

**ЕЛЕКТРОННА СИСТЕМА МОНІТОРИНГУ ТА КЕРУВАННЯ
СИСТЕМОЮ АВТОМАТИЧНОГО ПОЛИВУ РОСЛИН У ТЕПЛИЦЯХ
НА БАЗІ НЕЧІТКОЇ ЛОГІКИ**

ШИФР – СИСТЕМА АВТОЗРОЩУВАННЯ

ЗМІСТ

АНОТАЦІЯ.....	3
1. АНАЛІЗ ІСНУЮЧИХ РОЗРОБОК.....	5
1.1. Система крапельного поливу від ТМ «Рідна Роса».....	5
1.2. Розробка автоматичного крапельного поливу тепличних культур від науковців Оренбурзького державного аграрного університету	6
1.3. Актуальні розробки на базі нечіткої логіки	8
2. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ.....	9
2.1. Методи вимірювання вологи та температури в ґрунті	9
2.1.1. Методи вимірювання вологості	9
2.1.2. Методи вимірювання температури	12
2.2. Створення структурної схеми системи автоматичного поливу	14
2.3. Алгоритм роботи автоматичного поливу на базі нечіткої логіки	15
2.3.1. Блок вхідних даних.....	15
2.3.2. Блок вихідних даних.....	17
2.3.3. Блок створення бази правил.....	19
2.4. Створення теоретичної моделі системи автоматичного поливу	20
2.5. Створення реальної моделі на базі мікроконтролера Arduino.....	23
ВИСНОВКИ ТА ПЕРСПЕКТИВНІ НАПРЯМКИ ДОСЛІДЖЕНЬ.....	26
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	27
Додаток А – Лістинг програмного коду автоматичної системи на базі Fuzzy logic	29

АНОТАЦІЯ

В сільському господарстві дуже активно використовується тепличний спосіб вирощування рослин, як для вирощування розсади так і для отримання зрілих овочів. Тепличне вирощування подовжує вегетативний період та забезпечує фермерів свіжими овочами не тільки в теплу пору року, а також і протягом зимнього та весняного періоду.

Теплиця представляється у вигляді конструкції зі скляним або пластиковим дахом та стінами з такого ж матеріалу. Вона нагрівається, тому що сонячне випромінення або система обігріву в самій теплиці зігріває рослини, ґрунт та інші об'єкти, що знаходяться всередині. Іншими словами, теплиця – це структура, як правило, зі скла або прозорого пластику, яка забезпечує захист та контрольоване середовище для вирощування рослин в приміщенні.

Актуальність. Вода – один із важливіших елементів, який потрібен для життєдіяльності, росту та родючості рослин. Об'єм води та частота поливу змінюється в залежності від температури, довжини світлового дня, фази розвитку рослини та виду ґрунту. Як відомо більшість садівників використовує ручний спосіб поливу. Ця система неефективна через те, що під час ручного способу поливу вірогідність надмірного зволоження, а саме великого відсотку вологи в ґрунті (вологості ґрунту) дуже велика. Надмірна вологість ґрунту може призвести до затоплення рослин або до загнивання частин рослини. Недостатня вологість призводить до заповільненого розвитку рослини, її плодів або навіть до повного засихання та загибелі рослини.

Для того, щоб вирішити цю проблему необхідно використовувати автоматичну систему поливу. Необхідна така система, котра б була здатна зчитувати основні параметри, що впливають на відсоток вологості ґрунту в реальному часі та не допускала б перерегулювання. Дана система повинна опитувати датчики та в залежності від вхідних параметрів обирати найліпший варіант для поливу ґрунту. Такими датчиками є:

– датчик температури ґрунту;

– датчик вологості ґрунту.

Окрім цього бажано інтегрувати дану систему автоматичного поливу в загально доступному для садівників програмному забезпеченні та апаратному виконанні.

Мета. Основна мета наукової роботи – це створення та дослідження електронної системи автоматичного поливу за допомогою нечіткої логіки (Fuzzy logic). Дослідження та моніторинг градієнту розповсюдження вологи в залежності від початкового відсотка вологості.

