** id: **41**

«Інтелектуальні комп’ютерні системи та мережі».

– Тернопіль: ЗУНУ. – 2020. – С. 37. Гросуляк П.І. Використання сучасних інформаційних технологій в сільському господарстві / Гросуляк П.І. , Цьома Б.І. // Тези доповідей Науково-практичної конференція молодих вчених і студентів

Цитирования: **0,03%** id: **42**

«Інтелектуальні комп’ютерні системи та мережі».

– Тернопіль: ЗУНУ. – 2020. – С. 38. Мартыненко И.И. Автоматика и автоматизация производственных процессов / Мартыненко И.И., Головинский Б.Л., Проценко Р.Д., Резниченко Т.Ф. // М.: Агропромиздат - 1995. - 335 с. Кэмп П. Компьютерное управление микроклиматом в теплицах / Кэмп П, Тиммерман Г. // Центр инноваций и практического обучения в Эдде - 1997. – 193с. Технологічні процеси галузей промисловості: Навч. посібник / Д.М. Колотило, А.Т. Соколовський, С.В. Гарбуз; За наук. ред. Д.М. Колотила, А.Т.Соколовського. — К.: КНЕУ, 2003. — 380 с. International Standart ISO 9126-1. Software engineering – Product quality – Part 1: Quality. – 2001. P. 32. Kharchenko V. Green Computing and Communications in Critical Application Domains: Challenges and Solutions/ Kharchenko V.,Sklyar V., Gorbenko A., Philips C// Proceedings of International Conference on Digital Tenchnologies, May, 29-31, 2013, Zilina, Slovakia, 2013, P.24-29. Условия, необходимые для роста и развития

растений. [Электронный ресурс] – URL: http://ogoroniks.ru/vegetable/plant_care/7-usloviya-neobhodimye-dlya-rosta-i-razvitiya-rasteniy.html. Маркелова И.В. Лимоны. Уход и выращивание ООО

Цитирования: **0,01%** id: **43**

"Авеонт",

2006. – 96 с. Илья Мельников. Теплица, парник, зимний сад. ЛитРес, 2009. – 110 с. Сааков С.Г. Оранжерейные и комнатные растения и уход за ними, 1985. – 623 с. С. Раделов. Все о теплицах и парниках на садовом участке. Вильнюс, Bestiary. 2013 Людмила Шульгина. Теплицы и парники. Строительство и рекомендации по выращиванию овощей, цветов, грибов. Книжный клуб

Цитирования: **0,02%** id: **44**

"Клуб семейного досуга".

Харьков, Книжный клуб

Цитирования: **0,02%** id: **45**

"Клуб семейного досуга".

Белгород, 2012. Алан Титчмарш. Умная теплица. Петроглиф, 2013. Н. Новикова. Архитектура теплиц и оранжерей. Архитектура-С, 2006. Галина Кизима. Урожайная теплица. Что когда сажать. Как удобрять и защищать. АСТ, 2013. Анализ системы управления микроклиматом оранжереи компании

Цитирования: **0,01%** id: **46**

"Альфа"

на плате Arduino Mega. [Электронный ресурс] – URL: http://meandr.org/archives/25973. Исследования математической модели микроклимата оранжереи. [Электронный ресурс] – URL: http://cyberleninka.ru/article/n/matematicheskaya-model-mikroklimata-teplitsy. Токманов Н.М. Математическая модель системы управления микроклиматом ангарных теплиц Н.М., В.С. Грудинин. Петроглиф, 2008. Буклагина Г.В. Автоматизация теплиц различных поколений/ Инженерно-техническое обеспечение АПК. Реферативный журнал. 2005. –№ 4. – 922 с. Крупнов Б.А., Шарафадинов Н.С. Руководство по проектированию систем отопления, вентиляции и кондиционирования воздуха. Вена. 2008 – 220 с.. Рысс А.А Автоматическое управление температурным режимом в теплицах. - М.: Агропромиздат, 1986 - 126 с. Токманов Н.М. Особенности управления микроклиматом в ангарных теплицах / Гавриш №3, 2008 г., с. 28-32. Гудвин Г.К. Проектирование систем управления. / Г.К. Гудвин, С.Ф. Гребе, М.Э. Салъгадо.- Москва.: БИНОМ.2004. Исследование динамических характеристик системы управления микроклиматом оранжереи. [Электронный ресурс] – URL: http://dspace.mnau.edu.ua:8080/jspui/handle/123456789/313. Дащенко А.Ф. и др. MATLAB в инженерных и научных расчетах. Одесса, 2003. - 214 с. Дьяконов В. П. MatLab 6.5 SP1 (7.0). Simulink 5/6. Основы применения. СОЛОН-пресс, 2005. – 800 с. Егоренков Д.Л., Фрадков А.Л., Харламов В.Ю.

Обнаружен Плагиат: **0,06%** https://rutracker.org/forum/viewtopic.... id: **47**

Основы математического моделирования. Построение и анализ моделей с примерами на языке

MATLAB. Учебное пособие. – СПб.: БГТУ, 1994. – 190 с. Понятие о температуре и температурных шкалах [Электронный ресурс] – URL: http://www.physicedu.ru/phy-540.html. Устройства для измерения температур [Электронный ресурс] – URL: http://100pudov.com.ua/subject/80/34560/. Жидкостные стеклянные термометры [Электронный ресурс] – URL: http://studbooks.net/1936578/matematika_himiya_fizika/termometry_steklyannye_zhidkostnye. Г.Г.Раннев, А.П.Тарасенко

Цитирования: **0,02%** id: **48**

"Методы и средства измерений"

2004г Термометр ТС -7А [Электронный ресурс] – URL: http://www.steklopribor.com/termometri-steklyannie-ts-7-m1. Влажность [Электронный ресурс] – URL: https://ru.wikipedia.org/wiki/%D0%92%D0%BB%D0%B0%D0%B6%D0%BD%D0%BE%D1%81%D1%82%D1%
Влажность в почве [Электронный ресурс] – URL: http://www.activestudy.info/rezhim-vlazhnosti-v-teplicax/. Устройства для измерения влажности [Электронный ресурс] – URL: http://www.vseinstrumenti.ru/instrument/izmeritelnyj/gigrometry/articles/vidy_ustrojstvo/. Афонский А.А., Дьяконов В.П. Измерительные приборы и массовые электронные измерения. Солон-Пресс, 2007. Термогигрометр НТ-86 [Электронный ресурс] – URL:

