

Міністерство освіти і науки України
Державний заклад
«Луганський національний університет імені Тараса Шевченка»

Навчально-науковий інститут математики та інформаційних технологій

Кафедра інформаційних технологій та систем

Козлов Володимир Андрійович

**РОЗРОБКА ІНФОРМАЦІЙНО-ВИМІРЮВАЛЬНОЇ СИСТЕМИ
КОНТРОЛЮ ТЕМПЕРАТУРИ І ВОЛОГОСТІ ПОВІТРЯ НА ОСНОВІ
МАТЕМАТИЧНОГО ДОСЛІДЖЕННЯ МІКРОКЛІМАТУ
ТЕПЛИЧНИХ ПРИМІЩЕНЬ**

кваліфікаційна робота

здобувача вищої освіти другого (магістерського) рівня

освітньої програми «Комп'ютерні мережі»

за спеціальністю 123 Комп'ютерна інженерія

Особистий підпис _____ Володимир КОЗЛОВ

Науковий керівник _____ Володимир ДОНЧЕНКО,
старший викладач
кафедри інформаційних технологій
та систем

В.о. завідувача кафедри _____ Микола СЕМЕНОВ,
кандидат педагогічних наук, доцент
кафедри інформаційних технологій
та систем

Полтава – 2024

АНОТАЦІЯ

Козлов В. А.

Тема: Розробка інформаційно-вимірювальної системи контролю температури і вологості повітря на основі математичного дослідження мікроклімату тепличних приміщень.

Спеціальність: 123 «Комп'ютерна інженерія».

Установа: ЛНУ імені Тараса Шевченка, 2024р.

Магістерська робота містить: 67 с., 39 рис., 2 табл., 34 джерела.

Об'єктом дослідження є теплиці як об'єкт автоматизації та удосконалення систем автоматичного керування.

Предмет дослідження – математична модель мікроклімату і система автоматичного контролю параметрів мікроклімату в теплиці.

Мета роботи – розробка інформаційно-вимірювальної системи контролю температури і вологості повітря на основі математичного дослідження мікроклімату тепличних приміщень.

Результати роботи. Проведено аналіз існуючого вигляду пристрою, використовуваного для контролю вологості і температури в теплиці на основі плати Arduino Mega. Описано необхідні умови для вирощування рослин і представлена математична модель теплиці при управлінні температурою і вологістю повітря, яка розроблена для застосування в освітніх цілях з метою продемонструвати фізичні явища, що відбуваються в теплиці і принципи управління мікрокліматом. Описано етапи розробки та робота електронного пристрою автоматизованої системи управління мікрокліматом теплиці 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

Kozlov Volodymyr

Theme: Development of an information-measuring system for temperature and air humidity control based on a mathematical study of the microclimate of greenhouse premises.

Speciality: 123 " Computer Engineering ".

Institution: Luhansk Taras Shevchenko National University (LTSNU), 2024 year.

Master's work of: 67 p., 39 im, 34 sources.

Object of research is the greenhouses as an object of automation and improvement of automatic control systems.

Subject of research: a mathematical model of the microclimate and a system of automatic control of microclimate parameters in the greenhouse.

An aim of research is - development of an information and measurement system for controlling temperature and air humidity based on a mathematical study of the microclimate of greenhouse premises.

The results of the work An analysis of the existing form of the device used to control humidity and temperature in the greenhouse based on the Arduino Mega board was carried out. The necessary conditions for growing plants are described and a mathematical model of a greenhouse with temperature and humidity control is presented, which was developed for use in educational purposes in order to demonstrate the physical phenomena that occur in a greenhouse and the principles of microclimate management. The stages of development and operation of the electronic device of the automated greenhouse microclimate control system based on the mathematical study of the microclimate are described.

Conclusions. As a result of development in the Diptrace application, a printed circuit board was built, on the basis of which the device was designed. The device operation algorithm was presented in the form of a block diagram.

Keywords: MICROCONTROLLER, TEMPERATURE, DS18B20, AM2321, ATMEGA16.

ЗМІСТ

ВСТУП.....	5
РОЗДІЛ 1 АНАЛІЗ СИСТЕМИ УПРАВЛІННЯ	9
1.1. Необхідні умови для вирощування рослин.....	9
1.2. Аналіз сучасних засобів моніторингу температури і вологості	11
1.3. Аналіз системи управління мікрокліматом теплиці компанії "Альфа" на платі Arduino Mega".....	21
1.4. Побудова математичної моделі мікроклімату теплиці	29
1.5. Аналіз динамічних характеристик системи управління мікрокліматом теплиці	34
1.5. Висновки до розділу	40
РОЗДІЛ 2 ПОНЯТТЯ ПРО ТЕМПЕРАТУРУ ТА ВОЛОГІСТЬ. ПРИСТРОЇ ДЛЯ ВИМІРЮВАННЯ ТЕМПЕРАТУРИ І ВОЛОГOSTІ ...	42
2.1. Поняття про температуру і про температурні шкали	42
2.2. Пристрої для вимірювання температур.....	43
2.3. Пристрої для вимірювання вологості	46
2.4. Висновки до розділу	56
РОЗДІЛ 3 РОЗРОБКА ПРИНЦИПОВОЇ СХЕМИ ПРИСТРОЮ	58
3.1. Сімейство AVR	58
3.2. Архітектура ATmega16	60
3.3. Принцип роботи електронного пристрою.....	62
3.4. Розробка і виготовлення друкованої плати.....	65
3.5. Розробка програмного забезпечення	69
3.6. Алгоритм роботи пристрою.....	71
3.7. Висновки до розділу	72
ВИСНОВКИ	73
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	75
ДОДАТОК А.....	80

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. Тепличні 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 тепличних 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 теплицях 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.

В наш час багато уваги приділяється розробці та вивченню комп'ютеризованих інформаційно-вимірювальних систем, зокрема в агротехніці, і це становить широкий спектр наукових досліджень. Визначений внесок у розвиток теорії побудови вискоефективних інформаційно-вимірювальних систем здійснено видатними вітчизняними вченими та інженерами, такими як В.Б. Большаков, Є.Т. Володарський, А.А. Зорі, В.П. Лисенко, В.О. Поджаренко, В.А. Порєв, Ю.М. Туз та інші, а їхній внесок продовжує розвивати нове покоління дослідників.

Також значний вклад у цю галузь внесли відомі закордонні вчені, такі як R. Ben Ali, A.J. Both, S.C. Mukhopadhyay, J.A. Sanchez-Molina, D. Schmoldt, J.K. Schueller, Z.M. Selivanova та інші. Їхні наукові праці сприяли сформуванню сучасного стану і напрямку подальших досліджень у сфері комп'ютеризованих систем моніторингу.

Враховуючи значний прогрес у цій галузі, важливо відзначити, що більшість наукових досліджень спрямовані на розробку апаратних і програмних рішень для вимірювання окремих параметрів. Навіть при якісному розв'язанні завдань, пов'язаних із розробкою програмно-апаратних засобів комп'ютеризованих систем, теорія комплексного моніторингу фізичних середовищ поки що перебуває на етапі формування. Основними аспектами, які потребують подальшого розвитку на основі існуючих теорій, є: урахування принципів системності та комплексності збору та обробки розподіленої вимірювальної інформації; удосконалення теоретичних засад вибору місць розташування сенсорів; розширення фізико-математичного апарату для екстраполяції результатів лабораторного моніторингу на реальні об'єкти; врахування взаємозв'язків між вимірюваними параметрами під час інтерпретації інтегрального стану середовища.

Виходячи з вищесказаного, розробка інформаційно-вимірювальної системи контролю температури і вологості повітря на основі математичного дослідження мікроклімату тепличних приміщень вельми актуальна.

Об'єктом дослідження є теплиці як об'єкт автоматизації та вдосконалення систем автоматичного керування.

Предмет дослідження – математична модель мікроклімату і система автоматичного контролю параметрів мікроклімату в теплиці.

Мета роботи – розробка інформаційно-вимірювальної системи контролю температури і вологості повітря на основі математичного дослідження мікроклімату тепличних приміщень.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

1. розглянути необхідні умови для вирощування рослин;
2. розглянути і проаналізувати способи вимірювання температури і вологості;
3. дослідити математичну модель мікроклімату в теплиці;
4. розглянути і проаналізувати динамічні характеристики системи управління мікрокліматом теплиці;
5. розробити принципову схему і сконструювати пристрій;
6. 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 теплиці на основі плати Arduino Mega. Описані необхідні умови для вирощування рослин і представлена математична модель теплиці при управлінні температурою і вологістю повітря, яка розроблена для застосування в освітніх цілях з метою продемонструвати фізичні явища, що відбуваються в теплиці і принципи управління мікрокліматом.

У другому розділі проаналізувавши способи вимірювання температури було визначено, що рідинні скляні термометри є надійними приладами з високою точністю вимірів. 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.

Третій розділ присвячений створенню пристрою. У додатку Diptrace була побудована схема друкованої плати, на базі якої було сконструйовано пристрій. Так само в розділі описано середовище розробки, в якому

розроблявся програмний додаток для пристрою. Був представлений алгоритм роботи пристрою у вигляді блок-схеми.

РОЗДІЛ 1

АНАЛІЗ СИСТЕМИ УПРАВЛІННЯ

1.1. Необхідні умови для вирощування рослин

Умови, які є важливими для зростання та розвитку рослин, включають тепло, світло, повітря, воду та харчування. Кожен з цих факторів необхідний і виконує конкретні функції у 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 температури на насіння можна прискорити розвиток рослин і збільшити урожай. На підставі цього наукою розроблені, а практикою широко використовуються спеціальні рекомендації по прогріванню, пророщуванню, загартовуванню насіння, тривалість і температура яких різні, а також залежать від культур.

Рослини потребують тепла протягом усіх етапів свого росту та розвитку. Вимоги до теплового режиму різних культур варіюються і залежать від їх походження, виду, біологічних особливостей, фази розвитку та віку рослини.

Насіння теплолюбних культур проростають при температурі вище 10°C. Такі рослини не переносять заморозків і тривалого похолодання. При температурі нижче 10-12 °C їх ріст і розвиток припиняються, вони слабшають і швидше уражаються грибними і бактеріальними хворобами. При більш низьких температурах вони вмирають. Найбільш відповідна температура для росту, розвитку і плодоношення теплолюбних культур перевищує 20 °C [3,4].

Насіння морозостійких рослин проростають при температурі нижче 10 °C. Температура в діапазоні 17-20 °C є найбільш сприятливою для розвитку і плодоношення рослин цієї групи. Хоча зростання морозостійких культур продовжується при зниженні температури, тривалий вплив низьких температур (2-0 °C) може спричинити передчасне викидання квітконосів у багатьох рослин, не утворюючи ні повноцінного врожаю, ні насіння [5,6].

Основним джерелом світла є сонце, при якому рослини перетворюють воду та вуглекислий газ на складні органічні сполуки. Тривалість освітлення

має значущий вплив на ріст і розвиток рослин, при цьому вимоги до умов освітлення різняться. Деякі рослини, визначені як короткоденні, вимагають менше 12 годин світла на день, тоді як інші, довгоденні рослини, потребують більше 12 годин.

Волога є важливою для рослин протягом їхнього життя, і це стосується як ґрунту, так і повітря. Вода разом з теплом відіграє важливу роль у пробудженні насіння та підтриманні життєдіяльності рослин. Основна участь вода бере у формуванні органічних речовин і їхньому транспорту по всій рослині.

Рослини виявляють високу вимогливість до вологості, особливо це помітно в змісті води у фруктах, який коливається від 65% до 97% в залежності від виду. Велика випаровуваність з поверхні листя також пояснюється цією вимогою. Важливо, щоб вміст води в тканинах листа складав не менше 90-95%, оскільки інакше може відбутися в'янення та порушення їхньої функціональності.

Рослини забезпечують себе необхідним вуглекислим газом з повітря, який є єдиним джерелом вуглецевого живлення. У повітрі вміст цього газу дуже низький - всього 0,03%. Збагачення повітря вуглекислим газом відбувається головним чином через його виділення з ґрунту.

Для нормального росту і розвитку рослин важливі різноманітні харчові елементи.

Давайте розглянемо випадок вирощування цитрусових рослин. Для створення цитрусового саду потрібно вкласти певні зусилля та мати терпіння.

Догляд за цитрусовими рослинами вимагає уваги до деталей. Взимку не рекомендується розміщувати контейнер біля вікна, оскільки холод виходитиме від скла, а тепло - від батареї, що може негативно вплинути на здоров'я рослини. Найкращим варіантом для вирощування є тепле вологе місце з достатнім освітленням. У весняний та літній періоди рослину слід розміщувати ближче до вікна, спрямованого на південний захід або південний схід.

Температурний режим також має значення. У теплий період оптимальна температура становить 19-27 градусів, а взимку можна знизити її до 12-17 градусів. Оскільки цитрусові реагують негативно на різкі зміни температури, важливо взяти рослину в тепле приміщення перед початком морозів восени.

Зважаючи на те, що цитрусові рослини походять з тропіків, важливо дбати про вологість повітря. Якщо повітря недостатньо вологе, рекомендується використовувати зволожувач.

Освітлення грає ключову роль у вирощуванні цитрусових. Необхідно забезпечити достатнє освітлення цитрусового саду, уникаючи прямих сонячних променів влітку. У осінні та зимові періоди, коли недостатньо природного світла, необхідно забезпечити штучне освітлення протягом 12 годин на день для успішного вирощування врожаю.

1.2. Аналіз сучасних засобів моніторингу температури і вологості

В наш час теплиці широко використовуються повсюди через непередбачувані погодно-кліматичні умови з великими і різкими коливаннями температури та вологості. Вирощування сільськогосподарської продукції на відкритих ґрунтах у таких умовах стає проблематичним. Для уникнення цих труднощів часто використовують різні апарати для підтримання оптимальних кліматичних умов.

Сучасні теплиці, як об'єкти управління, характеризуються низькою динамікою та непостійністю параметрів, обумовлених особливостями технологічних процесів. У той час як агротехнічні норми вимагають високої точності стабілізації температури (± 1 градус), а також оперативної зміни її відповідно до фотосинтетичного опромінення та фаз розвитку рослин. Всі ці умови ставлять високі вимоги до функціонування та технічного вдосконалення обладнання програмно-апаратного забезпечення [37].