Поставлені задачі:

- дослідження та аналіз вже існуючих систем автоматичного поливу;
- аналіз методів вимірювання вологості ґрунту;
- розробка алгоритму автоматичного поливу за допомогою нечіткої логіки;
- створення програмного коду на базі мікроконтролера Arduino;
- реалізація досліджуваного макету електронної системи моніторингу й керування вологості ґрунту.

Загальна методика досліджень: основні науково-практичні результати роботи отримані на базі методів аналізу й розрахунку електронних компонент вимірювальної систем; способів інструментального комп'ютеризованого вимірювального моніторингу багатофазних дисперсних середовищ; методик математичного планування експерименту; методів комп'ютерного моделювання інформаційно-вимірювальних систем систем.

Структура та обсяг роботи. Робота складається зі вступу, двох розділів, обсягом 28 сторінок та 1 додатку на 8 сторінках, загальних висновків та переліку посилань. Робота містить 12 рисунків та 9 таблиць.

Ключові слова: електронна система, вологість, моніторинг, теплиця, датчик, Arduino.

1. АНАЛІЗ ІСНУЮЧИХ РОЗРОБОК

В світі дуже гостро стоїть питання автоматизації систем поливу, тому дуже багато науковців та дослідників намагаються вирішити це питання. Для того, щоб створити таку систему необхідно, по-перше, обрати метод, яким буде вимірюватись температура й вологість ґрунту та система поливу, що буде використана в системі.

У сучасному світі існує дві найпопулярніші види системи поливу: дощування та крапельна система. Ці системи мають різні конструкції та способи реалізації. Система крапельного поливу має такі переваги, як:

- відносно низьку ціну на закупівлю необхідних витратних матеріалів (клапани, шланги, крапельниці, резервуар для води та насос);
- малі витрати на електроенергію;
- невеликий робочий тиск, що попереджує надмірний полив ґрунту, вимивання ґрунту з під кореня та не пошкоджує листя через попадання води на нього;
- надходження води здійснюється біля кореня рослини, що сприяє точному потраплянню вологи в необхідному рослині об'ємі.

На теперішній час існує декілька рішень для систем автоматизованого та автоматичного крапельного поливу, що відрізняються системою керування поливом, наявністю або відсутністю систем моніторингу стану рослин, можливістю дистанційного контролю.

Розглянемо декілька варіантів таких систем:

1.1. Система крапельного поливу від ТМ «Рідна Роса».

Автоматична система крапельного зрошення продумана таким чином, щоб повністю звести трудовитрати на полив до нуля, при цьому економічно використовує воду. Щоб у теплиці працювала система автополиву не потрібна електрична мережа (працює від 8 батарейок АА) і нормальний водопровід

(бажано підключити до водопроводу, але можна ємність наповнювати відрами). Автоматична система поливу оснащена контролером або пультом управління, завдяки якому можна вибрати зручний для поливу режим тієї чи іншої ділянки. Також до системи додається програмований таймер, на якому можна виставити час поливу, бажаний обсяг води для поливу.

До переваг цієї системи можна віднести:

- разом з водою можна вносити добрива або мікроелементи для «підкормки» рослин;
- економія води на 40-50%;
- бур'яни ростуть повільніше, рослини не так хворіють фітофторою або грибковими хворобами;
- ґрунт залишається рихлим, не забитим;
- збільшується врожайність, товарний вигляд, строк зберігання плодів.

1.2. Розробка автоматичного крапельного поливу тепличних культур від науковців Оренбурзького державного аграрного університету [1]

Автори даної наукової статті розглядають питання, що пов'язані з процесом автоматизації керування крапельним поливом в теплицях. Вони виявили основні недоліки промислових автоматизованих систем поливу, такими недоліками є:

- відсутність постійного моніторингу вологості ґрунту;
- тривалість поливу регулюється таймером часу, в результаті якого з'являється необхідність періодично вносити корегування;
- мають один канал вимірювання, що не дозволяє одним пристроєм обслуговувати різні культури.