http://simvolt.ua/termoggrometr-ht-86.html. Датчик температуры и влажности AM2321 [Электронный ресурс] – http://radiokot.ru/forum/viewtopic.php?f=25&t=117421. Температура и влажность комбинированного датчика AM2321 AM2321 [Электронный ресурс] –http://akizukidenshi.com/download/ds/aosong/AM2321_e.pdf. Семейство AVR [Электронный ресурс] –http://www.alldatasheet.com/view.jsp?Searchword=Atmega16&gclid=CMXMgIK36NICFcVUGAodHPsMjA. Гребнев В.В. Микроконтроллеры семейства AVR фирмы Atmel. - М.: Радиософт, 2002. Datasheet: AT90S8515, ATmega16, ATmega8, PCA9544/PCA-9544A, AVR315. Разработка и изготовление печатной платы [Электронный ресурс] – DipTrace 2.1.0.7 XRUST.ru [Электронный ресурс] – URL: http://xrust.ru/soft/free_soft/93371-diptrace-2107.html. В помощь радиолюбителю. Солон-Пресс 2012. – 200 с. Белов А.В. - Конструирование устройств на микроконтроллерах. НиТ, 2005– 256 с. Разработка программного обеспечения [Электронный ресурс] – URL: http://cxem.net/software/soft_mcu.php. Б. Керниган, Д. Ритчи: Язык программирования Си Издание 3-е, исправленное. Перевод с английского под редакцией Вс. С. Штаркмана, 1992 –253 с. Герберт Шилдт. Полный справочник по C (4-ое издание) 2004 – 704 с. Прокопенко В.С. – Программирование микроконтроллеров ATMEL на языке С. МК-Пресс, 2012 – 320 с. Шпак Ю.А. Программирование на языке Си для AVR и PIC микроконтроллеров 2006. – 300 с.

Цитирования: **0,03%** id: **49**

"Краткий справочник по языку “C”"

В.А.Ильина, П.К.Силаев, 2012, Московский государственный университет, физический факультет, кафедра квантовой теории и физики высоких энергий. Хартов В.Я. Проектирование и отладка программ для микроконтроллеров AVR фирмы Atmel. - М.: Изд-во МГТУ им. Н.Э. Баумана, 2004. ДОДАТОК А Лістинг програмного коду /************************************ Контроллер - ATmega16A/16L/32A/32L (камень без буквы может не заработать от 3х вольтовой батарейки) Частота чипа - 8,000000 MHz **********/ //поставьте единичку напротив примененного вами контроллера #define ATMEGA16 1 #define ATMEGA32 0 //*****************************/ //поставьте единичку напротив примененных вами индикаторов #define OK 1 // для матрицы с

Цитирования: **0,01%** id: **50**

"OK"

(на выводах 5, 2, 7, 1, 12, 8, 14, 9 катоды диодов) #define OA 0 // для матрицы с

Цитирования: **0,01%** id: **51**

"OA"

(на выводах 5, 2, 7, 1, 12, 8, 14, 9 аноды диодов) #define SAMPLE 0 // моя макетка..... на ней все не как у людей)))) ..... //***************/ //поставьте напротив FONT номер желаемого шрифта цифр (от 0 до 6) #define FONT 0 // выбираем шрифт цифр //*****************************/ # if ATMEGA16 #include mega16.h # endif # if ATMEGA32 #include mega32.h # endif #include 1wire.h #include ds18b20_.h //#include delay.h #define BUT_OK button==1 #define BUT_STEP button==2 #define POWER power #define ON 1 #define OFF 0 //*****************************/ bit mig, flg_min=0, bud_flg=0, but_flg=0, zv_kn=0, zv_chs=1, but_on=0, line=0, power=1, ds1820_d, ds1820_y, temp5=0; unsigned char sek=0, chislo=1, mesec=1, god=11, den_nedeli, bud_temp, stroka, meny=10, button, but_pause=0, z, z1, bud, temp, temp1, temp2, t, flg_korr=1, speed=28, devices, i, ekran [24], // Экранный буфер beg_info [99], // Бегущая строка в основном режиме rom_code [2][9], // масив с адресами найденых датчиков DS18B20 budilnik_Install [9], // храним настройки будильников budilnik_Interval[9]; // храним значение длительности сигнала будильника unsigned int budilnik_time [9]; // храним время сработки будильников const unsigned char budilnik_txt []= {11,29,14,18,21,38,23,18,20,42,255}, // текст

Цитирования: **0,02%** id: **52**

"Будильник" korekt_txt []

= {20,24,26,26,15,20,32,18,41,42,42,255}, // текст

Цитирования: **0,02%** id: **53**

"Коррекция" nastroiki_txt []

= {29,27,28,10,23,24,12,20,18,42,255}, // текст

Цитирования: **0,02%** id: **54**

"Установки" den_txt []