Автоматизація системи управління мікрокліматом у міні-теплиці може призвести до важливих результатів: економії від 15 до 25% необхідного тепла, покращення умов праці, підвищення загальної культури виробництва, створення чітких рамок регулювання мікрокліматичними умовами міні-

оранжереї, точного забезпечення подачі корисних речовин овочам, що призводить до збільшення їх врожайності. Для забезпечення високої продуктивності тепличних господарств потрібно утримувати багато параметрів у визначених межах або на певному рівні. Серед них основні: обігрів ґрунту, обігрів повітря в середині теплиці, концентрація CO₂, циркуляція повітря в теплиці, освітленість, вологість та вентиляція [37].

На ринку існує різноманіття обладнання для автоматизації цих процесів в теплиці, представлене різними компаніями, такими як "ICP DAS", ТОВ "ФІТО", "ОВЕН" і "ЕКФ" і т. п. Кожна з них має свої характеристики для забезпечення необхідних параметрів в теплицях [37]. Наприклад, "ЕКФ" є однією з передових компаній у галузі електротехніки, пропонуючи широкий асортимент низьковольтних та високоякісних продуктів, включаючи пристрій для контролю мікроклімату, зображений на рисунку 1.1.



Рис. 1.1. Вигляд пристрою управління мікрокліматом

Цей пристрій призначений для моніторингу та управління основними параметрами теплиці, гроубокса, оранжереї тощо. За його допомогою можна регулювати температуру, освітлення, вентиляцію та полив [37]. Пульт для управління мікрокліматом може працювати в автоматичному режимі,

підтримуючи встановлені вентиляційні та температурні режими, включаючи управління нагрівачами або вентиляторами, різними режимами освітлення, а також програмами для поливу та повітряних насосів.

Основні переваги використання цієї системи включають:

- простоту в установці;
- гнучкість цієї системи;
- легкість налаштування та експлуатації;
- можливість використовувати резервне живлення для збереження налаштувань системи у разі відключення електроживлення;
- вологозахист.

Очевидно, що система управління має свої переваги, такі як можливість вибору блоку з потрібними налаштуваннями [37]. Крім того, у компанії ЕКФ є 8-канальний пульт керування мікрокліматом теплиці, який базується на мікроконтролері DS 1820, його зображено на рисунку 1.2.



Рис. 1.2. Вигляд пульта управління мікрокліматом

Пристрій може регулювати вологість та атмосферу або ґрунт в міні теплиці, вмикати та вимикати насос, підігрівати воду, здійснювати полив та провітрювання повітря в теплиці. Управління навантаженням відбувається за допомогою таймера - ввімкнення навантаження протягом визначеного періоду

часу, а також забезпечення контролю мікроклімату через налаштування температури (термостатичний режим).

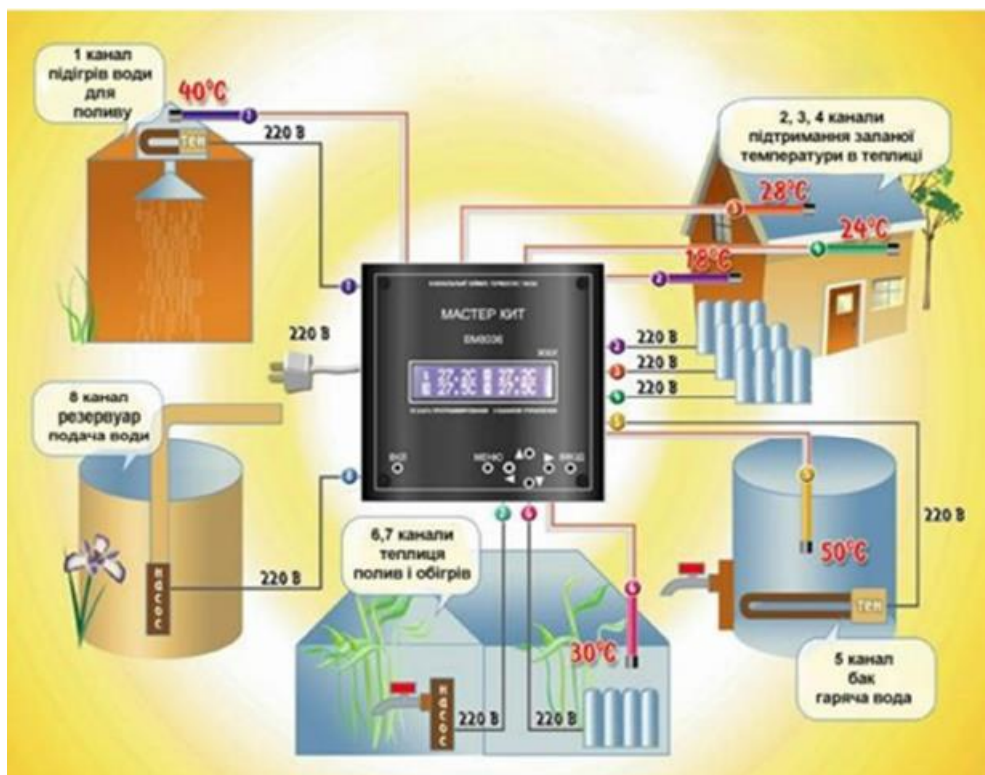


Рис. 1.3. Блок-схема взаємодії з теплицею

На рисунку 1.3 зображено схема роботи системи управління мікрокліматом теплиці в цілому. Насос, який подає воду до баку для поливу, підключений до блоку. Вода в баці нагрівається до необхідної температури і використовується для поливання за потреби. Опалення реалізоване за допомогою електричних обігрівачів, які також можна підключати для забезпечення температурного режиму ґрунту і повітря [37].

Для зручного підключення датчиків вологості, силових виходів та температурних датчиків, а також інтерфейсу RS 232 (COM-порт ПК), в пульті автоматичного керування передбачені конкретні клемні роз'єми. Енергія надходить зі спеціального роз'єму на адаптері з напругою 9-12 В [37].

Технічна специфікація системи управління мікрокліматом включає:

1. Кількість процесів управління: 8;

2. Режими таймера: ввімкнення навантаження у заданому інтервалі часу; керування навантаженням у певні дні тижня, дні місяця або кілька місяців;
3. Режими термостата: керування як нагрівач; керування як охолоджувач;
4. Вбудований таймер із світлом і звуком (підсвічення дисплею): входи для підключення датчиків;
5. Цифровий вхід для підключення до 32 датчиків;
6. Аналогові входи: 2;
7. Енергонезалежний годинник з реальним часом та календарем до 2100 року;
8. Зберігання налаштувань у енергонезалежній пам'яті; продовження роботи програми при тимчасовому відключенні від мережі;
9. Інші виходи: 8 оптоізольованих 7-канальних 2-амперних каскадів з можливістю підключення силових 7-канальних для управління більш потужними навантаженнями (понад 2 А);
10. Віддалене управління термостатом через СОМ-порт комп'ютера за допомогою конкретного програмного забезпечення [37].

Кількість налаштувань, які можна контролювати за допомогою цього модуля, дозволяє його ефективно використовувати в міні-теплицях. Його можна вважати одним із найкращих у своєму роді, з урахуванням його невеликої вартості та простоти в експлуатації. Однак слід зазначити, що програмування модуля може бути складним завданням через синтаксис певних мов програмування [37].

З метою кращої реалізації автоматичного управління, компанія "ICP DAS" рекомендує використання інтелектуального модуля введення/виведення WISE-7118 Z. Зовнішній вигляд цього розумного пристрою автоматизації представлений на рисунку 1.4.



Рис. 1.4. Повний вигляд розумного модуля введення/виведення WISE-7118Z

Цей модуль має 6 цифрових виходів та 10 аналогових входів. Користувач може налаштовувати канали в широкому діапазоні напруги і струму, а також використовувати різні типи термічних пар для вимірювання параметрів мікроклімату в теплиці, таких як вологість чи температура [37].

За допомогою цифрових виходів можна керувати обігрівачем, люком-вентиляцією, поливом, освітленням та системою випарного охолодження. Інтелектуальний пульт WISE-7118 Z постійно відслідковує стан мікроклімату в теплиці і, в залежності від змін у параметрах освітлення, ґрунту та повітря, використовує відповідне керування [37].

Крім того, розумний модуль WISE використовує протокол передачі даних Modbus TCP Slave. Це дозволяє об'єднати ці системи у єдиний диспетчерський вузол, де, використовуючи систему SCADA, користувач може керувати всім процесом і отримувати точну інформацію про стан мікроклімату в кожній теплиці. На рисунку 1.5 представлена блок-схема з'єднання інтелектуального пристрою з обладнанням у теплиці [37].

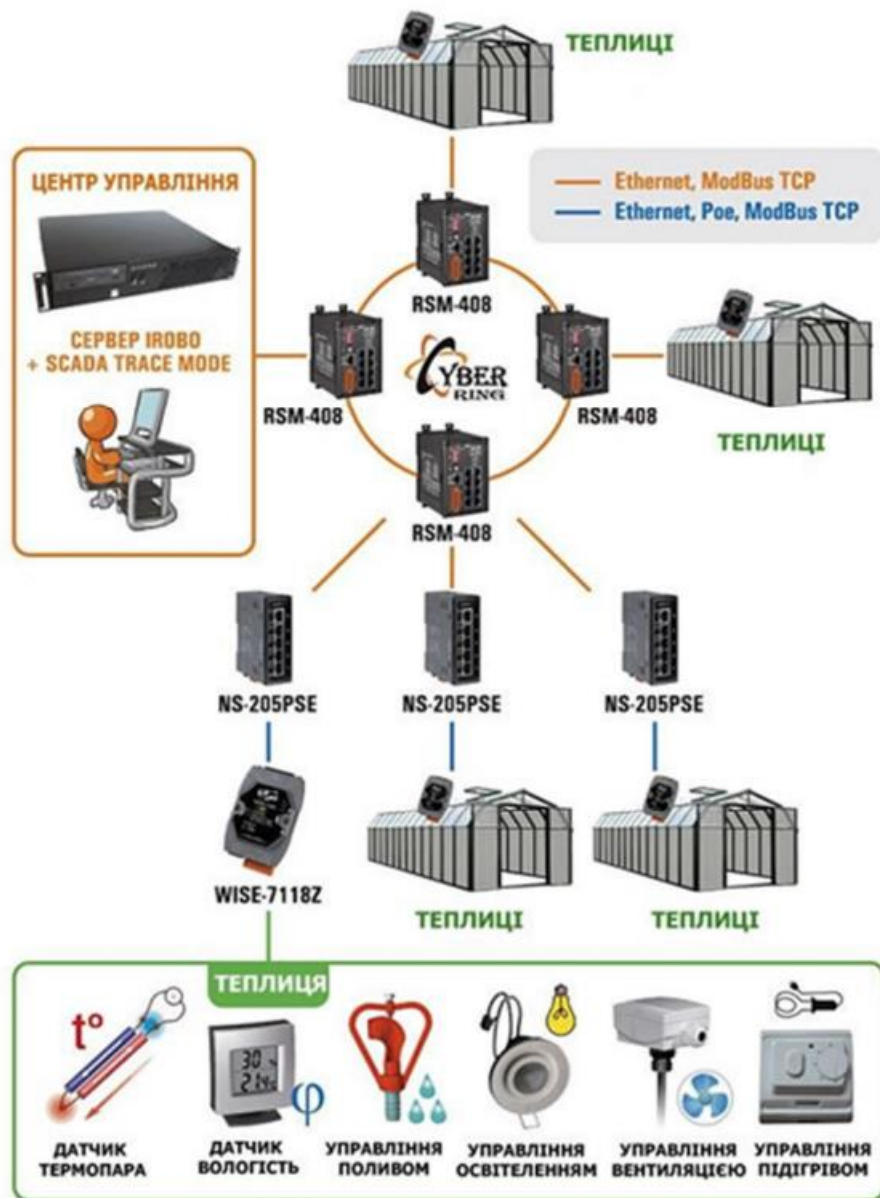


Рис. 1.5. Блок-схема керування мікрокліматом в теплицях

Переваги використання WISE включають в себе використання логіки IF-THEN-ELSE для створення конфігурації керування у модулі. Користувач може визначити до 36 правил IF-THEN-ELSE, які, коли завантажені на модуль, виконують певні рішення при виникненні певної умови. Налаштування всіх необхідних дій формується у звичайному веб-браузері всього за кілька хвилин. Крім того:

- підтримує Modbus TCP для інтеграції у систему SCADA;

- WISE-7118 Z використовує технологію POE живлення, що усуває необхідність прокладання кабельних ліній для отримання живлення.

Компанія "ФІТО" пропонує системи керування мікрокліматом у теплицях серії FC, а повний вигляд цього мікрокліматичного комп'ютера від цієї фірми зображено на рисунку 1.6 [37].



Рис. 1.6. Повний вигляд мікрокліматичного комп'ютера

Це є оптимальним рішенням для різних типів теплиць, незалежно від їхніх розмірів, будь то скляні чи плівкові. Для керування основними параметрами теплиці встановлюють виконавчі системи, такі як вентиляція, обігрів, освітлення та додавання CO₂. Суворе дотримання ключових показників клімату є гарантією високого врожаю та стійкості рослин до захворювань. Однак сучасні технології вирощування вимагають розширеного набору показників мікроклімату, таких як вологість листа, температура листа, розподіл температури у повітрі вздовж вертикального профілю теплиць та швидкість руху повітря. Компанія "ФІТО" розробляє клімат-комп'ютери, які

не лише контролюють основні показники клімату, але й дозволяють регулювати цей розширений спектр показників [37].

Архітектура клімат-комп'ютера забезпечує повноцінне автоматизоване управління різними системами керування теплицями, з дотриманням конкретного агрономного режиму. Зростання цін на енергоносії вимагає не лише забезпечення необхідного мікроклімату, але й раціонального використання ресурсів, чи то увімкнення систем опалення чи освітлення, подача CO₂ чи регулювання вентиляції. Функціональні можливості систем керування дозволяють розробляти "стратегії управління", де агроном може вибирати пріоритет економії енергії чи найбільш точного дотримання технології, враховуючи фазу росту рослини чи економічну доцільність [37].

Принцип роботи клімат-комп'ютера базується на використанні промислового контролера, спеціально розробленого для всіх типів теплиць фірмою "ФІТО". Завдяки сучасній елементній базі з японських і американських компонентів, ці контролери володіють високою надійністю та ефективністю. Крім контролера, система керування кліматом включає в себе підсистему встановлених всередині теплиці вимірювальних датчиків [37].

У випадку потреби система може автоматично інтегруватися з котельнею. Для цього передбачений конкретний модуль, який через інтерфейс FIDUFACE передає дані до котельні для управління виробництвом тепла, CO₂ або електроенергії. Моніторинг кліматичного процесу та внесення корекцій у завдання можна виконати за допомогою звичайного комп'ютера. Функція віддаленого керування системою через Інтернет також доступна [37].