Звертаючи увагу на ці недоліки автори розробили пристрій, який здатний виконувати автоматичний полив овочевих культур на п'яти різних ділянках теплиці, проводить постійний моніторинг вологості, повідомляє про несправності в системі поливу. Непереривне вимірювання вологості ґрунту

одночасно виводиться на рідкокристалічний дисплей, це дає можливість керувати поливом на п'яти різних ділянках теплиці. В пристрої в якості датчика вологості використовується резистивний датчик. Пристрій визначає опір ґрунту наступним чином. Змінна напруга подається з понижуючого трансформатора 1 (див. рис. 1.1) на датчик 4 через шунт 5. В залежності від опору ґрунту і напруги живлення в колі встановлюється певний струм I . На виводи мікроконтролера 3 надходить напруга від джерела струму U та з шунта $U_{ш}$.

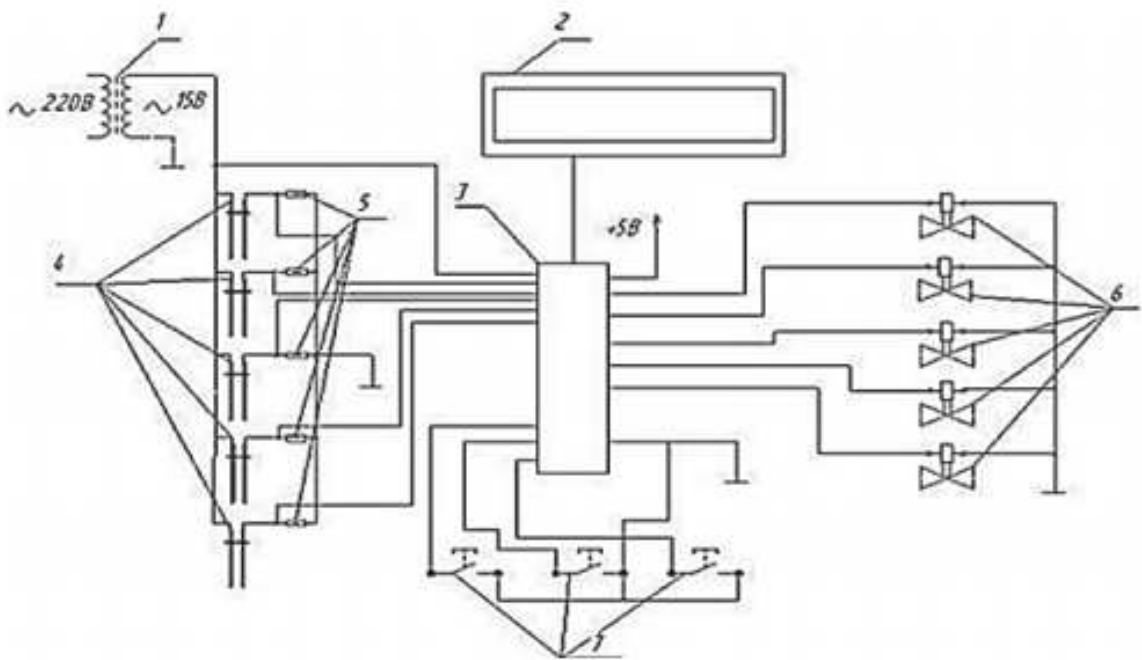


Рисунок 1.1 – Структурно функціональна схема пристрою керування крапельним поливом ґрунту : 1 – знижуючий трансформатор ТП115-10; 2 – РК-індикатор МТ-16S2D-2YLG; 3 – мікроконтролер АТmega16; 4 – резистивний датчик; 5 – шунт; 6 – електромагнітний клапан; 7 – кнопка керування

Аналізуючи цю систему можна виділити такі недоліки:

- резистивний датчик схильний до дострокового виходу із ладу через корозійну дію розчинів у ґрунті.
- використання дворівневої системи поливу, через використання такої системи стає можливим перерегулювання системи;

– відсутня залежність розрахунку та прогнозування зміни вологості від температури ґрунту.