= {14,15,23,38,255}, // текст

Цитирования: **0,02%** id: **55**

"День" data_txt []
= {14,10,28,10,255}, // текст
Цитирования: **0,02%** id: **56**
"Дата" god_txt []
= {13,24,14,255}, // текст
Цитирования: **0,02%** id: **57**
"Год" nastr_stroki_txt []
= {23,10,27,28,26,24,19,20,10,42,255}, // текст
Цитирования: **0,02%** id: **58**
"Настройки" den_nedeli_txt [7][12]
= {{25,24,23,15,14,15,21,38,23,18,20,255}, // Понедельник // названия дней недели {12,28,24,26,23,18,20,255}, // Вторник // {27,26,15,14,10,255}, // Среда // {33,15,28,12,15,26,13,255}, // Четверг // {25,41,28,23,18,32,10,255}, // Пятница // {27,29,11,11,24,28,10,255}, // Суббота // {12,24,27,20,26,15,27,15,23,38,15,255}}, // Воскресенье // den_nedeli_letter[7][2]= {{25,23}, // Пн // сокращенные названия дней недели {12,28}, // Вт // {27,26}, // Ср // {33,28}, // Чт // {25,28}, // Пт // {27,11}, // Сб // {12,27}}, // Вс // name_mesec_txt [12][9]= {{41,23,12,10,26,41,255}, // Января {30,15,12,26,10,21,41,255}, // Февраля {22,10,26,28,10,255}, // Марта {10,25,26,15,21,41,255}, // Апреля {22,10,41,255}, // Мая {18,40,23,41,255}, // Июня {18,40,21,41,255}, // Июля {10,12,13,29,27,28,10,255}, // Августа {27,15,23,28,41,11,26,41,255}, //Сентября {24,20,28,41,11,26,41,255}, //Октября {23,24,41,11,26,41,255}, // Ноября {14,15,20,10,11,26,41,255}}; // Декабря flash const unsigned char simvol [][5]= { # if FONT==0 // Шрифт цифр №0 { 0x3E, 0x51, 0x49, 0x45, 0x3E }, // 0 { 0x00, 0x42, 0x7F, 0x40, 0x00 }, // 1 { 0x42, 0x61, 0x51, 0x49, 0x46 }, // 2 { 0x21, 0x41, 0x45, 0x4B, 0x31 }, // 3 { 0x18, 0x14, 0x12, 0x7F, 0x10 }, // 4 { 0x27, 0x45, 0x45, 0x45, 0x39 }, // 5 { 0x3C, 0x4A, 0x49, 0x49, 0x30 }, // 6 { 0x01, 0x71, 0x09, 0x05, 0x03 }, // 7 { 0x36, 0x49, 0x49, 0x49, 0x36 }, // 8 { 0x06, 0x49, 0x49, 0x29, 0x1E }, // 9 # endif # if FONT==1 // Шрифт цифр №1 { 0x7F, 0x7F, 0x41, 0x7F, 0x7F }, { 0x00, 0x00, 0x7F, 0x7F, 0x00 }, { 0x61, 0x71, 0x59, 0x4F, 0x47 }, { 0x41, 0x49, 0x49, 0x7F, 0x7F }, { 0x1F, 0x1F, 0x10, 0x7F, 0x7F }, { 0x4F, 0x4F, 0x49, 0x79, 0x79 }, { 0x7F, 0x7F, 0x49, 0x79, 0x79 }, { 0x01, 0x71, 0x79, 0x0F, 0x07 }, { 0x7F, 0x7F, 0x49, 0x7F, 0x7F }, { 0x5F, 0x5F, 0x51, 0x7F, 0x7F }, # endif # if FONT==2 // Шрифт цифр №2 { 0x7F, 0x7F, 0x41, 0x7F, 0x7F }, { 0x00, 0x01, 0x7F, 0x7F, 0x00 }, { 0x63, 0x73, 0x59, 0x4F, 0x47 }, { 0x63, 0x63, 0x49, 0x7F, 0x77 }, { 0x1F, 0x1F, 0x10, 0x7F, 0x7F }, { 0x6F, 0x6F, 0x49, 0x79, 0x79 }, { 0x7F, 0x7F, 0x49, 0x7B, 0x7B }, { 0x03, 0x73, 0x79, 0x0F, 0x07 }, { 0x77, 0x7F, 0x49, 0x7F, 0x77 }, { 0x6F, 0x6F, 0x49, 0x7F, 0x7F }, # endif # if FONT==3 // Шрифт цифр №3 { 0x7F, 0x41, 0x41, 0x7F, 0x7F }, { 0x00, 0x00, 0x7F, 0x7F, 0x00 }, { 0x61, 0x71, 0x59, 0x4F, 0x47 }, { 0x41, 0x49, 0x49, 0x7F, 0x7F }, { 0x1F, 0x10, 0x10, 0x7F, 0x7F }, { 0x4F, 0x49, 0x49, 0x79, 0x79 }, { 0x7F, 0x49, 0x49, 0x79, 0x79 }, { 0x01, 0x01, 0x01, 0x7F, 0x7F }, { 0x7F, 0x49, 0x49, 0x7F, 0x7F }, { 0x1F, 0x11, 0x11, 0x7F, 0x7F }, # endif # if FONT==4 // Шрифт цифр №4 { 0x7F, 0x41, 0x41, 0x7F, 0x7F }, { 0x00, 0x01, 0x7F, 0x7F, 0x00 }, { 0x63, 0x71, 0x59, 0x4F, 0x47 }, { 0x63, 0x41, 0x49, 0x7F, 0x7F }, { 0x1F, 0x10, 0x10, 0x7F, 0x7F }, { 0x6F, 0x49, 0x49, 0x79, 0x79 }, { 0x7F, 0x49, 0x49, 0x7B, 0x7B }, { 0x03, 0x01, 0x01, 0x7F, 0x7F }, { 0x7F, 0x49, 0x49, 0x7F, 0x7F }, { 0x1F, 0x11, 0x11, 0x7F, 0x7F }, # endif # if FONT==5 // Шрифт цифр №5 { 0x3E, 0x7F, 0x41, 0x7F, 0x3E }, { 0x00, 0x02, 0x7F, 0x7F, 0x00 }, { 0x62, 0x73, 0x59, 0x4F, 0x46 }, { 0x22, 0x63, 0x49, 0x7F, 0x36 }, { 0x18, 0x14, 0x12, 0x7F, 0x7F }, { 0x2F, 0x6F, 0x45, 0x7D, 0x39 }, { 0x3E, 0x7F, 0x49, 0x7B, 0x32 }, { 0x03, 0x73, 0x79, 0x0F, 0x07 }, { 0x36, 0x7F, 0x49, 0x7F, 0x36 }, { 0x2E, 0x6F, 0x49, 0x7F, 0x3E }, # endif # if FONT==6 // Шрифт цифр №6 { 0x3E, 0x41, 0x41, 0x7F, 0x3E }, { 0x00, 0x02, 0x7F, 0x7F, 0x00 }, { 0x62, 0x71, 0x59, 0x4F, 0x46 }, { 0x22, 0x41, 0x49, 0x7F, 0x36 }, { 0x18, 0x14, 0x12, 0x7F, 0x7F }, { 0x2F, 0x45, 0x45, 0x7D, 0x39 }, { 0x3E, 0x49, 0x49, 0x7B, 0x32 }, { 0x03, 0x71, 0x79, 0x0F, 0x07 }, { 0x36, 0x49, 0x49, 0x7F, 0x36 }, { 0x26, 0x49, 0x49, 0x7F, 0x3E }, # endif { 0x7C, 0x12, 0x12, 0x7C, 0xAA }, // A { 0x7E, 0x4A, 0x4A, 0x32, 0xAA }, // Б { 0x7E, 0x4A, 0x4A, 0x34, 0xAA }, // В { 0x7E, 0x02, 0x02, 0xAA, 0x00 }, // Г { 0x60, 0x3C, 0x22, 0x3E, 0x60 }, // Д { 0x7E, 0x4A, 0x4A, 0xAA, 0x00 }, // Е { 0x4A, 0x4A, 0x7E, 0xAA, 0x00 }, // З { 0x66, 0x18, 0x7E, 0x18, 0x66 }, // Ж { 0x7E, 0x10, 0x08, 0x7E, 0xAA }, // И { 0x7C, 0x11, 0x09, 0x7C, 0xAA }, // И { 0x7E, 0x18, 0x24, 0x42, 0xAA }, // К { 0x78, 0x04, 0x02, 0x7E, 0xAA }, // Л { 0x7E, 0x04, 0x08, 0x04, 0x7E }, // М { 0x7E, 0x08, 0x08, 0x7E, 0xAA }, // Н { 0x3C, 0x42, 0x42, 0x3C, 0xAA }, // О { 0x7E, 0x02, 0x02, 0x7E, 0xAA }, // П { 0x7E, 0x12, 0x12, 0x0C, 0xAA }, // Р { 0x3C, 0x42, 0x42, 0x24, 0xAA }, // С { 0x02, 0x7E, 0x02, 0xAA, 0x00 }, // Т { 0x4E, 0x50, 0x50, 0x3E, 0xAA }, // У { 0x0C, 0x12, 0x7E, 0x12, 0x0C }, // Ф { 0x66, 0x18, 0x18, 0x66, 0xAA }, // Х { 0x7E, 0x40, 0x40, 0x7E, 0xC0 }, // Ц { 0x0E, 0x10, 0x10, 0x7E, 0xAA }, // Ч { 0x7E, 0x40, 0x7E, 0x40, 0x7E }, // Ш { 0x7E, 0x40, 0x7E, 0x40, 0xFE }, // Щ { 0x02, 0x7E, 0x48, 0x30,