Основні можливості системи включають:

- вимірювання параметрів клімату в декількох зонах;
- автоматичний контроль опалення, вентиляції, зашторювання, CO₂, освітлення;
- створення оптимальної стратегії управління;
- інтеграція з системою управління котельнею (FIDUFACE);
- зручний інтерфейс;

- функція економії енергетичних ресурсів;
- віддалений моніторинг і аналіз з ПК [38].

Отже, оглянувши усі зазначені засоби для утримання мікроклімату в теплицях, можна зауважити, що на сьогоднішній день вибір є досить різноманітним, і користувач може обирати параметри, які відповідають його вимогам. Усі засоби володіють великим потенціалом [38]. Пульт управління мікрокліматом компанії "ЕКФ" рекомендується для застосування в невеликих теплицях. Хоча пульт може виконувати функції, такі як контроль освітленості, полив, регулювання температури та вентиляції теплиці, кількість параметрів є середньою порівняно з його аналогами [38].

Інший виріб від цієї компанії - це блок управління мікрокліматом, що базується на мікроконтролері DS1820. Подібно до його попередника, він відрізняється компактними розмірами, простотою експлуатації та можливістю регулювання різних параметрів залежно від підключених датчиків. Здатність підключити до 32 датчиків дозволяє повністю автоматизувати процес регулювання мікроклімату на одному модулі. Вісім виходів дозволяють підключити пристрої для управління потужнішими навантаженнями. Можливість підключення до комп'ютера надає користувачу можливість в реальному часі відслідковувати всі параметри і швидко реагувати в разі потреби. Цей блок може бути використаний як для невеликих, так і для великих теплиць [38].

Найбільш передовим, на наш погляд, є інтелектуальний модуль WISE-7118Z. Незважаючи на його компактні розміри, він може керувати значною кількістю параметрів, маючи при цьому додаткові можливості. Зокрема, він забезпечує постійний моніторинг мікроклімату в теплиці та підтримує режим передачі Modbus TCP, що дозволяє збирати дані про стан теплиці та відображати їх на моніторі користувача. Це особливо корисно при контролі кількох теплиць одночасно. Хоча кількість підключених датчиків обмежена 10 одиницями, використання комутаторів, таких як RSM-408 і NS-205PSE, дозволяє розширити кількість підключених модулів. Дані передаються на

сервер і обробляються в SCADA системі. Також передбачено логічне управління подіями, що забезпечує додаткові можливості мікроконтролера [38].

Останнім прикладом, який ми розглядали, був кліматичний комп'ютер від компанії "ФІТО". Незважаючи на його значні розміри, він вражає потужністю контролю параметрів. Цей пристрій ідеально підходить для теплиць різних розмірів, контролює основні параметри і володіє додатковими функціями, такими як моніторинг температури та вологості листка, регулювання розподілу температури по всій теплиці і т. д. Кліматичний комп'ютер виявляється економічно вигідним, а його архітектура мінімізує необхідність людського втручання в роботу системи [38].

З нашого погляду, закордонні виробники пропонують більш вдосконалені та надійні продукти, тому, ймовірно, вони мають вищу якість порівняно з вітчизняними виробниками. Якщо фінансові можливості дозволяють, то більш раціональним вибором буде використання кліматичного комп'ютера від "ФІТО" або інтелектуального модуля WISE-7118Z. Ці дві опції призначені як для великих, так і для середніх теплиць і повністю відповідають всім основним вимогам до автоматизації мікроклімату в теплиці [38].

У свою чергу, пульт управління мікрокліматом від "ЕКФ" і блок управління мікрокліматом на мікроконтролері DS1820 краще використовувати для невеликих і середніх теплиць.

1.3. Аналіз системи управління мікрокліматом теплиці компанії "Альфа" на платі Arduino Mega"

Існують різні способи зберігання рослин і плодів: в закритих або відкритих контейнерах. Вибір способу вирощування та зберігання залежить від виду оранжерей або теплиць, стану 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 теплицях (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 теплиці, довжиною 6 метрів, шириною 3 метри, висотою 2 метри. В теплиці є одні двері і 2 квартирки, проведені електрика та водогін. Нагрівання води відбувається в ємності об'ємом 70 літрів. Тиск в ємності складає близько двох атмосфер. В теплиці вирощується близько 35 рослин [10] Система має наступний вигляд:

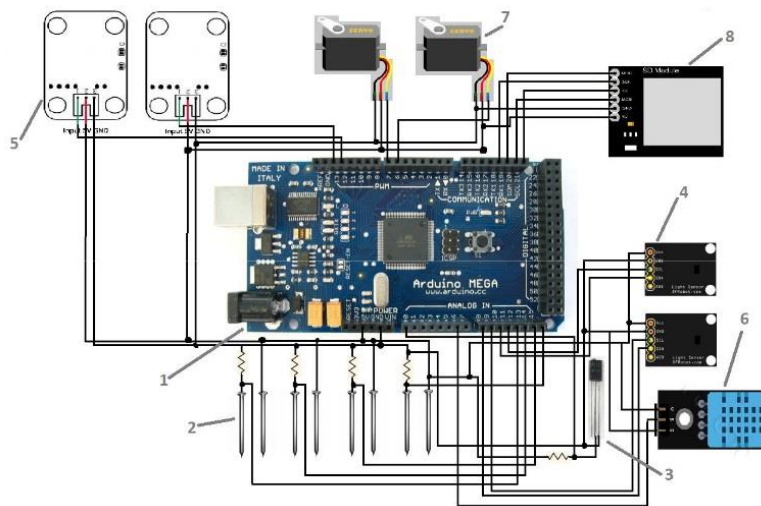


Рис. 1.7. Схема системи управління мікрокліматом в теплиці

Платформа Arduino Mega займає ключове положення у цій системі. (рис. 1.8):



Рис. 1.8. Arduino Mega

Arduino - це повністю відкрита система, що включає в себе плату та середовище розробки, основані на модифікованій версії мови програмування 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 живлення 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 / виведення;
- 16 аналогових входів;
- струм споживання на одному виводі: до 40 мА;
- струм споживання виведення 3.3В: 50 мА;
- пам'ять Flash Memory: 128 КВ, з яких 4КВ використовуються завантажувачем;
- ОЗУ: 8 КВ;
- незалежна пам'ять: 4 КВ;
- тактова частота: 16 МГц;

- розмір: 75x54x15 мм;
- вага: 45 г.

До Arduino Mega підключені необхідні датчики і модулі.

Включення / вимкнення 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 аналоговим сигналом можна судити про ступінь вологості. Схема датчика представлена на рисунку:

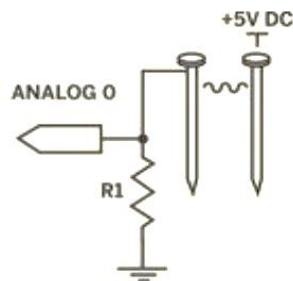


Рис. 1.9. Датчик вологості ґрунту

Для вимірювання температури води використовується LM335Z - аналоговий термодатчик термостабілітрон:



Рис. 1.10. Аналоговий термодатчик LM335Z

Використовуваний датчик має такі характеристики:

- діапазон: $-40 \dots + 100$;
- точність: $1\text{ }^{\circ}\text{C}$;
- залежність: $10\text{ мВ} / ^{\circ}\text{C}$.

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, яка в десятках мВ представляє абсолютну температуру в градусах Кельвіна. Схема підключення представлена на рисунку 1.11:

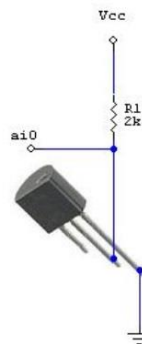


Рис. 1.11. Схема підключення термодатчика

Для того, щоб полив включався тільки в темний час доби, використовуються 2 датчики світла Light Sensor-BH1750 (див. рис. 1.12):

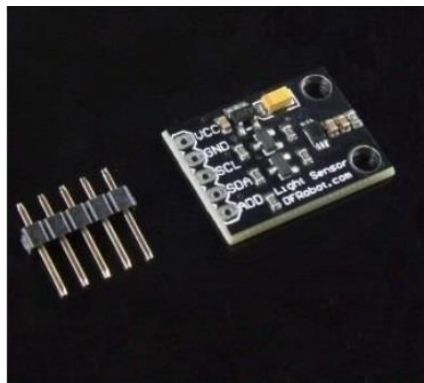


Рис. 1.12. Датчик світла Light Sensor-BH1750

Даний датчик служить для вимірювання освітленості в межах від 1 до 65535 люкс. Він має наступні характеристики:

- напруга живлення: $3\text{--}5\text{ В}$;
- дозвіл: 16 біт ;
- габарити: $19 \times 14 \times 3\text{ мм}$;

- похибка: $\pm 20\%$.

Підключення датчика здійснюється наступним чином, як показано на рисунку 1.13:

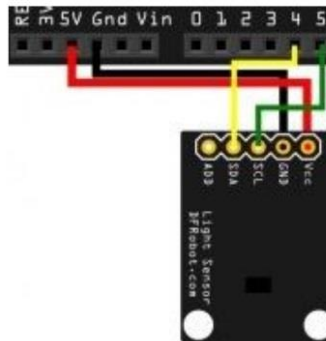


Рис. 1.13. Підключення датчика світла Light Sensor-BH1750

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 світлодіоду. Цей модуль управляється за допомогою цифрового порту вводу-виводу. Час перемикання контакту становить 10 мс. Як і датчики для вимірювання температури і вологості ґрунту, релейний модуль підключається в керуючій електроніці через три дроти:

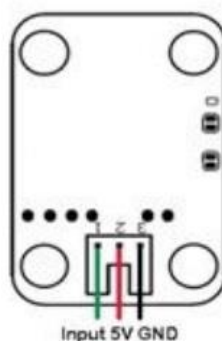


Рис. 1.14. Призначення контактів роз'єму релейного модуля

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 $\pm 5\%$.

Для регулювання температури повітря в теплиці використовується два режими: пасивне і активне провітрювання. Пасивне провітрювання це відкриття / закриття кватирок, а активне - вмикання / 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 x 19мм x 43 мм;
- вага: 55 м

Підключення вентилятора проводиться таким же чином, 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 мікроконтролерами;
- напруга живлення: 5 В;

- розмір: 36 x 30 x 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 теплиці

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 теплиці є розробка моделей об'єкта, що відбивають процеси, що відбуваються в ній, 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 "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 часі ґрунтується на наступних спрощених припущеннях:

- 1) Модель трактує оранжерею як об'єм повітря, обмежений стінами, дахом і підставою (землею), не враховуючи просторового розподілу змінних, що характеризують мікроклімат. Це припущення виправдовується наявністю постійної конвекції повітря і примусової вентиляції.

- 2) Зміна біомаси рослин у процесі їх розвитку розглядається як зовнішній фактор, непов'язаний з параметрами мікроклімату. Це припущення виправдовується тим, що система управління підтримує мікроклімат згідно з технологічними вимогами, а біомаса рослин вважається постійною.

- 3) Зміни в параметрах моделі в часі відбуваються настільки повільно, що їх можна розглядати як постійні при аналізі динамічних процесів в системі

управління. Мікрокліматична модель розглядається як детермінована і зосереджена на врахуванні взаємозв'язків масового і теплового балансів з урахуванням стадій біологічного розвитку рослин (форма обліку - зміна параметрів моделі в часі) [13,14].

У моделі виділяються дві підсистеми: повітря теплиці та навколишній шар із рослинами та ґрунтом. При описі цих підсистем використовуються такі змінні, як температура повітря в оранжереї, температура рослин в оранжереї (температури навколишнього ґранту), вологість (відносна або абсолютна) повітря в теплиці[15].

Рівняння масового балансу води в оранжереї має вигляд:

$$\rho \cdot V \cdot \frac{dX(t)}{dt} = F(t) + C_{sat}(t) \cdot [E(t) + fog(t)] \quad (1.1)$$

де ρ - щільність повітря в теплиці, (кг / м³);

V - об'єм повітря в теплиці, (м³);

$X(t)$ - абсолютна вологість в теплиці, (кгвода / кгвоздух);

t - час, (с);

$F(t)$ - витрати 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).

Вираз (1.1) вказує на те, що зміна обсягу води в теплиці внутрішнього повітря залежить від таких складових:

- кількості води, що потрапляє з зовнішнього повітря через віконні щілини - $F(t)$;
- кількості води, що випаровується рослинами в процесі свого зростання - $E(t)$;
- кількості води, що розпилюється системою туману - $fog(t)$ [16].

Рівняння енергетичного теплового балансу, що визначає вплив на зміну температури внутрішнього повітря оранжереї, представлено таким чином:

$$\rho \cdot V \cdot C_v \cdot \frac{dT(t)}{dt} = Q_s(t) - Q_{cc}(t) + Q_p(t) - C_{sat}(t) \cdot (Q_u(t) + Q_t(t)) - Q_v(t) + W(t) \quad , (1.2)$$

де ρ - щільність повітря в теплиці, (кг / м³);

V - об'єм повітря в теплиці, (м³);

C_v - теплоємність повітря, [Дж / (кг · ° C)];

$T(t)$ - температура повітря всередині теплиці, (° C);

t - час, (с);

$Q_s(t)$ - сонячна енергія, що передається повітрю в теплиці, (Вт);

$Q_{cc}(t)$ - енергія обміну при провідності і конвекції, (Вт);

$Q_p(t)$ - обмін енергією з рослинами, (Вт);

$C_{sat}(t)$ - коефіцієнт насичення повітря;

$Q_u(t)$ - втрати енергії за рахунок сумарного випаровування води рослинами, (Вт);

$Q_t(t)$ - втрати енергії за рахунок розпилення води системою туману, (Вт);

$Q_v(t)$ - енергія обміну повітряної вентиляції, (Вт);

$W(t)$ - енергія системи обігріву, (Вт).

Співвідношення (1.2) показує, що зміна температури внутрішнього повітря в теплиці залежить від наступних складових:

- сонячної енергії, що нагріває внутрішнє повітря в теплиці - $Q_s(t)$;
- температури повітря навколишнього середовища, що надходить в теплицю через систему вентиляційних фрамуг і щілини - $Q_v(t)$;
- температура оточуючого середовища повітря, що впливає на передачу енергії через стіни шляхом провідності і конвекції - $Q_{cc}(t)$;
- загальна витрата води рослинами, що викликає охолодження повітря в теплиці - $Q_u(t)$;
- енергії системи обігріву - $W(t)$;
- втрат енергії від розпилення води системою туману - $Q_t(t)$;

- теплообміну з рослинами - $Q_p(t)$.