1.3. Актуальні розробки на базі нечіткої логіки

Актуальними також є розробки системи автоматичного поливу, які засновані на базі нечіткої логіки. Нечітка логіка надає можливість створити залежність мінімум двох параметрів (температура та вологість ґрунту), що є на вході та за допомогою функцій керування отримати на виході певний, унікальний для конкретної комбінації вхідних сигналів, варіант ввімкнення або вимкнення системи (керування потужністю або часом роботи поливного насосу).

Дослідники з Катарського університету розробили систему, що базується на лінгвістичних змінних. Вони описують мікроконтролерну систему, яка зчитує стан вологості ґрунту, температуру навколишнього середовища та інтенсивність сонячної радіації. Система приймає ці три параметри як вхідні та, засновуючись на створеній таблиці правил, вона вмикається на необхідну тривалість поливу. Окрім цього система за допомогою безпроводного контролю ZigBee забезпечує передачу даних із датчиків та зберігає їх, час поливу та його тривалість, об'єм поливної води та можливість вмикання або вимикання системи самостійно. [2]

2. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

2.1. Методи вимірювання вологи та температури в ґрунті

Першим пунктом розробки системи автоматичного поливу є визначення методів, якими буде вимірюватись волога та температура в ґрунті.

2.1.1. Методи вимірювання вологості

На теперішній час використовуються доволі різні за принципом роботи методи вимірювання вологості ґрунту, які наведено в табл. 2.1 [3–7].

Таблиця 2.1 – Методи вимірювання вологості ґрунту

Метод	Описання методу
Гравіметричний метод	базуються на вилученні води зі зразка шляхом випаровування, вимивання або хімічної реакції.
Нейтронне розсіювання	середнє значення втрати енергії або термалізація нейтронів значно вищі, коли нейтрони стикаються з атомами з низькою атомною вагою, ніж коли з важчими атомами. В ґрунті атоми з малою вагою представлені переважно воднем. В результаті водень уповільнює швидкі нейтрони значно ефективніше, ніж будь-який інший елемент у ґрунті.
Згасання гамма-випромінювання	використовується для вимірювання вологості ґрунту в шарі 1-2 см. Ступінь зниження інтенсивності гамма-променів під час їх проходження через ґрунт залежить від складу ґрунту та його щільності.

Електромагнітний метод	базуються на впливі вологості на електричні властивості ґрунту.
Тензометричний метод	прилади здатні фіксувати зміни вологоємності ґрунту, що є наслідком інфільтрації води, поливу, випаровування та транспірації.
Мікрохвильовий метод	випромінювання тепла поверхнею ґрунту в мікрохвильовому діапазоні можна визначити дистанційно відповідними вимірювальними приладами – пасивними (радіометричними) або активними (радар) методами.
Ядерно магнітний резонанс	базується на здатності резонансу виявити концентрацію атомів водню і, відповідно, вологи.
Термічний метод	базується на зв'язку теплової інерції та вологості ґрунту.
Рефлектометричний метод	вимірює час проходження електричного імпульсу по кабелю, який залежить від електропровідності, а, отже, вологості, ґрунту.
Оптичний метод	базується на тому, що за наявності вологи на поверхні світло, відбите від неї, поляризується.
Діелькометричний	оснований на залежності діелектричних властивостей матеріалу від вологості. Оскільки для сухих речовин діелектрична проникність $\epsilon=2$, а для води $\epsilon= 81$, то невелика зміна вологості матеріалу призводить до значної зміни діелектричної проникності. Як вимірювальні кола в ємнісних вологомірах найчастіше використовуються трансформаторні мости з тісним індуктивним зв'язком плеч, а також резонансні кола.