0xAA }, // Ъ { 0x7E, 0x48, 0x30, 0x00, 0x7E }, // Ы { 0x7E, 0x48, 0x30, 0xAA, 0x00 }, // Ь { 0x24, 0x42, 0x4A, 0x3C, 0xAA }, // Э { 0x7E, 0x08, 0x3C, 0x42, 0x3C }, // Ю { 0x4C, 0x32, 0x12, 0x7E, 0xAA }, // Я { 0x00, 0x00, 0xAA, 0xAA, 0x00 }, // пробел { 0x40, 0xAA, 0xAA, 0xAA, 0x00 }, // точка 43 { 0xAA, 0xAA, 0xAA, 0xAA, 0x00 }, // маленький пробел 44 { 0x00, 0x00, 0x00, 0x00, 0x00 }, //

Цитирования: **0,01%** id: **59**

"полный"

пробел 45 { 0x08, 0x08, 0x08, 0x08, 0x08 }, // минус 46 { 0x08, 0x08, 0x3E, 0x08, 0x08 }, // плюс 47 { 0x06, 0x09, 0x09, 0x06, 0xAA }, // знак градуса 48 { 0x3E, 0x41, 0x41, 0x41, 0x22 }, // большая C 49 { 0x04, 0x3F, 0x44, 0x20, 0xAA }, // прописная t 50 { 0x08, 0x08, 0x08, 0xAA, 0x00 }, // маленький минус 51 }, dnei_v_mes[]= {31,29,31,30,31,30,31,31,30,31,30,31}; // Количества дней по месяцам unsigned int time=1392 , Interval , Interval_2, data1; signed int temperature [2]={200,300}; float temperature_temp; eeprom signed int korr_den=0; // значение коррекции хода eeprom unsigned char str=255; // конфигурация бегущей строки // eeprom unsigned char ds1820_d_eeprom, ds1820_y_eeprom; //___Возвращает абсолютное значение числа_____ signed int abs (signed int x) { if (x 0) x=(x*(-1)); return x; } //_____вычислить день недели по дате_____ unsigned char Day_week (void) { unsigned char y, m, myday; y = god; m = mesec; if (m 2) { m -= 2; } else { m += 10; y--; } myday = ((chislo + y + (y 2) + ((31 * m) / 12)) % 7); if (myday) return myday-1; else return 6; } //_____коррекция времени_____ void correkt (void) { if(korr_den 0) {TCNT2=255+((korr_den%10)*25); sek=59+((korr_den%600)/10); if (sek==60) {sek=0;} time=1439;} else {TCNT2=25*(korr_den%10); sek=((korr_den%600)/10);} } //_Диамическая индикация.вывод данных из экранного буфера на светодиодную матрицу__ interrupt [TIM0_OVF] void timer0_ovf_isr(void) { # if OK || OA stroka = 1; if (stroka==0) stroka=0b10000000; // перебираю строки if (stroka==0b00000001) PORTB.7= 1; else PORTB.7=0; # endif # if SAMPLE stroka = 1; if (stroka==0) stroka=0b00000001; // перебираю строки if (stroka==0b10000000) PORTB.7= 1; else PORTB.7=0; # endif PORTB.6=1; // # if OK PORTC.3 = (ekran [0] & stroka); PORTC.0 = (ekran [1] & stroka); PORTD.7 = (ekran [2] & stroka); PORTC.5 = (ekran [3] & stroka); PORTD.6 = (ekran [4] & stroka); PORTC.4 = (ekran [5] & stroka); PORTC.2 = (ekran [6] & stroka); PORTC.1 = (ekran [7] & stroka); PORTA.5 = (ekran [8] & stroka); PORTD.5 = (ekran [9] & stroka); PORTD.4 = (ekran [10] & stroka); PORTA.3 = (ekran [11] & stroka); PORTD.3 = (ekran [12] & stroka); PORTA.4 = (ekran [13] & stroka); PORTA.6 = (ekran [14] & stroka); PORTA.7 = (ekran [15] & stroka); PORTA.0 = (ekran [16] & stroka); PORTD.2 = (ekran [17] & stroka); PORTD.1 = (ekran [18] & stroka); PORTB.1 = (ekran [19] & stroka); PORTD.0 = (ekran [20] & stroka); PORTB.0 = (ekran [21] & stroka); PORTA.1 = (ekran [22] & stroka); PORTA.2 = (ekran [23] & stroka); # endif # if OA PORTC.3 = (ekran [0] & stroka); PORTC.0 = (ekran [1] & stroka); PORTD.7 = (ekran [2] & stroka); PORTC.5 = (ekran [3] & stroka); PORTD.6 = (ekran [4] & stroka); PORTC.4 = (ekran [5] & stroka); PORTC.2 = (ekran [6] & stroka); PORTC.1 = (ekran [7] & stroka); PORTA.5 = (ekran [8] & stroka); PORTD.5 = (ekran [9] & stroka); PORTD.4 = (ekran [10] & stroka); PORTA.3 = (ekran [11] & stroka); PORTD.3 = (ekran [12] & stroka); PORTA.4 = (ekran [13] & stroka); PORTA.6 = (ekran [14] & stroka); PORTA.7 = (ekran [15] & stroka); PORTA.0 = (ekran [16] & stroka); PORTD.2 = (ekran [17] & stroka); PORTD.1 = (ekran [18] & stroka); PORTB.1 = (ekran [19] & stroka); PORTD.0 = (ekran [20] & stroka); PORTB.0 = (ekran [21] & stroka); PORTA.1 = (ekran [22] & stroka); PORTA.2 = (ekran [23] & stroka); # endif # if SAMPLE PORTB.1 = (ekran [0] & stroka); // 0 PORTB.0 = (ekran [1] & stroka); // 1 PORTA.3 = (ekran [2] & stroka); // 2 PORTA.1 = (ekran [3] & stroka); // 3 PORTA.4 = (ekran [4] & stroka); // 4 PORTA.2 = (ekran [5] & stroka); // 5 PORTA.0 = (ekran [6] & stroka); // 6 PORTA.5 = (ekran [7] & stroka); // 7 PORTA.6 = (ekran [8] & stroka); // 8 PORTA.7 = (ekran [9] & stroka); // 9 PORTC.5 = (ekran [10] & stroka); // 10 PORTC.4 = (ekran [11] & stroka); // 11 PORTC.3 = (ekran [12] & stroka); // 12 PORTC.2 = (ekran [13] & stroka); // 13 PORTC.1 = (ekran [14] & stroka); // 14 PORTC.0 = (ekran [15] & stroka); // 15 PORTD.2 = (ekran [16] & stroka); // 16 PORTD.7 = (ekran [17] & stroka); // 17 PORTD.0 = (ekran [18] & stroka); // 18 PORTD.6 = (ekran [19] & stroka); // 19 PORTD.4 = (ekran [20] & stroka); // 20 PORTD.5 = (ekran [21] & stroka); // 21 PORTD.3 = (ekran [22] & stroka); // 22 PORTD.1 = (ekran [23] & stroka); // 23 # endif PORTB.6=0; but_pause++; if (but_pause==100) { but_pause=0; but_on=1; } // если с момента прошлого нажатия кнопки прошло больше 0,3 сек - разрешаю очередное чтение кнопок if ((but_pause==30)&& (but_flg)) { but_flg=0; PORTB.5=0; TIMSK=0x41;} // включаю