Рівняння теплового балансу енергії, яке визначає вплив на зміну температури рослин в теплиці, має вигляд:

$$S_{outside} \cdot C_p \cdot \frac{dT_p(t)}{dt} = Q_{ur}(t) - Q_p(t) - Q_g(t), \quad (1.3)$$

де $S_{outside}$ - площа поверхні теплиці, (м²);

C_p - теплоємність рослин, [Дж / (° С · м²)];

$T_p(t)$ - температура рослин всередині в теплиці, (° С);

t - час, (с);

$Q_{ur}(t)$ - енергія, засвоюється рослинами протягом дня, (Вт);

$Q_p(t)$ - обмін енергією з рослинами, (Вт);

$Q_g(t)$ - втрати енергії через ґрунт, (Вт).

Співвідношення (1.3) показує (1.4), що зміна температури рослин в теплиці залежить від наступних складових:

- енергії, засвоюваній рослинами - $Q_{ur}(t)$;
- втрат енергії через ґрунт - $Q_g(t)$;
- теплообміну повітря з рослинами - $Q_p(t)$.

Таким чином, система рівнянь (1.4), що наближено описує мікроклімат в теплиці, має вигляд:

$$\begin{cases} \rho \cdot V \cdot \frac{dX(t)}{dt} = F(t) + C_{sat}(t) \cdot (E(t) + fog(t)) \\ \rho \cdot V \cdot C_v \cdot \frac{dT(t)}{dt} = Q_s(t) - Q_{ca}(t) + Q_p(t) - C_{sat}(t) \cdot (Q_u(t) + Q_r(t)) - Q_v(t) + W(t) \\ S_{outside} \cdot C_p \cdot \frac{dT_p(t)}{dt} = Q_{ur}(t) - Q_p(t) - Q_g(t) \end{cases}, \quad (1.4)$$

Модель (1.4) є приблизною. У ній не враховано просторовий розподіл температури і вологості по площі теплиці, і розподіл цих змінних по висоті враховується тільки в агрегованому вигляді. Вплив температурно-вологісного режиму на показники розвитку рослин зведений до детермінованої залежності коефіцієнтів моделі від часу. Однак, враховуючи мету використання моделі (для аналізу і синтезу алгоритмів керування, а не для визначення технологічно

доцільних режимів зміни температури і вологості), представлена на рисунку 1.18 модель є прийнятною. [17]. Модель мікроклімату в теплиці

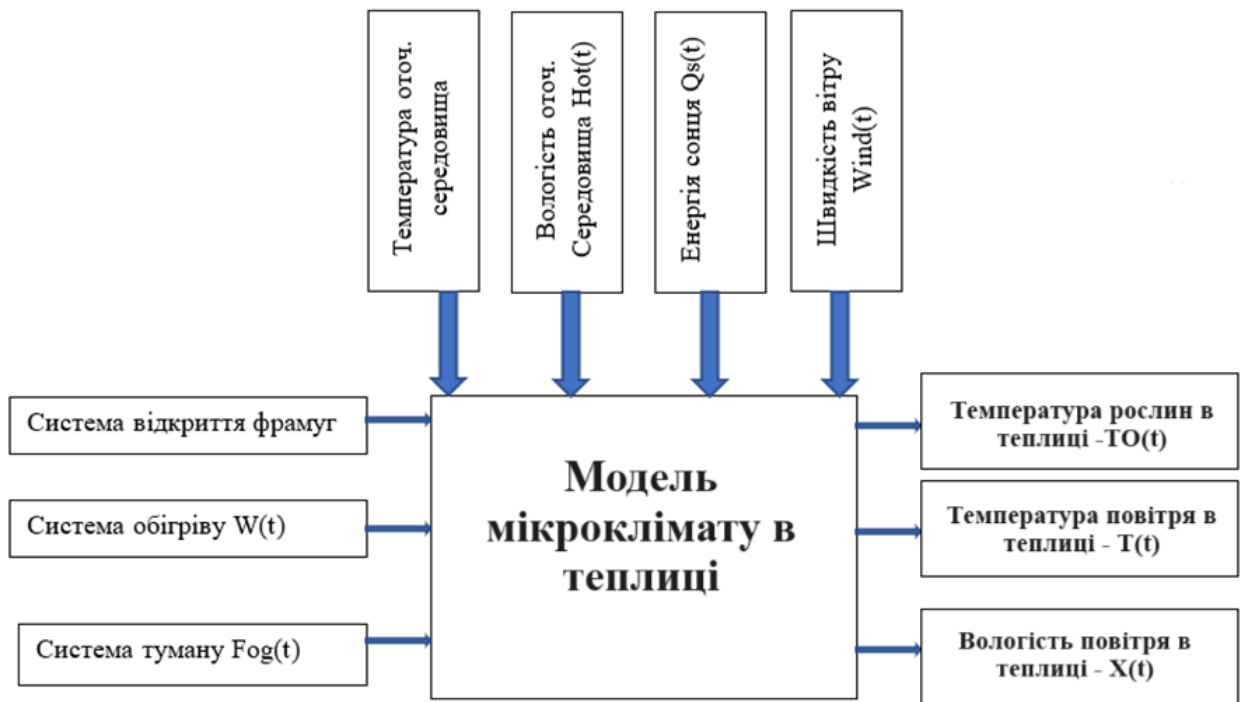


Рис. 1.18. Математична модель температурно-вологісного режиму теплиці

1.5. Аналіз динамічних характеристик системи управління мікрокліматом теплиці

Дослідження математичних моделей теплиці як об'єкта управління в останні десятиліття істотно просунулися.

Найбільш прості моделі, представлені у вигляді аперіодичних ланок першого або другого порядку з запізненням. Такі моделі є досить грубими і наближено 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 методів визначення математичних моделей є їхня інтуїтивна зрозумілість та можливість застосування традиційних методів синтезу регуляторів для управління параметрами мікроклімату.

З іншого боку, альтернативний підхід до визначення математичних моделей теплиці базується на детальному описі цієї складної динамічної системи, в якій параметри, такі як температура і вологість, взаємодіють за допомогою нелінійних законів термодинаміки. Цей підхід використовує модель у просторі станів такого формату:

$$\dot{x} = f(t, x, u, v), \quad (1.5)$$

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 CO_2 ;

v - зовнішні впливи, наприклад, температура і вологість зовнішнього повітря, сонячна радіація, вітровий вплив та ін.

Модель в просторі станів більш точно описує процеси, що відбуваються в теплиці, враховуючи взаємозв'язок змінних стану, зовнішніх збурень і сигналів управління, проте в свою чергу суттєво ускладнюється моделювання внаслідок нелінійності моделі.

Головною метою цього дослідження є вивчення математичної моделі теплиці при управлінні температурою і вологістю повітря, що ґрунтується на принципах збереження енергії та маси. Крім того, планується розробка моделі у середовищі Simulink з метою використання в освітніх цілях для ілюстрації фізичних явищ, що відбуваються в теплиці, а також принципів управління мікрокліматом.

$$\frac{dT_{\text{внутр}}(t)}{dt} = \frac{1}{\rho C_B V_m} [Q_H(t) + S_n(t) - \lambda Q_m(t)] - \left(\frac{v_B(t)}{V_m} + \frac{k_{\text{т.огор}}}{\rho C_B V_m} \right) [T_{\text{внутр}}(t) - T_{\text{зовн}}(t)], \quad (1.6)$$

$$\frac{d\varphi_{\text{внутр}}(t)}{dt} = \frac{1}{V_B} Q_m(t) + \frac{1}{V_B} [E(S_n(t), \varphi_{\text{внутр}}(t))] - \frac{v_B(t)}{V_m} [\varphi_{\text{внутр}}(t) - \varphi_{\text{зовн}}(t)], \quad (1.7)$$

$$E(S_n(t), \varphi_{\text{внутр}}(t)) = \alpha \frac{S_n(t)}{\lambda} - \beta \varphi_{\text{внутр}}(t), \quad (1.8)$$

де $T_{\text{внутр}}, T_{\text{зовн}}$ - температура повітря всередині і ззовні теплиці відповідно ($^{\circ}\text{C}$);

$\varphi_{\text{внутр}}, \varphi_{\text{зовн}}$ - відносна вологість повітря всередині і зовні теплиці відповідно (%);

$k_{\text{т.огор}}$ - коефіцієнт теплопередачі матеріалу огорожі теплиці ($\text{Вт} / \text{K}$);

V - повний геометричний об'єм теплиці (м^3);

$V_{\text{т}}, V_{\text{в}}$ - об'єми теплиці що обігріваються та зволожуються відповідно (м^3).

Зазвичай складають не більше 60% -70% загального обсягу теплиці;

ρ - щільність повітря ($1,2 \text{ кг} / \text{м}^3$);

$C_{\text{в}}$ - питома теплоємність повітря ($1,005 \text{ кДж} \cdot \text{кг}^{-1} \cdot \text{K}^{-1}$);

$Q_{\text{н}}$ - потужність обігрівачів повітря теплиці (Вт);

$Q_{\text{т}}$ - продуктивність системи туманоутворення (гвода / с);

$S_{\text{п}}(t)$ - поглинене теплицею сонячне випромінювання (Вт);

λ - питома теплота пароутворення ($2256 \text{ кДж} / \text{кг}$);

$v_B(t)$ - повітрообмін, який забезпечувався б системою вентиляції теплиці ($\text{м}^3 / \text{с}$);

$E, S_{\text{п}}(t), \text{внутр}(t)$ - евапотранспірації рослин, в функції від поглинання сонячного випромінювання і вологості повітря в теплиці (гвода / с);

α, β – 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 $\beta \varphi_{\text{внутр}}(t)$.

Підставляючи (3) в (1), після перетворень отримаємо

$$\frac{dT_{\text{внутр}}(t)}{dt} = \frac{1}{C_B} [Q_{\text{н}}^{\text{max}} \tilde{Q}_{\text{н}}(t) + S_{\text{н}}(t) - \lambda \tilde{Q}_{\text{м}}(t)] - \left(\frac{\tilde{v}_B(t)}{V_m} + \frac{k_{\text{т.огор}}}{\rho C_B V_m} \right) [T_{\text{внутр}}(t) - T_{\text{зовн}}(t)], \quad (1.9)$$

$$\frac{d\varphi_{\text{внутр}}(t)}{dt} = \frac{1}{V} \tilde{Q}_m(t) + \alpha' S_n(t) - \frac{\tilde{v}_v(t)}{T_v} [\varphi_{\text{внутр}}(t) - \varphi_{\text{зовн}}(t)], \quad (1.10)$$

де,

$$\tilde{Q}_n = \frac{Q_n}{Q_n^{\max}}, \quad \tilde{Q}_m = \frac{Q_m}{Q_m^{\max}}, \quad \tilde{v}_v = \frac{v_v}{v_v^{\max}} \quad - \text{нормалізовані управляючі змінні};$$

$$\lambda' = \lambda Q_m^{\max}, \quad V' = \frac{V_m}{Q_m^{\max}}, \quad \alpha' = \frac{\alpha}{V \lambda} \quad - \text{нормалізовані коефіцієнти};$$

$$T_v = \frac{V_v}{v_v^{\max}} \quad - \text{час, необхідний для повної заміни повітря у зволоженій}$$

частині теплиці.

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].

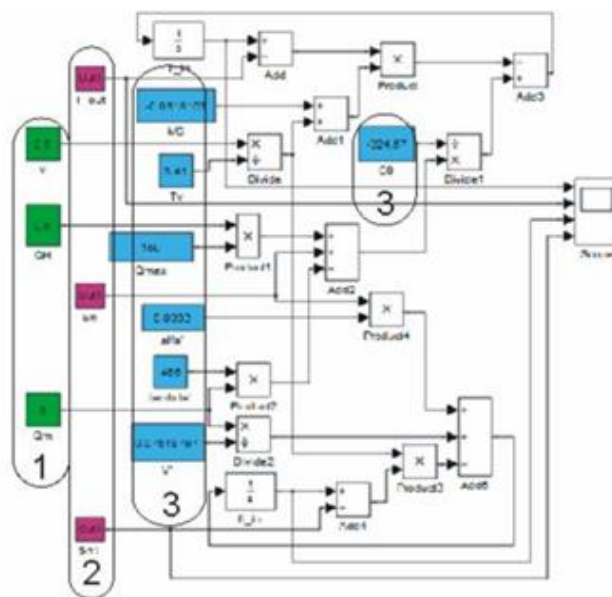


Рис. 1.19. Математична модель теплиці в середовищі Matlab-Simulink

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 теплиці. Решта блоки які призначені для виконання математичних операцій та для візуалізації результатів моделювання.

У цьому прикладі була промодельована теплиця на 100 м² висотою 2,5 м, з параметрами, представленими в таблиці 1. Всі параметри наведені до одного квадратного метра площі теплиці. Встановлені в блоках інтегрування початкові значення вологості і температури: 15 °С, і 50%.

Таблиця 1.1

Параметри моделі теплиці

Найменування	Значення
C_g	-324,67 хв·Вт/°С
$k_{m,огор}$	29,81Вт/°С
T_v	3 хв
λ'	460 Вт
α'	0.0033 г/(м ³ ·хв·Вт)
V	13,3 г/(м ³ ·хв)

Моделювання проводилося в середовищі Matlab / Simulink на базі операційної системи Windows 10. Для виконання інтеграції використовувався метод Рунге-Кутта четвертого порядку з фіксованим кроком в 1 хвилину. Діапазон інтегрування становив 1 доба або 1440 хвилин.

При моделюванні всі параметри математичної моделі теплиці 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.

$$T_{зовн}(t) = T_{cp} + T \sin(\omega t) + \delta(t) \quad (1.11)$$

де $T_{cp} = 25$ °С середня температура;

$T = 10$ °С - амплітуда коливань температури протягом доби;

$\omega = 2\pi / 1440$ частота гармонійної складової;

$\delta(t)$ - випадкова складова (білий шум).

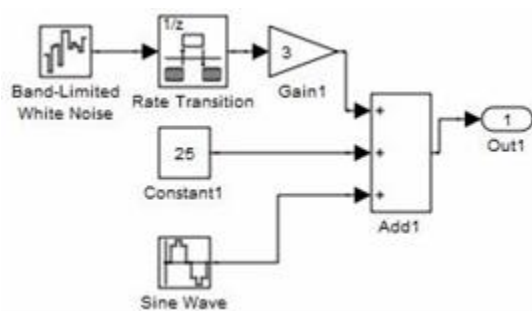


Рис. 1.20. Модель завдання обуреного впливу

Також було проведено моделювання коливань вологості та сонячного випромінювання протягом доби. Графічне відображення результатів моделювання подано на рисунку 1.21.