Згідно з наведеним переліком методів, було обрано діелектричний метод. Датчики на його базі дозволяють контролювати показники вологості ґрунту безконтактно. Ємнісні датчики працюють шляхом швидкої зарядки і розрядки позитивного і заземленого електрода (конденсатора) в ґрунті, створюючи електромагнітне поле, час заряду якого пов'язане з ємністю ґрунту. Далі ємність пов'язана з діелектричної проникністю середовища між електродами конденсатора. На ємнісні датчики впливають вологість ґрунту, температура і електропровідність. Частота вибирається для вимірювання вологості ґрунту або масової електропровідності. Ємнісні датчики вимагають калібрування на місці для точного вимірювання об'ємного вмісту води в ґрунті (ОВВ %) [8, 9]. В цій роботі було використано ємнісний датчик вологості ґрунту від DFRobot [10], його характеристики наведено в табл. 2.2.

Таблиця 2.2 – Характеристики використовуваного датчика вологості DFRobot

Метод вимірювання вологості	Ємнісний
Тип виходу	Аналоговий
Напруга живлення	Від 3,3 В до 5,5 В
Вихідна напруга	Від 1,2 В до 2,5 В
Інтерфейсний роз'єм	PH 2.0-3P
Розмір	98x23 мм
Вага	15 г



Рисунок 2.1 – Ємнісний датчик вологості ґрунту від DFRobot

2.1.2. Методи вимірювання температури

Згідно з керівним документом «Настанова гідрометеорологічним станціям і постам» від Державної гідрометеорологічної служби 2011-го року визначають такі методи спостереження температури [11]:

– спостереження за станом підстильної поверхні проводить візуально. Стан поверхні ґрунту і снігового покриву оцінюють відповідною цифрою коду КС-01;

– температуру ґрунту й снігового покриву вимірюють за допомогою термометрів, що перебувають у постійному контакті з відповідним середовищем.

В свою чергу існують такі засоби вимірювання температури ґрунту (див. табл. 2.3).

Таблиця 2.3 – Засоби вимірювання температури ґрунту

Термометр	Діапазон вимірювання, °С	Примітка
ТМ1-1	-35/+50	Вимірювання максимальної температури підстильної поверхні
ТМ1-2	-20/+70	
ТМ2-1	-70/+20	Вимірювання мінімальної температури підстильної поверхні
ТМ2-2	-60/+30	
ТМ2-3	-50/+40	
ТМ3-1	-35/+60	Вимірювання температури поверхні ґрунту та снігового покриву
ТМ3-2	-25/+70	
ТМ3-3	-10/+85	
Колінчастий термометр ТМ5(Савінова)	-10/+50	Вимірювання температури ґрунту на оголеній ділянці
ТМ10	-5/+40 або -20/+30	Вимірювання температури ґрунту на глибинах на ділянці з природним покривом
Термометр опору	-50/+50	Вимірювання температури на глибинах і на поверхні ґрунту
Електронний термометр DS18B20	-55/+125	Вимірювання температури на глибинах і на поверхні ґрунту

Звісно це не єдині термометри, котрими можна б було вимірювати температуру ґрунту, але саме з цього переліку було обрано електронний термометр на базі сенсора DS18B20.

DS18B20 – це цифровий датчик температури, з дозволом перетворення 9-12 розрядів та функцією аварійного сигналу контролю температури. Датчик використовує інтерфейс 1-Wire, що розроблений виробником – компанією Dallas. Цей інтерфейс широко розповсюджений та доволі легкий у використанні. Доволі важливо ще те, що є можливість підключення декількох датчиків на одну шину даних. Живлення датчик може отримувати безпосередньо від лінії даних, без використання зовнішнього джерела живлення. В цьому випадку живлення датчика відбувається за рахунок енергії, що була запасена за допомогою «паразитичної» ємності. Діапазон вимірювання температури ставить від -55 до +125 °С. Для діапазону від -10 до +85 °С похибка не перевищує $\pm 0,5$ °С. Мікросхема 18B20 легко підключається до мікроконтролерних пристроїв на базі Arduino, AVR, PIC, ARM та ін. Окрім цього для роботи з Arduino існує готова бібліотека, що створена виробником.