Цитирования: **0,01%** id: **60**

"писк"

при нажатии кнопки Interval++; Interval_2++; // отсчитываем интервал для бегущих строк и прочих нужд } //_____Чтение состояния кнопок____ unsigned char button_read (void) { button=0; if ((PINB.2==0) && (but_on)) { but_on=0; but_pause=0; # if OK || OA switch (stroka) { case 0b10000000: button=2; break; case 0b00000001: button=1; break; } # endif # if SAMPLE

switch (stroka) { case 0b00010000: button=2; break; case 0b00001000: button=1; break; } # endif if (zv_kn) {but_flg=1; TIMSK=0x51;} // Если не отключен - короткий писк динамика при нажатии кнопки . if (bud_flg) {bud_flg=0; button=0;} // Выключаем (если был включен) сигнал будильника.это нужно чтоб можно было выключить сигнал будильника просто нажатием любой кнопки. bud_flg=0; } return button; } //_____Прерывание от компаратора(заходим сюда когда пропадает и когда появляется внешнее напряжение)_____ interrupt [ANA_COMP] void ana_comp_isr(void) { if(ACSR&0x20) // если пропало внешнее питания { POWER = OFF; // переходим на резервное питание от батареек # if ATMEGA16 MCUCR=0b01110000; // разрешаем усыплять контроллер по команде SLEEP (для мега16) # endif # if (ATMEGA32) MCUCR=0b10110000; // разрешаем усыплять контроллер по команде SLEEP (для мега32) # endif TCCR0=0x00; // останавливаем T/C0 PORTA=0; DDRB=0b11110111; // перевожу порты в состояние наименьшего потребления PORTB=0; PORTC=0; PORTD=0; } else // если внешнее питания появилось { POWER = ON; // переходим на внешнее питание MCUCR=0b00110000; // запрещаем усыплять контроллер по команде SLEEP PORTB=0x04; DDRB=0xE3; // возвращаем конфигурацию портов в рабочее состояние TCCR0=0x03; // запускаем T/C0 } } //_____пищу динамиком_____ interrupt [TIM1_COMPA] void timer1_compa_isr(void) { static char dl_1=0,dl_2=0; dl_2++; if (dl_2 =150) {dl_2=0; dl_1++; if (dl_1 =5) dl_1=0; } if (but_flg) OCR1A=5; // если

Цитирования: **0,01%** id: **61**

"пищим"

по нажатию кнопки, или

Цитирования: **0,01%** id: **62**

"ежечасно"

то сигнал однотонный else OCR1A = dl_1+3; // если

Цитирования: **0,01%** id: **63**

"пищит"

будильник, то сигнал

Цитирования: **0,01%** id: **64**

"музыкальный"

PORTB ^= (1 5); } //_____отсчет времени_____ interrupt [TIM2_OVF] void timer2_ovf_isr(void) { mig++; sek++; // Инкреминируем секунды if (sek==60) {sek=0; time++; flg_min=1; if (time==1440) { if(flg_korr) {correkt(); flg_korr=0; if(korr_den 0) goto m2;} // в 00-00, если еще сегодня не проводилась коррекция - провести коррекцию. time=0; chislo++; if (chislo dnei_v_mesc[mesec-1]) {chislo=1; mesec++; if (mesec==13) {mesec=1; god++; } } if ((god%4) && (mesec==2) && (chislo==29)) {mesec=3; chislo=1;} // если НЕвысокосный год, то 29.02 01.03 flg_korr=1; m2: } } } //___загрузка в экранный буфер символов + мигающая точка в центре_____ void ekran_cifri (unsigned int data) { unsigned char x,x1,x2,x3,y,y1,y2,y3,temp; static unsigned char Shift_zn; x = data/600; x1 = (data%600)/60; x2 = (data%60)/10; x3 = data%10; y = data1/600; y1 = (data1%600)/60; y2 = (data1%60)/10; y3 = data1%10; if (data != data1) // если сменилась информация - запускаем скроллинг {if (Interval_2 50){Interval_2=0; Shift_zn++; if(Shift_zn==8) {Shift_zn=0;data1=data;} } {for (temp=0; temp 5; temp++) { if(x !=y) {ekran [temp] = ((simvol [y][temp] Shift_zn) + (simvol [x][temp] (8-Shift_zn)));} else { ekran [temp]=(simvol [x][temp]); } if(x1!=y1) {ekran [temp+ 6] = ((simvol [y1][temp] Shift_zn) + (simvol [x1][temp] (8-Shift_zn)));} else { ekran [temp+ 6]=(simvol [x1][temp]); } if(x2!=y2) {ekran [temp+13] = ((simvol [y2][temp] Shift_zn) + (simvol [x2][temp] (8-Shift_zn)));} else { ekran [temp+13]= (simvol [x2][temp]); } ekran [temp+19] = ((simvol [y3][temp] Shift_zn) + (simvol [x3][temp] (8-Shift_zn))); } } } else { for (temp=0; temp 5; temp++) { ekran [temp]=(simvol [x][temp]); ekran [temp+ 6]=(simvol [x1][temp]); ekran [temp+13]=(simvol [x2][temp]); ekran [temp+19]= (simvol [x3][temp]); } Interval_2=0; } if (mig) {ekran [12]=0; ekran [11]=128;} // перемигивание точек в основном режиме. else {ekran [12]=128; ekran [11]=0;} // }//загрузка в отределенное место экрана 1 символа_____ void ekran_1_figure (unsigned char x,unsigned char x1,) { unsigned char temp; for (temp=0; temp 5; temp++) { if (simvol [x][temp]==0xAA) return; ekran [temp+x1]=(simvol [x][temp]) +(128*mig*line); } } //_____Гасим весь экран_____ void ochistka (void) { unsigned char temp; for (temp=0; temp 24; temp++) {ekran[temp]=0 ;} } //__установки!__2_символа_____ unsigned char ystanovki_2 (unsigned char x,unsigned char x1,unsigned char x2) { unsigned char temp; while ((button_read() != 2) && POWER) // POWER { if (BUT_OK) {x++; if (x x1) x=0;} for (temp=0; temp 5; temp++) { ekran [temp+x2] =(simvol [x/10][temp]) +(128*mig); ekran [temp+x2+6]=