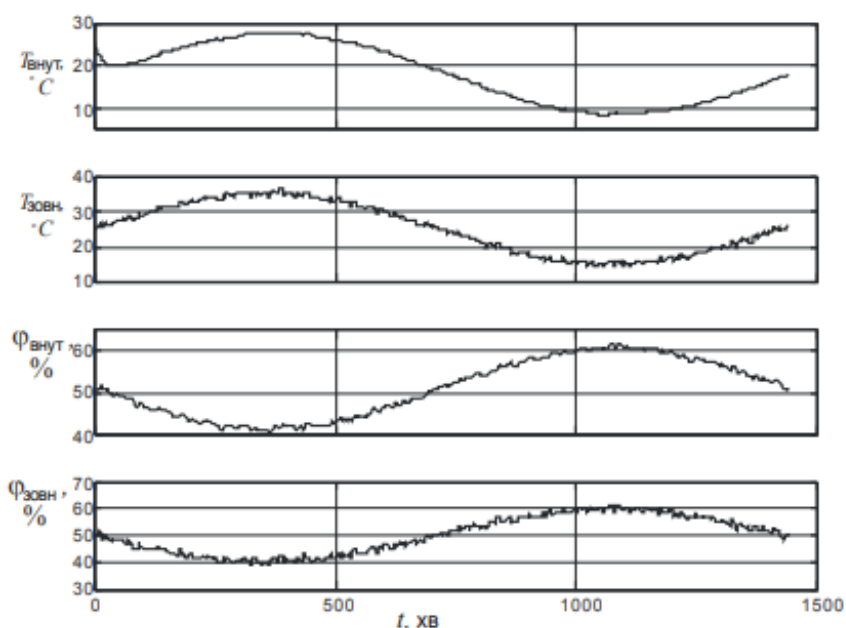


Рис. 1.21. Результати моделювання температури і вологості повітря всередині теплиці

1.5. Висновки до розділу

Перший розділ містить опис існуючого вигляду пристрою, використовуваного для контролю вологості і температури в теплиці на основі плати Arduino Mega.

Математична модель мікроклімату теплиці 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 теплиці 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 теплиці 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 2

ПОНЯТТЯ ПРО ТЕМПЕРАТУРУ ТА ВОЛОГІСТЬ.

ПРИСТРОЇ ДЛЯ ВИМІРЮВАННЯ ТЕМПЕРАТУРИ І ВОЛОГОСТІ

2.1. Поняття про температуру і про температурні шкали

Термін "температура" використовується для опису властивості тіла, що визначає його тепловий стан. Згідно з кінетичною теорією, температуру визначають як міру кінетичної енергії молекул у поступальному русі. Таким чином, температура є умовною статистичною величиною, яка пропорційна середній кінетичній енергії молекул тіла [22].

Усі запропоновані температурні шкали будувалися однаковим чином, за винятком деяких рідкісних випадків: для надання числових значень використовувалися щонайменше дві постійні точки. Передбачалося, що видимі термометричні властивості використовуваної в термометрі речовини лінійно пов'язані з температурою t :

$$t = kE + D, \quad (2.1)$$

де k - коефіцієнт пропорційності;

E - термометрична властивість;

D - постійна.

Визначаючи певні значення температур для двох постійних точок, можна розрахувати константи k , D і на їх основі побудувати температурну шкалу. Коефіцієнт k змінюється при зміні температури і є різним для різних термометричних речовин. Таким чином, термометри, що базуються на різних термометричних речовинах і мають рівномірну градацію, надають різні відліки при температурах, відмінних від температур постійних точок. Це особливо помітно при високих (температура кипіння води) і дуже низьких температурах.

Термодинамічна шкала ідентична шкалі ідеального газу, яка будується на залежності тиску ідеального газу від температури. Хоча закони зміни тиску від температури для реальних газів відхиляються від ідеальних, поправки на

ці відхилення є невеликими і можуть бути визначені з високою точністю. Таким чином, шляхом вивчення розширення реальних газів і введення корекцій, можна оцінити температуру за термодинамічною шкалою.

На початку XX століття широко використовувалися шкали Цельсія і Реомюра, а в наукових дослідженнях також використовувалися шкали Кельвіна і воднева. Перерахунки між цими шкалами становили значні труднощі і призводили до розбіжностей. Тому у 1933 році було прийнято рішення про створення Міжнародної температурної шкали (МТШ).

Досвід використання МТШ підкреслив необхідність уточнень і доповнень, щоб максимально наблизити її до термодинамічної шкали. Таким чином, МТШ була переглянута і адаптована до нових даних на той час. У 1960 році було затверджено новий "Статут Міжнародної практичної температурної шкали 1948 року. Редакція 1960 р".

2.2. Пристрої для вимірювання температур

Температуру визначають за допомогою пристроїв, які використовують різні термометричні властивості рідин, газів і твердих тіл. В індустрії, наукових дослідженнях та для спеціальних потреб використовується багато різноманітних пристроїв.

У таблиці 2.1 представлені найбільш поширені пристрої для вимірювання температури та практичні межі їх застосування. Ми розглянемо декілька пристроїв з цієї таблиці [23].

Таблиця 2.1

Пристрої вимірювання температури

Термометрична властивість	Найменування пристрою	Межі тривалого застосування, °C	
		Нижня	Верхня
Теплове розширення	Рідинні скляні термометри	-190	600
Зміна тиску	Манометричні термометри	-160	60

Термометрична властивість	Найменування пристрою	Межі тривалого застосування, °C	
		Нижня	Верхня
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 на основі скляної рідини використовуються для вимірювання температур в діапазоні від -200 до +750 °C. Хоча існують інші пристрої для вимірювання температур, які відповідають вимогам сучасного контролю технологічних процесів, скляні рідинні термометри здобули велику популярність в лабораторній та промисловій практиці через їх простоту використання, високу точність та доступну ціну.

Принцип дії скляних рідинних термометрів базується на тепловому розширенні термометричної рідини, що міститься у термометрі. Показання рідинного термометра залежать від зміни не лише об'єму термометричної рідини, але й об'єму скляного резервуара, в якому розташована ця рідина. Таким чином, відбувається видима зміна об'єму рідини в розмірі, пропорційній збільшенню об'єму резервуара (і частково капіляра) [24].

Для заповнення рідинних термометрів використовують ртуть, толуол, етиловий спирт, газ, петролейний ефір, пентан і т.д. Зокрема, ртутні термометри найбільш поширені, оскільки вони мають кілька переваг, таких як незмочування скла ртуттю, легкість отримання хімічно чистої ртуті та залишання її у рідкому стані в широкому інтервалі температур (від $-38,87$ до $+356,58$ °C) [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 шкальною пластиною з паперу. Термометр повного занурення. При вимірюванні температури повинен повністю перебувати в вимірюваному середовищі. При перегріві без руйнування термометр може витримати температуру вище верхньої межі за шкалою не більше 20 °C (табл. 2.2) [26].

Принцип роботи термометра ґрунтується на тепловій зміні об'єму термометричної рідини, яка відбувається залежно від температури навколишнього середовища. Структура термометра включає капілярну трубку з резервуаром, заповненим термометричною рідиною. Капілярна трубка обернена скляною оболонкою, всередині якої розташована шкала для вимірювання температури. Термометр закріплений на підставці.

Метрологічні і технічні характеристики

Марка	Діапазон вимірювання температури, °C	Ціна ділення шкали, °C	Довжина термометра, мм	Діаметр, мм	Термом. рідина
ТС-7А	-10...+60	1	170	11x7,7 (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 неоднаковий.

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 збільшенні температури на 5 °C відносна вологість впала з 76% до 54%. У випадках високої вологості та низької температури спостерігається зниження транспірації, порушення нормального перебігу фізіологічних процесів у листі, а також активізація різноманітних хвороб, таких як біла гниль, борошниста роса. На обгороджувальних поверхнях, таких як скло, плівка та інші, виникають "краплі" - результат конденсації, що 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, наприклад, в спекотні місяці зимово-весняного обороту (з квітня по липень в сьомій світловій зоні). У нас, вологість

грунту 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%.

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 вдосконалювався, 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.

Назва	Опис	Призначення
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 НТ-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 НТ-86

Особливості Термогігрометра НТ-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: $\pm 3\%$ RH (при 25°C, 20% ~80% RH), $\pm 3.5\%$ 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 с для вимірювання вологості повітря
Живлення	9В батарея 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

Датчик температури і вологості AM2321 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.

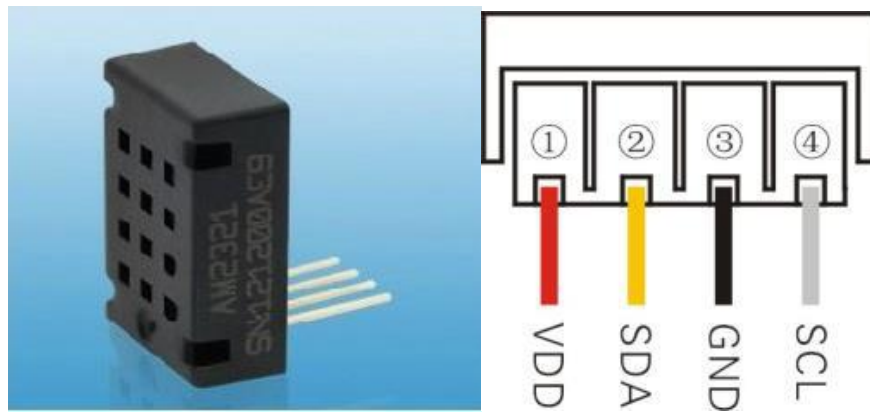


Рис. 2.3. Датчик вологості AM2321

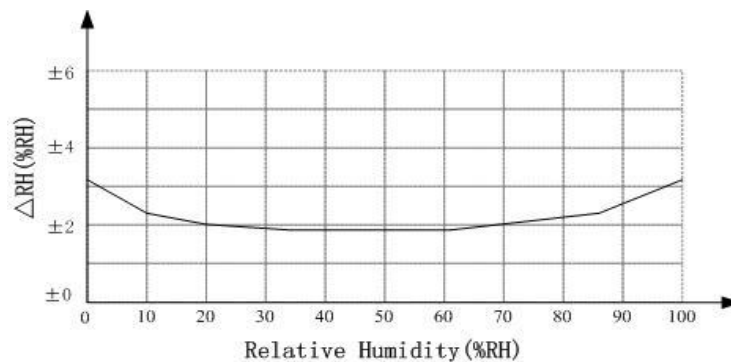


Рис. 2.4. Відносна вологість максимальної похибки AM2321

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 живлення і на шині SCL мінімум на 500мс встановити логічну одиницю. Для перемикавання 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).

AM2321 відносна вологість

Параметр	Стан	Мін	Тип	Макс	Од.вим
Роздільна здатність			0.1		%RH
Асортимент		0		99.9	%RH
Точність	25°C		±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 контрольною сумою температурної компенсації на виході.

Користувачам необов'язково проводити обчислення другорядного цифрового виводу, і немає потреби в температурній компенсації вологості. Два режиму зв'язку можуть вільно перемикатися, користувач може вільно вибирати, простий у використанні або широкий спектр застосування.

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 АМ2321, а на рисунку 2.5 - максимальна похибка датчика температури.

Таблиця 2.7

АМ2321 DC Характеристики

Параметр	стан	мін	тип	макс	блок
Напруга живлення		2.6	5	5.5	v
Споживана потужність	Сплячий режим, міра, в середньому	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 б	VDD = 5V VIN = VSS	30	45	60	kΩ
Вихідний струм	Open Tri-State (Off)	10	8 20		MA μA
Період вибірки		2.0			s

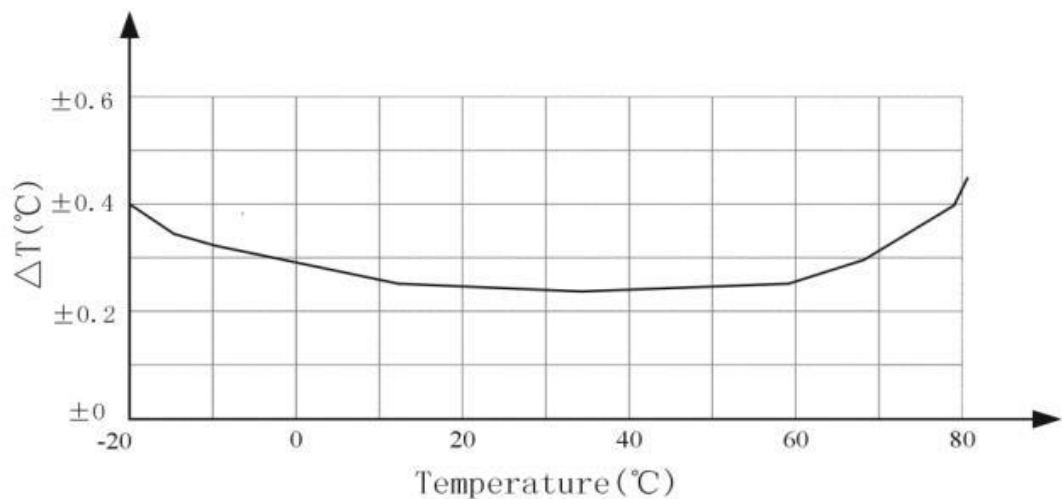


Рис. 2.5. Максимальна похибка датчика температури

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 - ІС-зв'язок.