Таблиця 2.4 - Характеристики використовуваного датчика DS18B20

Метод вимірювання температури	Електричний
Тип виходу	Цифровий
Діапазон вимірюваних температур	-55...+125 °С
Час отримання даних	750 мс при 12-бітному розширенні 94 мс при 9-бітному розширенні
Напруга живлення	3-5.5 В
Струм споживання в режимі очікування	750 нА
Струм споживання під час зчитування	1 мА
Інтерфейс зв'язку	1-Wire



Рисунок 2.2 – Електронний датчик вимірювання температури DS18B20

2.2. Створення структурної схеми системи автоматичного поливу

Структурна схема системи є наступною. Датчики температури та датчики вологості знаходяться в ґрунті. Вони зчитують необхідну інформацію та передають її в контролерну частину системи. В якості контролера виступає мікроконтролер Arduino UNO з запрограмованим алгоритмом на базі нечіткої логіки. Наступним кроком є обробка інформації за алгоритмом та видача керуючого сигналу на насос, котрий, в свою чергу, буде вмикатись та поливати ґрунт з необхідною тривалістю й інтенсивністю.

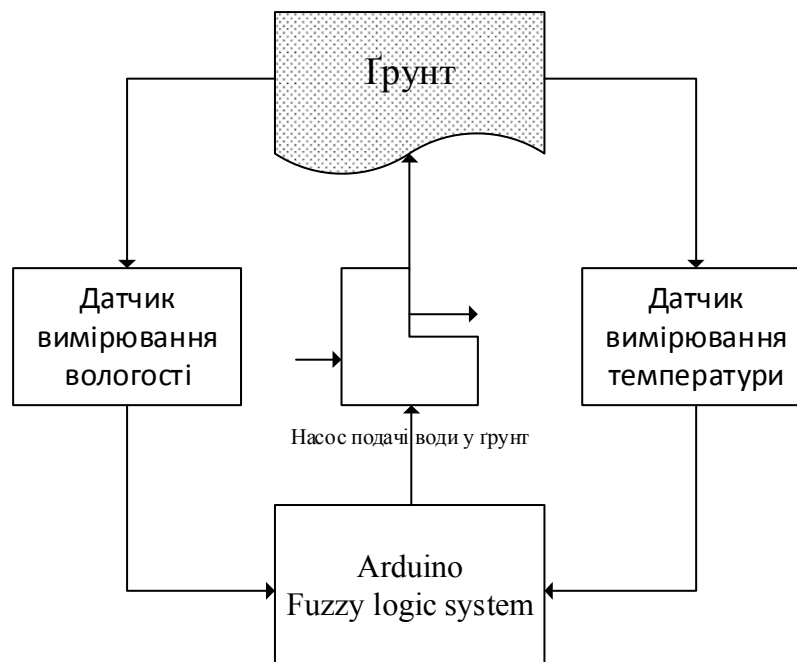


Рисунок 2.3 – Структурна схема автоматичної системи поливу

2.3. Алгоритм роботи автоматичного поливу на базі нечіткої логіки

Нечітка логіка (Fuzzy logic) дає змогу створити систему множин варіантів керування системою. Вона складається з блоків вхідних сигналів, бази правил керування та вихідного сигналу. За допомогою пакету програмного забезпечення MATLAB & SIMULINK та розширення Fuzzy Logic Toolbox створено автоматичну та повністю автономну систему управління поливом у теплиці. Дана система залежить від двох вхідних параметрів, що поділені на п'ять рівнів трикутних та трапецеїдальних вхідних функцій вологості та температури ґрунту та п'ять рівнів вихідних функцій керуванням насосом, що подає воду в систему поливу. Рівні поділяються згідно з державними нормами вирощування сільськогосподарських культур в теплиці [11].