(simvol [x%10][temp]) +(128*mig); } } ochistka(); return x; } //___установки_____ unsigned int ystanovki_23_59 (unsigned int x) { ochistka(); ekran_1_figure((x%60)/10,13); ekran_1_figure(x%10,19); x = (x%60)+ (ystanovki_2 ((x/60),23, 0))*60; ekran_1_figure(x/600,0); ekran_1_figure((x/60)%10,6); x = (x/60*60)+(ystanovki_2 ((x%60),59,13)); return x; } //_____бегущая строка_____ unsigned char beg_stroka (unsigned char x[],) { unsigned char temp; for (temp=0; temp 23; temp++) { ekran[temp]=ekran[temp+1]; } if ((z==5)|| (simvol[x[z1]][z]==0xAA)) {ekran[23]=0; z=0; z1++; if (x[z1]==255) {z1=0; return 255;} } else {ekran[23]=simvol[x[z1]][z]; z++;} }//_____вывод неподвижного текста_____ void txt_ekran (unsigned char x[],) { unsigned char temp =0, temp1=0, temp2=0; for(temp=0;temp 24;temp++) { if ((temp2==5) || (simvol[x[temp1]][temp2] == 0xAA)) {ekran[temp]=0; temp2=0; temp1++;} else {ekran[temp]=simvol[x[temp1]][temp2]; temp2++;} if (x[temp1]==255) return ; //{temp=24;} } } //**********// void main(void) { PORTA=0x00; DDRA=0xFF; PORTB=0x04; DDRB=0xE3 ; PORTC=0x00; DDRC=0x3F; PORTD=0x00; DDRD=0xFF; TCCR0=0x03; // Частота T0 125,000 kHz (8000000/64) TCCR1B=0x0C; // Частота T1 125,000 kHz (8000000/64) OCR1A=20; ASSR=0x08; // Такт от ног TOSC1,2 с кварцем на 32768 TCCR2=0x05; // 32768/128=256 TIMSK=0x41; // Конфигурирую прерывания от таймеров ACSR= 0x48; // Компаратор. SFIOR=0x00; MCUCR|=0b00110000; // выбираю режим пониженного энергопотребления - Power Sawe #asm // настраиваю шину 1 Wire .equ __w1_port=0x18 ;PORTB // на работу с портов B .equ __w1_bit=4 // пин 5 #endasm for (temp=0;temp 9;temp++){budilnik_time [temp] = budilnik_Install [temp] = budilnik_Interval[temp]= 0;} temp=0; bud_flg=0; //

Цитирования: **0,01%** id: **65**

"выключаем"

все будильники den_nedeli=Day_week (); // Вычисляем день недели ochistka(); // Очищаем весь экран devices=w1_search(0xf0,rom_code); // Ищим датчики, и записываем их адреса в масив ds1820_d=0; ds1820_y=1; #asm(

Цитирования: **0,01%** id: **66**

"sei"

) //**********// while (1) { button_read(); // Опрос кнопок switch (meny) { case 10: //_____основной режим (ЧАСЫ) ekran_cifri(time); //_____измерение температуры if (sek%5==0) // каждую секунду { //devices=w1_search(0xf0,rom_code); //ищим датчики, и записываем их адреса в масив for (i=0;i devices;i++) { temperature_temp=ds18b20_read_temp (&rom_code[i][0]); // читаю датчик температуры DS18B20 if (temperature_temp!=(-9999)) {temperature[i]=temperature_temp*10; temperature[i]+=5;} // если температура и влажность прочитались правильно, заносим их в

Цитирования: **0,01%** id: **67**

"temperature"

и

Цитирования: **0,01%** id: **68**

"humidity"

ds18b20_convert_temp(&rom_code[i][0]); // команда на измерение температуры и влажности } } if ((sek==temp) && (str != 0)) { meny=11 ;z=0; z1=0; temp2=str; } // Запускаем бегущую строку if (BUT_STEP) {meny=20; temp=30; ochistka(); bud_flg=0;data1=(((time-(time/60*60))*60)+sek);} if (BUT_OK) {meny=11 ;z=0; z1=0; temp2=255;} break; case 11: //_____Формируем и вывожу бег строку. //devices=w1_search(0xf0,rom_code); t=0; temp1=0; den_nedeli=Day_week (); beg_info[t++] = beg_info[t++] = beg_info[t++] = 42; // 3 пробела в начале строки if (temp2 & 0x01) // Если

Цитирования: **0,01%** id: **69**

"день недели"

нужно выводить { while (den_nedeli_txt[den_nedeli][t-3] != 255) { beg_info[t]=den_nedeli_txt[den_nedeli][t++-3]; // } beg_info[t++] = 42; // пробел } if (temp2 & 0x02) // Если

Цитирования: **0,01%** id: **70**

"дату"

нужно выводить { if (chislo 9) {beg_info[t++]=chislo/10;} // Если число больше 9, выводим

Цитирования: **0,02%** id: **71**

"десятки" числа beg_info[t++]

=chislo%10; // Выводим

Цитирования: **0,02%** id: **72**

"Единицы" числа beg_info[t++]

=42; // Пробел temp1=t; while (name_mesec_txt[mesec-1][(t-temp1)] != 255) // Выводим месяц { beg_info[t]=name_mesec_txt[mesec-1][t++-temp1]; } beg_info[t++] = 42; // пробел } if (temp2 & 0x04) // Если

Цитирования: **0,01%** id: **73**

"Год"

нужно выводить { beg_info[t++]=2; //

Цитирования: **0,02%** id: **74**

"Тысячи" года (2) beg_info[t++]

=0; //

Цитирования: **0,02%** id: **75**

"Сотни" года (0) beg_info[t++]

=(god%100)/10; //

Цитирования: **0,02%** id: **76**

"Десятки" года beg_info[t++]

=god%10; //

Цитирования: **0,02%** id: **77**

"Единицы" года beg_info[t++]

=13; //

Цитирования: **0,01%** id: **78**

"Г" beg_info[t++]

=42; beg_info[t++]=42; // 2 пробела } if (temp2 & 0x08) // Если

Цитирования: **0,02%** id: **79**

"Температура в доме"