Після вибору режиму живлення, режим зв'язку з датчиком залишається незмінним; якщо хочете змінити зв'язок, увімкніть живлення, натисніть на операцію вибору, щоб встановити зв'язок, при виборі ІС-зв'язку між мікропроцесором і AM2321, SCL для синхронного 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)

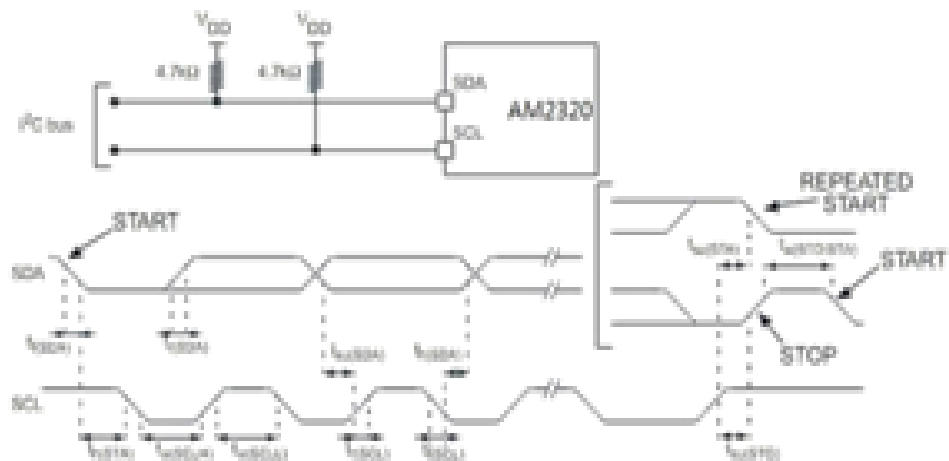


Рис. 2.6. AM2321 шина I2C. Типова схема застосування та тимчасова діаграма

Якщо потрібно отримати результати, використовуючи датчик, необхідно дотримуватися характеристик зазначених в таблиці 2.9 для даного датчика.

Таблиця 2.9

Характеристики інтерфейсу датчика С

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

Symbol	Parameters	Standard C mode		Unit
		min	Max	
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 сенсорні пристрої, дані для запису користувачем і т. д.

Наприклад, код функції 03 повідомляє хосту через значення регістра, що повернені датчиком дані (включаючи початкову адресу регістру, що підлягає читанню, 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 по сенсорному пристрою 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	Addresses	Register information	Addresses	Register information	Addresses	Register information	Addresses
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

Register information	Addresses	Register information	Addresses	Register information	Addresses	Register information	Addresses
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

Температурний дозвіл є 16-бітовим (табл. 2.11), старший біт (Bit15) дорівнює 1 і вказує на негативну температуру, якщо старший біт (Bit15) дорівнює 0, вказує на позитивну температуру; температура на додаток до найбільш значимого біту (Bit14 ~ Bit0) 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 приладами з високою точністю вимірів. 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 3

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 незалежної пам'яті: при його безпосередній участі були розроблені перші типи електрично стираємих 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-чипів 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 якій 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 виявилось значно зручніше одноразових кристалів або УФ-стираних, що вимагають тривалої "ручної" 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 такти - конвеєр вважає, що умова розгалуження не виконується).

Первісне ядро складалося всього з 32 тисяч транзисторів.

У роботі над ядром взяла участь згадувана фірма IAR System, розробник C-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 C при такій архітектурі значно простіше: фактично залишалось лише упакувати те, що знаменитий програміст Дейкстра назвав "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 C. 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 "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 подібні переваги архітектура AVR швидко стала популярною. Пізніше таким інтерфейсом були змушені обзавестися й інші великі сімейства МК.

До недоліків AVR можна частково віднести те, що електрично виходи портів за рівнями сигналів є КМОП-сумісними: це підвищує стійкість (поріг розрізнення низького-високого рівня лежить в районі середини живлення, а не ближче до нуля, як у x51), але викликає ряд проблем сумісності з різними стандартними інтерфейсами. Не дуже логічна і побудова деяких команд в області операцій з бітами. Але в цілому архітектура AVR, безсумнівно, більш прогресивна і якби в цій галузі зміни відбувалися 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 за технологією КМОП і використовує RICS архітектуру, що в поєднанні дозволяє досягти оптимального балансу між швидкістю та енергоспоживанням.

Основні переваги Atmega16 включають:

- здатність обробляти 131 потужну інструкцію, більшість з яких виконуються за один машинний цикл;
- наявність 32 8-розрядних регістрів загального призначення і регістрів управління вбудованою периферією;
- повністю статичну роботу; та продуктивність до 16 мільйонів операцій в секунду при тактовій частоті 16 МГц.
- Вбудований пристрій множення виконує множення за 2 машинних цикли;
- 16 Кбайт внутрисистемної 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 / виведення (рис. 3.2).

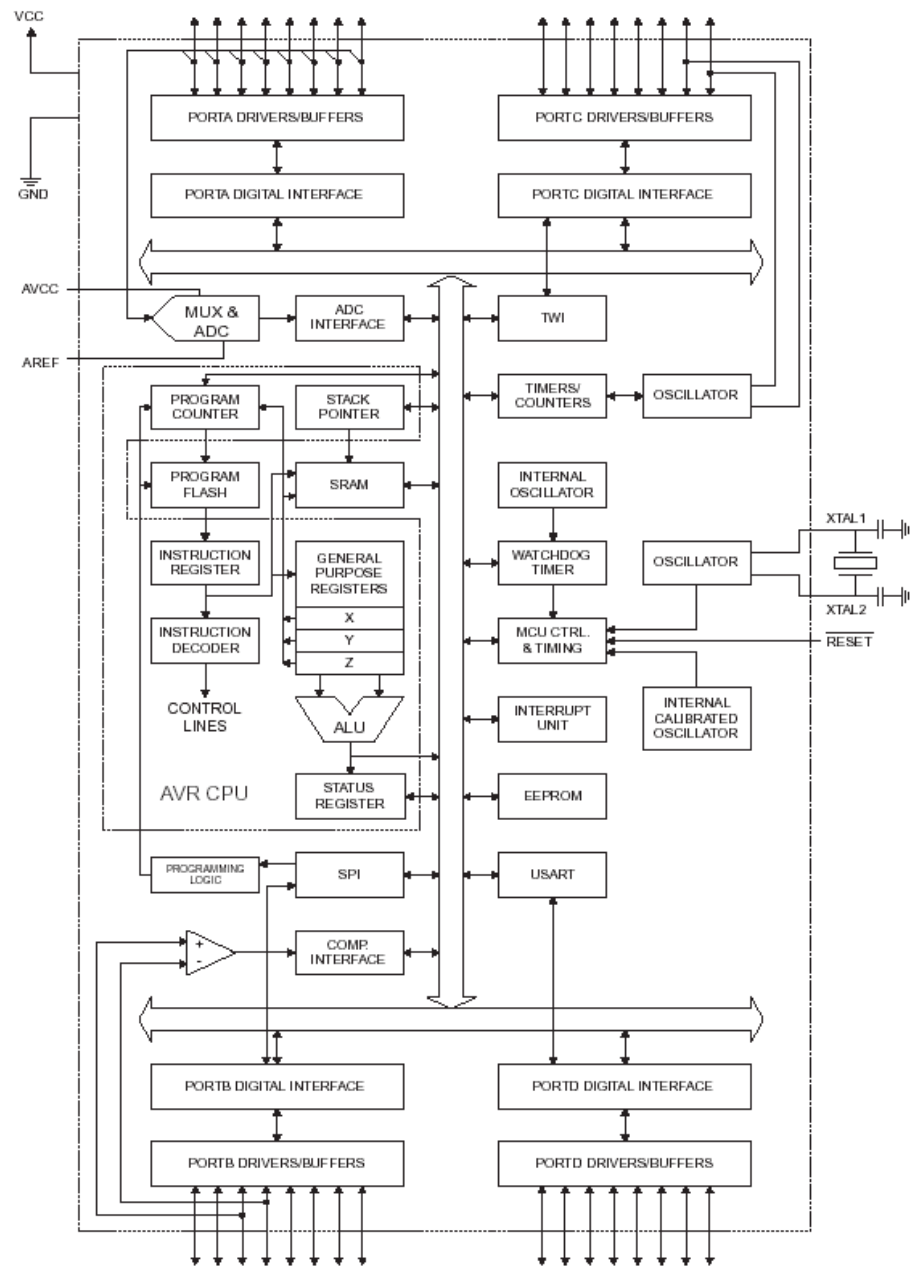


Рис. 3.1. Блок-схема ATmega16

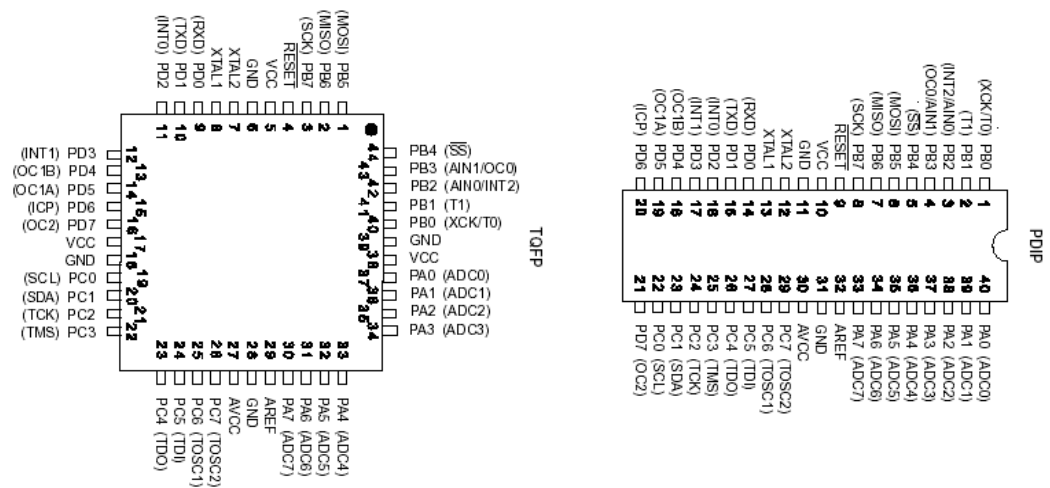


Рис. 3.2. Розташування виходів ATmega16

3.3. Принцип роботи електронного пристрою

Відлік часу проводиться всередині мікроконтролера за допомогою асинхронного таймера - T2. 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 74НС595.

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 теплиці. Рядок, що біжить, налаштовується, тобто можна ввімкнути / вимкнути відображення будь-якого з елементів. При виключенні відображення всіх елементів рядка, що біжить, вона не запускається зовсім, і годинник постійно відображає лише час.

Живлення будильника можливе 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, які можуть 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 багато точкових 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).

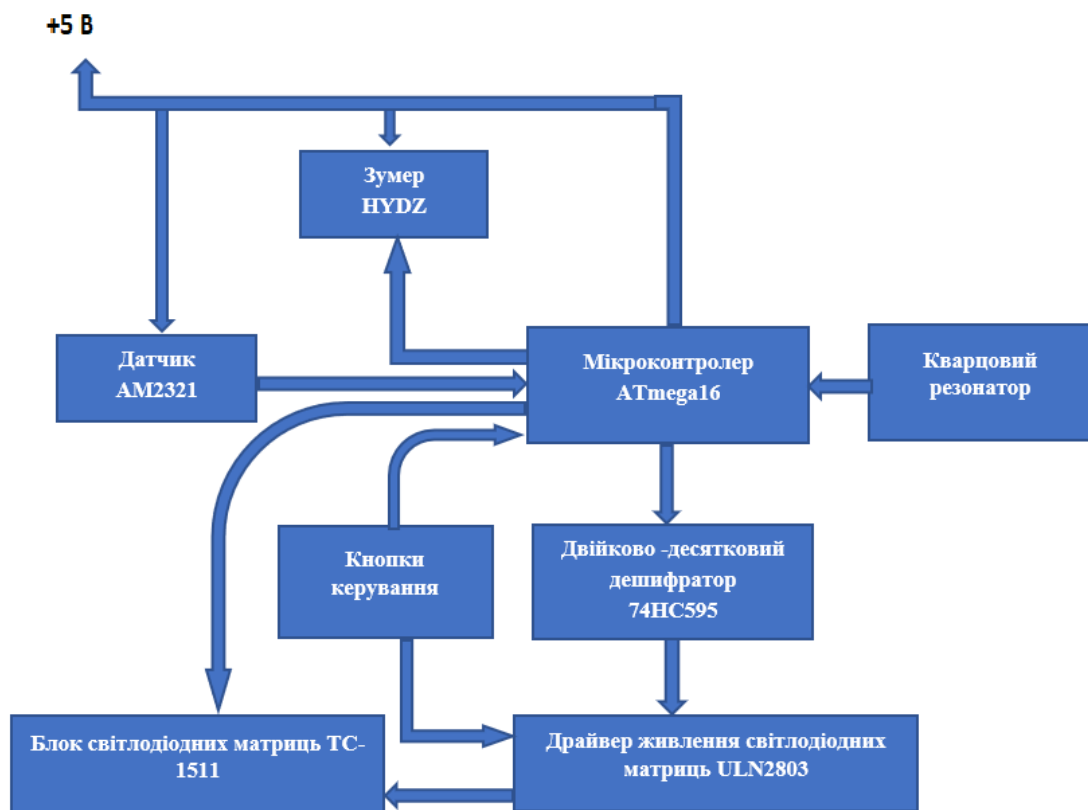


Рис. 3.3. Синтез структурно-функціональної схеми пристрою

Призначення кнопок будильника:

1. Кнопка 1 - умовна назва "ОК".
2. Кнопка 2 - умовна назва "КРОК".

Управління будильником здійснюється двома кнопками. Їх кількість можна було довести до 8 шт, не додаючи жодного компонента, крім самих кнопок, але для зручності "скористалися" всього двома. Кнопки умовно названі "ОК" і "КРОК". Кнопкою "КРОК" як правило відбувається перехід до наступного пункту меню, а кнопкою "ОК" зміна параметрів поточного меню. Сигнал спрацювання будильника також вимикається кнопками "ОК" або "КРОК". Натискання будь-якої кнопки під час сигналу будильника відключає його.

"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 живлення від батарейки, акумулятора, або від іоністори. Струм споживання будильника в черговому режимі становить 15 мкА. При живленні від іоністори на 1Ф, будильник пропрацював чотири доби. Цього цілком достатньо для підтримки ходу під час перебоїв живлення. Якщо застосувати батарейку CR2032, заряду повинно вистачити на 1,5 року. Наявність напруги контролер приймає через вихід PB.3. Напруга живлення через діодник R2-R3 подається на вихід PB.3, і в нормальному стані дорівнює приблизно 1,5 В. Якщо зовнішня напруга впаде нижче 4,1 вольт, то напруга на виводі PB.3 стане менше 1,23 вольт, при цьому згенерує переривання від компаратора, і в обробнику цього переривання вимикаються всі "зайві" вузли контролера, і сам контролер йде в режим очікування. В цьому режимі продовжує працювати і тільки відлічує час таймер T2. При появі зовнішнього

живлення, напруга на РВ.3 знову підніметься вище 1,23 В, контролер "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 перевіряв 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-середа 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 використовувався додаток DipTrace версії 2.3.1.0, що представляє собою систему автоматизованого проектування друкованих плат (PCB). Вона включає чотири програми:

- Schematic - для розробки принципових схем;
- DipTrace - для створення плат, ручного та автоматичного трасування;

- ComEdit - редактор корпусів;
- SchemEdit - редактор компонентів.