нужно выводить { beg_info[t++]=42; // пробел if (temperature[ds1820_d] 0) beg_info[t++]=51; else beg_info[t++]=47; // если темп меньше нуля - пишем знак минус, если больше - знак плюс if (abs(temperature[ds1820_d]) 99) {beg_info[t++]= (abs(temperature[ds1820_d])/100);}// Если темп 10, выводим

Цитирования: **0,01%** id: **80**

"десятки"

температуры дома beg_info[t++]=(abs(temperature[ds1820_d])%100)/10; // Выводим

Цитирования: **0,02%** id: **81**

"единицы температуры" beg_info[t++]

=48; // Знак градуса beg_info[t++]=44; // мал пробел if ((temp2 & 0x10) && (devices 1)) {beg_info[t++]=14; beg_info[t++]=24; beg_info[t++]=22; beg_info[t++]=42;} // слово

Цитирования: **0,01%** id: **82**

"ДОМ" beg_info[t++]

=42; // пробел } if ((temp2 & 0x10) && (devices 1)) // Если

Цитирования: **0,02%** id: **83**

"Температура в оранжерее"

нужно выводить { if (temperature[ds1820_y] 0) beg_info[t++]=51; else beg_info[t++]=47; // если темп меньше нуля - пишем знак минус if (abs(temperature[ds1820_y]) 99) {beg_info[t++]=(abs(temperature[ds1820_y])/100);}// Если модуль температуры 10, выводим

Цитирования: **0,01%** id: **84**

"десятки"

температуры дома beg_info[t++]=(abs(temperature[ds1820_y])%100)/10; // Выводим

Цитирования: **0,02%** id: **85**

"единицы температуры" beg_info[t++]

=48; // Знак градуса beg_info[t++]=42; // мал пробел beg_info[t++]=29; beg_info[t++]=21; // слово

Цитирования: **0,01%** id: **86**

"УЛ" beg_info[t++]

=42; beg_info[t++]=42; // пробел } beg_info[t++]=time/600; // десятки часов beg_info[t++]=(time%600)/60; // единицы часов beg_info[t++]=44; // разделительная точка beg_info[t++]=(time%60)/10; // десятки минут beg_info[t++]=time%10; // единицы минут beg_info[t]=255; // метка конца

Цитирования: **0,01%** id: **87**

"бегущей строки"

if (Interval = speed) {Interval=0; if (beg_stroka(beg_info)==255) {temp=sek-1; if (temp==255) temp=59; meny=10; ochistka(); data1=time;}} if (BUT_STEP) {meny=20; temp=30; ochistka();} if (BUT_OK) {z=0; z1=3; temp=0x0F; ochistka();} break; case 20: //**********Секунды********** mig=1; ekran_cifri(((time-(time/60*60))*60)+sek); if (BUT_STEP) {meny=temp; z=0; z1=0; ochistka();} if (BUT_OK) {if (sek 40) time++; sek=0; TCNT2=0; temp=10;} break; case 30: //*****Установка времени и даты********** if (Interval =speed) {Interval=0; beg_stroka(nastroiki_txt);} if (BUT_STEP) {meny=40; z=0; z1=0; ochistka();} if (BUT_OK) { time =ystanovki_23_59(time); ochistka(); ekran_1_figure(33,1); ekran_1_figure(43,6); chislo =ystanovki_2 (chislo,31, 13); ekran_1_figure(22,1); ekran_1_figure(43,7); mesec =ystanovki_2 (mesec, 12, 13); ekran_1_figure(13,1); ekran_1_figure(43,5); god =ystanovki_2 (god, 99, 13); button=0;meny=10;ochistka(); temp=0; data1=time; } break; //******Установки будильников********** case 40: // На экране текст -

Цитирования: **0,01%** id: **88**

"Будильник"

if (Interval =speed) {Interval=0; beg_stroka(budilnik_txt);} if (BUT_STEP) {meny=50;z=0; z1=0;ochistka();} if (BUT_OK) {ochistka(); bud=0; temp=0; meny=41;} break; case 41: // выбираем номер будильника if (BUT_STEP) {meny=42;} if (BUT_OK) {bud++; if (bud==9) bud=0; } ekran_1_figure(11,1); line=1; ekran_1_figure(bud+1,7); line=0; ekran_1_figure(((budilnik_Install[bud] & 0x80) ? (47):(46)),17); break; case 42: // включаем или отключаем его.если отключили - переходим в режим

Цитирования: **0,01%** id: **89**

"часы"

if (BUT_STEP) { if (budilnik_Install[bud] & 0x80) // если текущий будильник включен { budilnik_time[bud]= ystanovki_23_59 (budilnik_time[bud]); // устанавливаем время сработки будильника if (bud 3){budilnik_Install[bud] = 0xFF; goto m1;} // если будильник №1-3 то установки

Цитирования: **0,02%** id: **90**

"по дням недели"

не производим, и переходим сразу к настройке длительности сигнала этого будильника else {temp=0; meny=43;} // если будильник №4-9 то переходим к настройке будильника на сработку в определенные дни } else {meny=10;} ochistka(); break; // если текущий будильник отключен - переходим в режим

Цитирования: **0,01%** id: **91**

"часы"

} if (BUT_OK) {budilnik_Install[bud] ^= 0x80;} // каждое нажатие включает/отключает конкретный будильник (устанавливает/сбрасывает в 1 бит7) ekran_1_figure(11,1); ekran_1_figure(bud+1,7); line=1; ekran_1_figure(((budilnik_Install[bud] & 0x80) ? (47):(46)),17); line=0; break; case 43: // Настраиваем будильник на сработку в определенные дни, и длительность его сигнала. ekran_1_figure (den_nedeli_letter[temp][0],0); ekran_1_figure (den_nedeli_letter[temp][1],6); // вывожу названия дней недели. (массив

Цитирования: **0,01%** id: **92**

"beg_info"

содержит название дня недели) ekran_1_figure (((budilnik_Install[bud] & (1 temp)) ? 47:46),17); // вывожу знак

Цитирования: **0,01%** id: **93**

"+"

или

Цитирования: **0,01%** id: **94**

"-"

обозначающий вкл./выкл. будильника. if (BUT_STEP) {temp++; ochistka(); //

Цитирования: **0,01%** id: **95**

"перебираю"

дни недели для будильника if (temp==7){ m1: meny=10; budilnik_Interval[bud] = ystanovki_2(1,15,8); } // если все дни недели установлены, задаю время звучания сигнала. } if (BUT_OK){(budilnik_Install[bud]) ^= (1 temp);} // включаю/отключаю будильник в конкретный день недели. break; //*******Настройка бег строки. Выбираем какую информацию будем выводить с помощью бег.строки********** case 50: // на экране текст -