Ця програма відзначається найзручнішим інтерфейсом серед САПР для проектування друкованих плат, що призначений для користувача. Робота супроводжується підсвічуванням редагованих та залежних елементів, що надає можливість наочно оцінювати ситуацію. Логічна структура принципової схеми або плати формується відразу при їх побудові, і зміна одного елемента відображається на залежних від нього [37].

У редакторах корпусів і компонентів використовується пошарова система малювання, що дозволяє створювати компоненти та корпуси на основі готових шаблонів. Також можливе створення корпусів зі змінними параметрами (кількість ніжок, відстань між ними), які можна буде змінювати прямо в РСВ.

Схемотехніка підтримує створення багатолистових принципових схем, і перехід з одного аркуша на інший здійснюється за допомогою загальних шин, що з'єднані міжсторінковими переходами. Принципова схема, розроблена в даному додатку, може бути перетворена в друковану плату вибором відповідного пункту меню або за допомогою комбінації клавіш в схемотехніці. Перетворена плата представляє собою корпус компонентів із логічними зв'язками [38, 39].

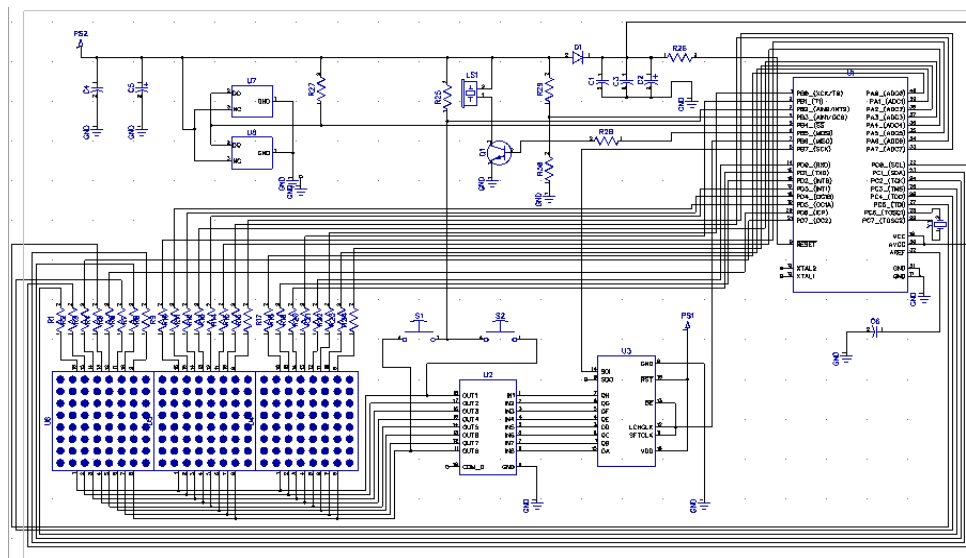


Рис. 3.4. Schematic. Принципова схема пристрою

При використанні ручного трасування провідників, всі виводи мережі підсвічуються, і вже розведені зв'язки автоматично видаляються. Під час редагування розведених трас можливе переміщення фрагментів провідників з дотриманням кутів 90 і 45 градусів або вільного розташування, зміна ширини, переклад з одного шару на інший і так далі. Автоматичне трасування виконується за допомогою вбудованого оптимізаційного трасувальника SimpleRouter, який працює на основі сіткової структури.

Закінчений варіант трасування плати представлений на рисунках 3.5 і 3.6.

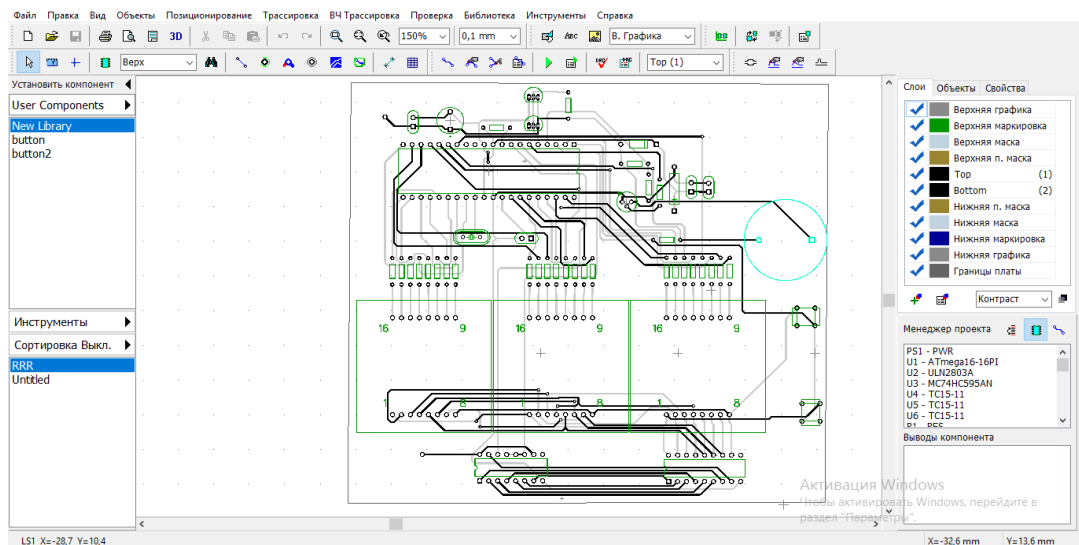


Рис. 3.5. PCBLAYOUT. Верхня сторона плати розроблювального пристрою

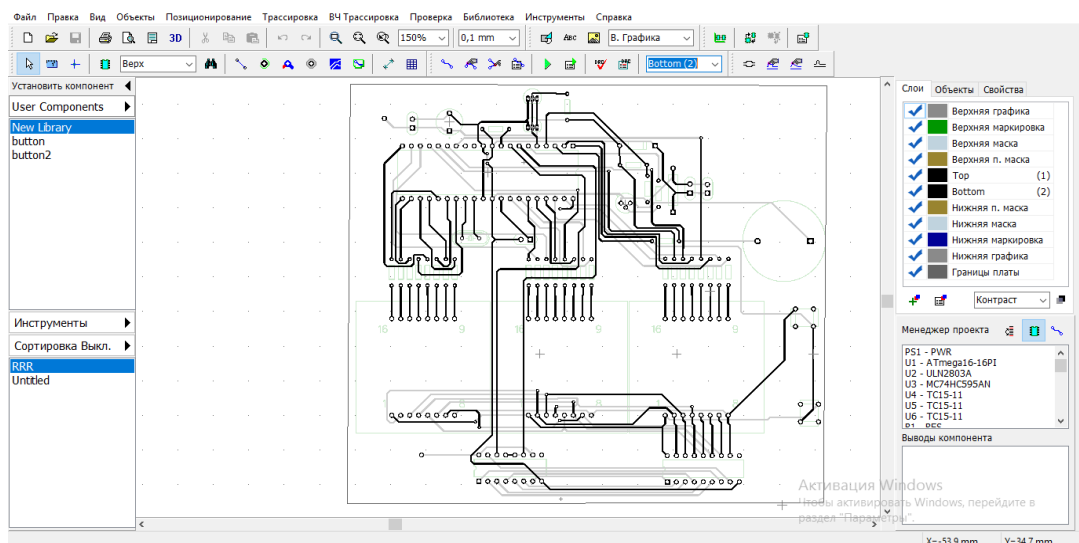


Рис. 3.6. PCBLAYOUT. Нижня сторона плати розроблювального

пристрою

Створення висновків для принципів схем і друкованих плат в програмі може бути виконане в будь-якому масштабі. Об'єкти для друкованих плат визначаються в довільному масштабі, а для принципів схем - у вигляді листів. Також є можливість вставляти рамки і штампи, а для рамок можна задавати розмітку зон. Створення штамтів відбувається безпосередньо у програмі за допомогою спеціального редактора, що працює на засадах електронних таблиць.

Особлива функція експорту для обведення в DXF дозволяє виготовляти плати за допомогою фрезерування (перетворення DXF у G-код обов'язкове).

DipTrace містить обширний набір бібліотек для корпусів і компонентів, включаючи стандартні бібліотеки з більш ніж 40 тис. компонентів від різних виробників.

Створений пристрій виглядає так, як показано на рисунку 3.7

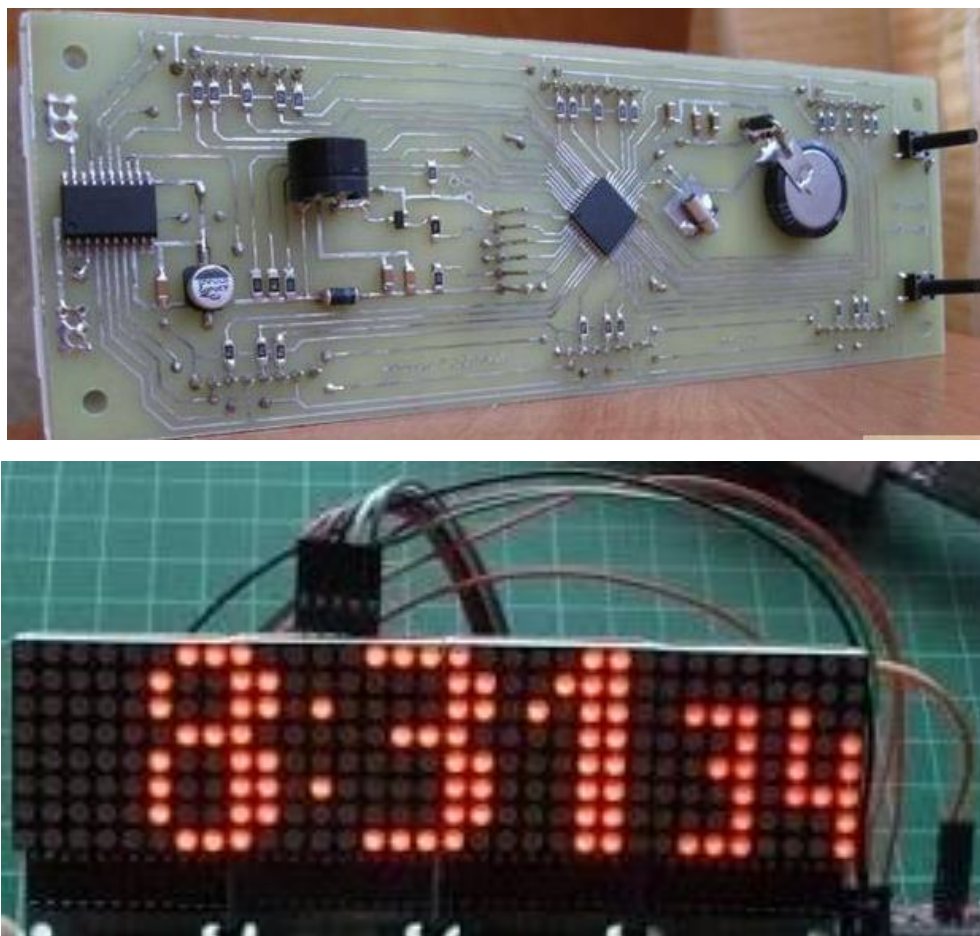


Рис. 3.7. Плата розроблювального пристрою

3.5. Розробка програмного забезпечення

Для написання програми роботи мікроконтролера була вибрана мова програмування C 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 розвиненого 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, контекстним підсвічуванням, автокорекцією, кодовими шаблонами і т. п.);
- використання 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 програматорами;
- 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 він 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 вивід даних на світлодіодні матриці.

Блок-схема алгоритму роботи пристрою приведена на рисунку 3.8.

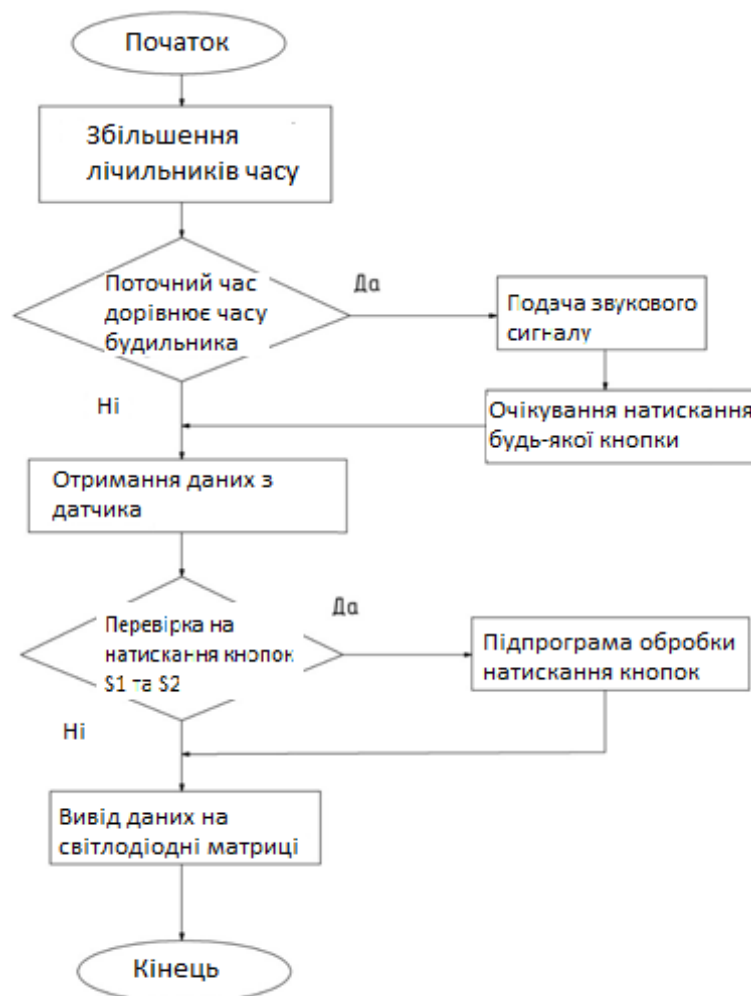


Рис. 3.8. Алгоритм роботи пристрою

3.7. Висновки до розділу

У додатку Diptrace була побудована схема друкованої плати, на базі якої було сконструйовано пристрій. Також в розділі описане середовище розробки, в якому розроблявся програмний додаток для пристрою. Представлений алгоритм роботи пристрою у вигляді блок-схеми.

ВИСНОВКИ

У магістерській роботі було розглянуто розробку інформаційно-виміральної системи контролю температури і вологості повітря на основі математичного дослідження мікроклімату тепличних приміщень.

Перший розділ містить опис існуючого вигляду пристрою, використовуваного для контролю вологості і температури в теплиці на основі плати Arduino Mega.

Так само в першому розділі описані необхідні умови для вирощування рослин і представлена математична модель теплиці при управлінні температурою і вологістю повітря, яка розроблена для застосування в освітніх цілях з метою продемонструвати фізичні явища, що відбуваються в теплиці і принципи управління мікрокліматом.

У другому розділі проаналізувавши способи вимірювання температури було визначено, що рідинні скляні термометри є надійними приладами з високою точністю вимірів. 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 приладом 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 описано середовище розробки, в якому розроблявся програмний додаток для пристрою. Був представлений алгоритм роботи пристрою у вигляді блок-схеми.

Отже, розробка, дослідження та впровадження передових адаптивних методів та комп'ютеризованих засобів інтелектуалізації та цифровізації

процесів моніторингу, контролю та управління станом мікроклімату в теплицях дозволять здійснити перехід від феноменологічного підходу до вирощування культур на захищених ґрунтах до використання сучасних аграрних рішень на базі апаратно-програмних комплексів. Це, в свою чергу, сприятиме підвищенню технологічних та економічних показників у процесі вирощування культур у теплицях.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. 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).
2. 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 «Інтелектуальні комп'ютерні системи та мережі». – Тернопіль: ЗУНУ. – 2020. – С. 37.
3. 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 «Інтелектуальні комп'ютерні системи та мережі». – Тернопіль: ЗУНУ. – 2020. – С. 38.
4. 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.
5. Кэмп П. Компьютерное управление микроклиматом в теплицах / Кэмп П, Тиммерман Г. // Центр инноваций и практического обучения в Эдде - 1997. – 193с.
6. 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.
7. International Standart ISO 9126-1. Software engineering – Product quality – Part 1: Quality. – 2001. P. 32.
8. Kharchenko V. Green Computing and Communications in Critical Applocation 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.

9. 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.
10. Маркелова И.В. Лимоны. Уход и выращивание ООО "Авеонт", 2006. – 96 с.
11. Илья Мельников. Теплица, парник, зимний сад. ЛитРес, 2009. – 110 с.
12. Сааков С.Г. Оранжерейные и комнатные растения и уход за ними, 1985. – 623 с.
13. С. Раделов. Все о теплицах и парниках на садовом участке. Вильнюс, Bestiary. 2013
14. Людмила Шульгина. Теплицы и парники. Строительство и рекомендации по выращиванию овощей, цветов, грибов. Книжный клуб "Клуб семейного досуга". Харьков, Книжный клуб "Клуб семейного досуга". Белгород, 2012.
15. Алан Титчмарш. Умная теплица. Петроглиф, 2013.
16. Н. Новикова. Архитектура теплиц и оранжерей. Архитектура-С, 2006.
17. Галина Кизима. Урожайная теплица. Что когда сажать. Как удобрять и защищать. АСТ, 2013.
18. Анализ системы управления микроклиматом оранжереи компании "Альфа" на плате Arduino Mega. [Электронный ресурс] – URL: <http://meandr.org/archives/25973>.
19. Исследования математической модели микроклимата оранжереи. [Электронный ресурс] – URL: <http://cyberleninka.ru/article/n/matematiceskaya-model-mikroklimate-teplitsy>.
20. Токманов Н.М. Математическая модель системы управления микроклиматом ангарных теплиц Н.М., В.С. Грудинин. Петроглиф, 2008.

21. Буклагина Г.В. Автоматизация теплиц различных поколений/ Инженерно-техническое обеспечение АПК. Реферативный журнал. 2005. – № 4. – 922 с.
22. Крупнов Б.А., Шарафудинов Н.С. Руководство по проектированию систем отопления, вентиляции и кондиционирования воздуха. Вена. 2008 – 220 с..
23. Рысс А.А. Автоматическое управление температурным режимом в теплицах. - М.: Агропромиздат, 1986 - 126 с.
24. Токманов Н.М. Особенности управления микроклиматом в ангарных теплицах / Гавриш №3, 2008 г., с. 28-32.
25. Гудвин Г.К. Проектирование систем управления. / Г.К. Гудвин, С.Ф. Гребен, М.Э. Сальгадо.- Москва.: БИНОМ.2004.
26. Исследование динамических характеристик системы управления микроклиматом оранжерей. [Электронный ресурс] – URL: <http://dspace.mnau.edu.ua:8080/jspui/handle/123456789/313>.
27. Дашенко А.Ф. и др. MATLAB в инженерных и научных расчетах. 28. Одесса, 2003. - 214 с.
29. Дьяконов В. П. MatLab 6.5 SP1 (7.0). Simulink 5/6. Основы применения. СОЛОН-пресс, 2005. – 800 с.
30. Егоренков Д.Л., Фрадков А.Л., Харламов В.Ю. Основы математического моделирования. Построение и анализ моделей с примерами на языке MATLAB. Учебное пособие. – СПб.: БГТУ, 1994. – 190 с.
31. Понятие о температуре и температурных шкалах [Электронный ресурс] – URL: <http://www.physicedu.ru/phy-540.html>.
32. Устройства для измерения температур [Электронный ресурс] – URL: <http://100pudov.com.ua/subject/80/34560/>.
33. Жидкостные стеклянные термометры [Электронный ресурс] – URL: http://studbooks.net/1936578/matematika_himiya_fizika/termometry_steklyannye_zhidkostnye.

34. Г.Г. Раннев, А.П. Тарасенко "Методы и средства измерений" 2004г
35. Термометр ТС -7А [Электронный ресурс] – URL:
<http://www.steklopribor.com/termometri-steklyannie-ts-7-m1>.
36. Вологість [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%8C>.
37. Вологість в ґрунті [Электронный ресурс] – URL:
<http://www.activestudy.info/rezhim-vlazhnosti-v-teplicax/>.
38. Устройства для измерения влажности [Электронный ресурс] – URL:
http://www.vseinstrumenti.ru/instrument/izmeritelnyj/gigrometry/articles/vi dy_ustrojstvo/.
39. Афонский А.А., Дьяконов В.П. Измерительные приборы и массовые электронные измерения. Солон-Пресс, 2007.
40. Термогигрометр HT-86 [Электронный ресурс] – URL:
<http://simvolt.ua/termogigrometr-ht-86.html>.
41. Датчик температуры и влажности AM2321 [Электронный ресурс] –
<http://radiokot.ru/forum/viewtopic.php?f=25&t=117421>.
42. Температура и влажность комбинированного датчика AM2321 AM2321 [Электронный ресурс] –
http://akizukidenshi.com/download/ds/aosong/AM2321_e.pdf.
43. Семейство AVR [Электронный ресурс] –
<http://www.alldatasheet.com/view.jsp?Searchword=Atmega16&gclid=CMX MgIK36NICFcVUGAodHPsMjA>.
44. Гребнев В.В. Микроконтроллеры семейства AVR фирмы Atmel. - М.: Радиософт, 2002.
45. Datasheet: AT90S8515, ATmega16, ATmega8, PCA9544/PCA-9544A, AVR315.
46. 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.

47. В помощь радиолюбителю. Солон-Пресс 2012. – 200 с.
48. Белов А.В. - Конструирование устройств на микроконтроллерах. НиТ, 2005– 256 с.
49. Розробка програмного забезпечення [Електронний ресурс] – URL: http://cxem.net/software/soft_mcu.php.
50. Б. Керниган, Д. Ритчи: Язык программирования Си Издание 3-е, исправленное. Перевод с английского под редакцией Вс. С. Штаркмана, 1992 –253 с.
51. Герберт Шилдт. Полный справочник по С (4-ое издание) 2004 – 704 с.
52. Прокопенко В.С. – Программирование микроконтроллеров ATMEЛ на языке С. МК-Пресс, 2012 – 320 с.
53. Шпак Ю.А. Программирование на языке Си для AVR и PIC микроконтроллеров 2006. – 300 с.
54. "Краткий справочник по языку "С"" В.А.Ильина, П.К.Силаев, 2012, Московский государственный университет, физический факультет, кафедра квантовой теории и физики высоких энергий.
55. 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 "ОК" (на выводах 5, 2, 7, 1, 12,
8, 14, 9 катоды диодов)
#define OA 0 // для
матрицы с "ОА" (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], //
массив с адресами найденных 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 "Будильник"
korekt_txt []=
{ 20,24,26,26,15,20,32,18,41,42,42,255}, //
текст "Коррекция"
nastroiki_txt []=
{ 29,27,28,10,23,24,12,20,18,42,255}, //
текст "Установки"
den_txt []= { 14,15,23,38,255},
// текст "День"
data_txt []= { 14,10,28,10,255},
// текст "Дата"
god_txt []= { 13,24,14,255},
// текст "Год"
nastr_stroki_txt []=
{ 23,10,27,28,26,24,19,20,10,42,255}, //
текст "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
// А { 0x7C, 0x12, 0x12, 0x7C, 0xAA },
// Б { 0x7E, 0x4A, 0x4A, 0x32, 0xAA },
// В { 0x7E, 0x4A, 0x4A, 0x34, 0xAA },
Г { 0x7E, 0x02, 0x02, 0xAA, 0x00 }, //
Д { 0x60, 0x3C, 0x22, 0x3E, 0x60 }, //
// Е { 0x7E, 0x4A, 0x4A, 0xAA, 0x00 },
// З { 0x4A, 0x4A, 0x7E, 0xAA, 0x00 },
Ж { 0x66, 0x18, 0x7E, 0x18, 0x66 }, //
И { 0x7E, 0x10, 0x08, 0x7E, 0xAA }, //
// И { 0x7C, 0x11, 0x09, 0x7C, 0xAA },
К { 0x7E, 0x18, 0x24, 0x42, 0xAA }, //
Л { 0x78, 0x04, 0x02, 0x7E, 0xAA }, //
М { 0x7E, 0x04, 0x08, 0x04, 0x7E }, //
Н { 0x7E, 0x08, 0x08, 0x7E, 0xAA }, //
// О { 0x3C, 0x42, 0x42, 0x3C, 0xAA },

П { 0x7E, 0x02, 0x02, 0x7E, 0xAA }, //
// Р { 0x7E, 0x12, 0x12, 0x0C, 0xAA },
С { 0x3C, 0x42, 0x42, 0x24, 0xAA }, //
Т { 0x02, 0x7E, 0x02, 0xAA, 0x00 }, //
У { 0x4E, 0x50, 0x50, 0x3E, 0xAA }, //
Ф { 0x0C, 0x12, 0x7E, 0x12, 0x0C }, //
Х { 0x66, 0x18, 0x18, 0x66, 0xAA }, //
Ц { 0x7E, 0x40, 0x40, 0x7E, 0xC0 }, //
Ч { 0x0E, 0x10, 0x10, 0x7E, 0xAA }, //
Ш { 0x7E, 0x40, 0x7E, 0x40, 0x7E }, //
Щ { 0x7E, 0x40, 0x7E, 0x40, 0xFE }, //
Ъ { 0x02, 0x7E, 0x48, 0x30, 0xAA }, //
Ы { 0x7E, 0x48, 0x30, 0x00, 0x7E }, //
Ь { 0x7E, 0x48, 0x30, 0xAA, 0x00 }, //
// Э { 0x24, 0x42, 0x4A, 0x3C, 0xAA },
Ю { 0x7E, 0x08, 0x3C, 0x42, 0x3C }, //
// Я { 0x4C, 0x32, 0x12, 0x7E, 0xAA },
{ 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 }, //
"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 С 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 korrekt (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); }
}

//_Динамическая 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;}
// выключаю "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 мег16)
# endif
# if (ATMEGA32)
MCUCR=0b10110000; //
разрешаем усыплять контроллер по
команде SLEEP (для мег32)
# 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
"пищим" по нажатию кнопки, или
"ежечасно" то сигнал однотоновый
    else OCR1A = dl_1+3; // если
"пищит" будильник, то сигнал
"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)      { korrekt();
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_mesec[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 отведенное 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; // "выключаем" все
будильники
den_nedeli=Day_week (); //
Вычисляем день недели
    ochistka(); // Очищаем весь
экран

```

```

    devices=w1_search(0xf0,rom_code);    //
Ищем датчики, и записываем их адреса в
массив
    ds1820_d=0; ds1820_y=1;
    #asm("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);
//ищем датчики, и записываем их адреса в
массив
                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,
записываем их в "temperature" и "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 "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 "дату" нужно выводить
                {
(chislo>9) {beg_info[t++]=chislo/10;} //
Если число больше 9, выводим "десятки"
числа
                beg_info[t++]=chislo%10;
// Выводим "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 "Год" нужно выводить
                {
                beg_info[t++]=2;
// "Тысячи" года (2)
                beg_info[t++]=0;
// "Сотни" года (0)

```



```

beg_info[t++]=(god%100)/10;
// "Десятки" года
        beg_info[t++]=god%10;
// "Единицы" года
        beg_info[t++]=13;
// "Г"
        beg_info[t++]=42;
beg_info[t++]=42;          // 2 пробела
    }
    if (temp2 & 0x08)      // Если
"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
"десятки" температуры дома

beg_info[t++]=(abs(temperature[ds1820_d]
)%100)/10;          // Выводим
"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 "ДОМ"
        beg_info[t++]=42;
// пробел
    }
    if ( (temp2 & 0x10) &&
(devices>1) )      // Если
"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
"десятки" температуры дома

beg_info[t++]=(abs(temperature[ds1820_y]
)%100)/10;          // Выводим
"единицы температуры"
        beg_info[t++]=48;
// Знак градуса
        beg_info[t++]=42;
// мал пробел
        beg_info[t++]=29;
beg_info[t++]=21;          //
слово "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 "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 -
"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 "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 "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 "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 "beg_info"
содержит название дня недели)
            ekran_1_figure
(((budilnik_Install[bud] & (1 << temp)) ?
47:46),17); // вывожу знак
"+" или "-" обозначающий вкл./выкл.
будильника.

```

```

    if (BUT_STEP) {temp++; ochistka();
// "перебираю" дни недели для
будильника
        if (temp==7){ m1:
meny=10; budilnik_Interval[bud] =
ustanovki_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 -
"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 -
"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 "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),1
2); 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),1
2);
            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{k
orr_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),1
2);
            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 "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 "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 "А" 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("sleep")       // спим.....
  ACSR=0x4b;          // включаю
компаратор, внутренний ИОН.
}
//*****
```