Цитирования: **0,01%** id: **96**

"Коррекция"

if (Interval =speed) {Interval=0; beg_stroka(nastr_stroki_txt);} if (BUT_STEP) {meny=60; z=0; z1=0; ochistka();} if (BUT_OK) {meny=51; ochistka(); temp=0;} break; case 51: if (BUT_STEP) {temp++; ochistka();} switch (temp) { case 0: if (BUT_OK){str ^= (1 0);} ekran_1_figure (((str & 0x01) ? 47:46),19); txt_ekran(den_txt); break; case 1: if (BUT_OK){str ^= (1 1);} ekran_1_figure (((str & 0x02) ? 47:46),19); txt_ekran(data_txt); break; case 2: if (BUT_OK){str ^= (1 2);} ekran_1_figure (((str & 0x04) ? 47:46),19); txt_ekran(god_txt); break; case 3: if (BUT_OK){str ^= (1 3);} ekran_1_figure (((str & 0x08) ? 47:46),19); ekran_1_figure(50,0); ekran_1_figure(48,4); ekran_1_figure(14,10); break; case 4: if (BUT_OK){str ^= (1 4);} ekran_1_figure (((str & 0x10) ? 47:46),19); ekran_1_figure(50,0); ekran_1_figure(48,4); ekran_1_figure(29,10); break; case 5: if (BUT_OK){zv_chs++; } ekran_1_figure (((zv_chs) ? 47:46),19); ekran_1_figure(16,0); ekran_1_figure(12,4); ekran_1_figure(43,8); ekran_1_figure(33,9); ekran_1_figure(27,14); break; case 6: if (BUT_OK){zv_kn++; } ekran_1_figure (((zv_kn) ? 47:46),19); ekran_1_figure(16,0); ekran_1_figure(12,4); ekran_1_figure(43,8); ekran_1_figure(20,9); ekran_1_figure(23,14); break; case 7: meny=52; temp=speed; break; } break; case 52: // настройка скорости бегущей строки if (BUT_OK) {temp+=3; if (temp =60) {temp=9;}} if (BUT_STEP) {speed=temp; if (devices==2) meny=53; else meny=53; ochistka();} if (Interval =temp) {Interval=0; beg_stroka(beg_info);} break; case 53: // переназначение датчиков ds18m20 (если подключено 2 датчика) if (BUT_OK) {temp5++;} if (BUT_STEP) { meny=10; if (temp5) {ds1820_d=1; ds1820_y=0;} else {ds1820_d=0; ds1820_y=1;} ochistka(); } ekran_1_figure (14,0); ekran_1_figure (((temperature[temp5] 0) ? 51:47),8); ekran_1_figure ((abs(temperature[temp5])/100),13); ekran_1_figure (((abs(temperature[temp5])%100)/10),19); break; //********Настройка коррекции хода********** case 60: // на экране текст -

Цитирования: **0,01%** id: **97**

"Коррекция"

if (Interval =speed) {Interval=0; beg_stroka(korekt_txt);} if (BUT_STEP) {meny=10; ochistka();} if (BUT_OK) {meny=61; temp=0;ochistka();} break; case 61: // Установка

Цитирования: **0,01%** id: **98**

"знака"

коррекции if (BUT_STEP) {meny=62;} if (BUT_OK) {korr_den = (korr_den * (-1));} line=1; ekran_1_figure(((korr_den 0)?46:47),0); line=0; ekran_1_figure((abs((korr_den))/100),6); ekran_1_figure(((abs(korr_den)%100)/10),12); ekran_1_figure((abs(korr_den)%10),19); break; case 62: // Установка секунд коррекции if (BUT_STEP) {meny=63;} if (BUT_OK) {korr_den= ((korr_den 0)?(korr_den-=10):(korr_den+=10));if(abs(korr_den) 599){korr_den=korr_den%10;}} ekran_1_figure(((korr_den 0)?46:47),0); line=1; ekran_1_figure((abs((korr_den))/100),6); ekran_1_figure(((abs(korr_den)%100)/10),12); line=0; ekran_1_figure((abs(korr_den)%10),19); break; case 63: // Установка десятых долей секунд коррекции if (BUT_STEP) {meny=10;} if (BUT_OK) {if(korr_den 0){korr_den-- ;if(korr_den%10==0)korr_den+=10;}else{korr_den++;if(korr_den%10==0)korr_den-=10;}} ekran_1_figure(((korr_den 0)?46:47),0); ekran_1_figure((abs((korr_den))/100),6); ekran_1_figure(((abs(korr_den)%100)/10),12); line=1; ekran_1_figure((abs(korr_den)%10),19); line=0; break; } //***Сюда заходим каждую минуту***** if (flg_min) { flg_min=0; //_____включение будильников for (bud_temp=0; bud_temp 9; bud_temp++) { if ((time==budilnik_time[bud_temp]) // если наступило время срабатывания будильника && (budilnik_Install[bud_temp] & 0x80) // И этот будильник включен (должен срабатывать) && (budilnik_Install[bud_temp] & (1 den_nedeli)) // И должен сработать в текущий день недели) bud_flg=1; // то включить сигнал //_____Выключение будильников if (time==budilnik_time[bud_temp] + budilnik_Interval[bud_temp]) // если будильник

Цитирования: **0,01%** id: **99**

"отзвенел"

установленное время(от 1 до 15 мин) { bud_flg=0; // то выключить сигнал if (bud_temp 3) {budilnik_Install[bud_temp] = 0;} // и если будильник

Цитирования: **0,01%** id: **100**

"одноразовый"

(№ 1-3), то выключаем его повторную сработку } } //_____ежечасный сигнал if ((time%60==0)&&(time 500)&&(zv_chs)) { but_flg=1; TIMSK=0x51; but_pause=0; } // Писк каждый час (с 00-00 до 08-00 сигнал не срабатывает) } if (((bud_flg)&&(mig)) || (but_flg)) { TIMSK=0x51;} // включаем прерывание по совпадению

Цитирования: **0,01%** id: **101**

"A"

с T1 (Пищим динамиком. Будильник) else {PORTB.5=0; TIMSK=0x41;} //********** while (POWER == OFF) // если работаем от батареи, то сидим здесь и не вылазим - усыпляем контроллер { ACSR=0x80; // выключаю компаратор и внутренний ИОН на время сна (для экономии энергии батарейки) #asm(

Цитирования: **0,01%** id: **102**

"sleep"

) // спим..... ACSR=0x4b; // включаю компаратор, внутренний ИОН. } //*********

Заявление об ограничении ответственности:

Этот отчет должен быть правильно истолкован и проанализирован квалифицированным специалистом, который несет ответственность за оценку!

Любая информация, представленная в этом отчете, не является окончательной и подлежит ручному просмотру и анализу.
Пожалуйста, следуйте инструкциям: Рекомендации по оценке

Детектор Плагиата - Ваше право на оригинальность! □ SkyLine LLC