2.3.1. Блок вхідних даних

Цей блок поділяється на дві частини:

– перша частина – це вхідні данні з датчику температури, діапазон встановлюється від 10 до 30 °С. У разі, якщо реальна температура нижче або вище заданого діапазону, у блоці сигналу приймається відповідно нижча або вища границя;

– друга частина – це вхідні данні з датчику вологості, діапазон встановлюється від 50 до 95 %. У разі якщо реальна вологість виходить за межі діапазону, то виконується така ж операція як і в першій частині.

В свою чергу діапазони поділяються на 5 відрізків, як показано на рис. 2.4.

На рисунку 2.4 – зображений блок для налаштування вхідних даних, а саме даних з датчика температури. Також на рисунку можна побачити чисельні дані пояснення, до яких знаходяться в табл. 2.5.

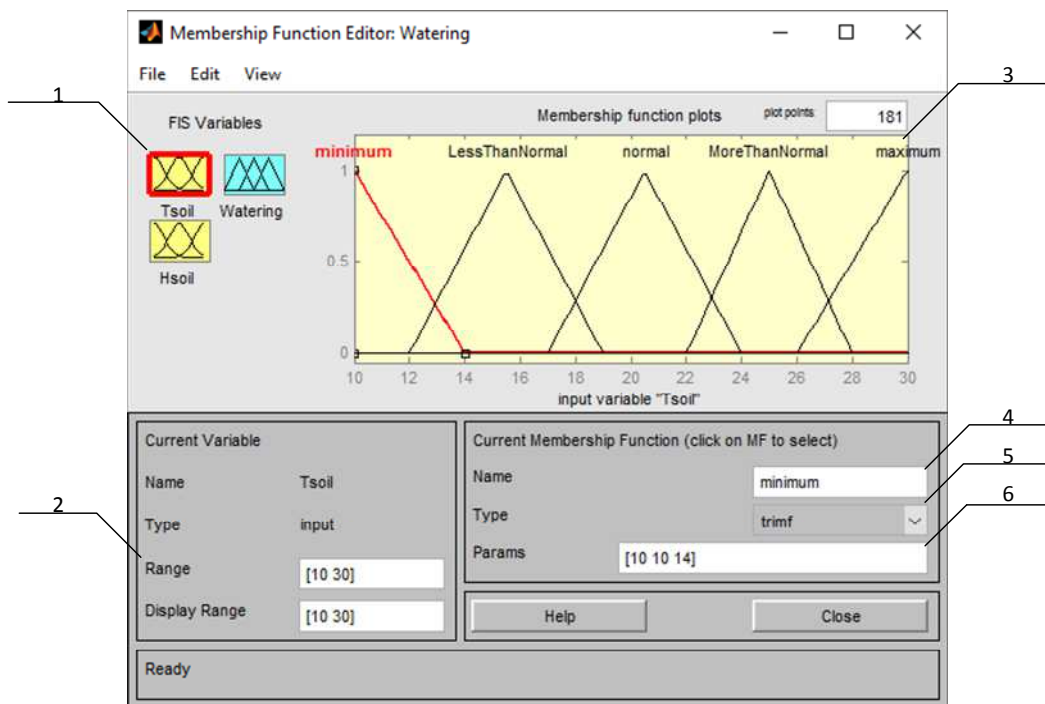


Рисунок 2.4 – Блок вхідних даних для датчика температури

Таблиця 2.5 – Пояснення конфігурацій для налаштування блоку вхідного сигналу

Номер позначення	Пояснення
1	Кнопка за допомогою якої обирається вікно налаштування вхідного сигналу з датчика температури
2	Діапазон можливих вхідних сигналів
3	Вікно відображення створених рівнів за якими буде реалізовуватись розподілення зазначеннями сигналу
4	Вікно перейменування функції рівня
5	Кнопка для вибору типу функції рівня
6	Вікно діапазону рівня функції

Подібну конструкцію має блок налаштування вхідних даних для датчика вологості (див. рис. 2.5).

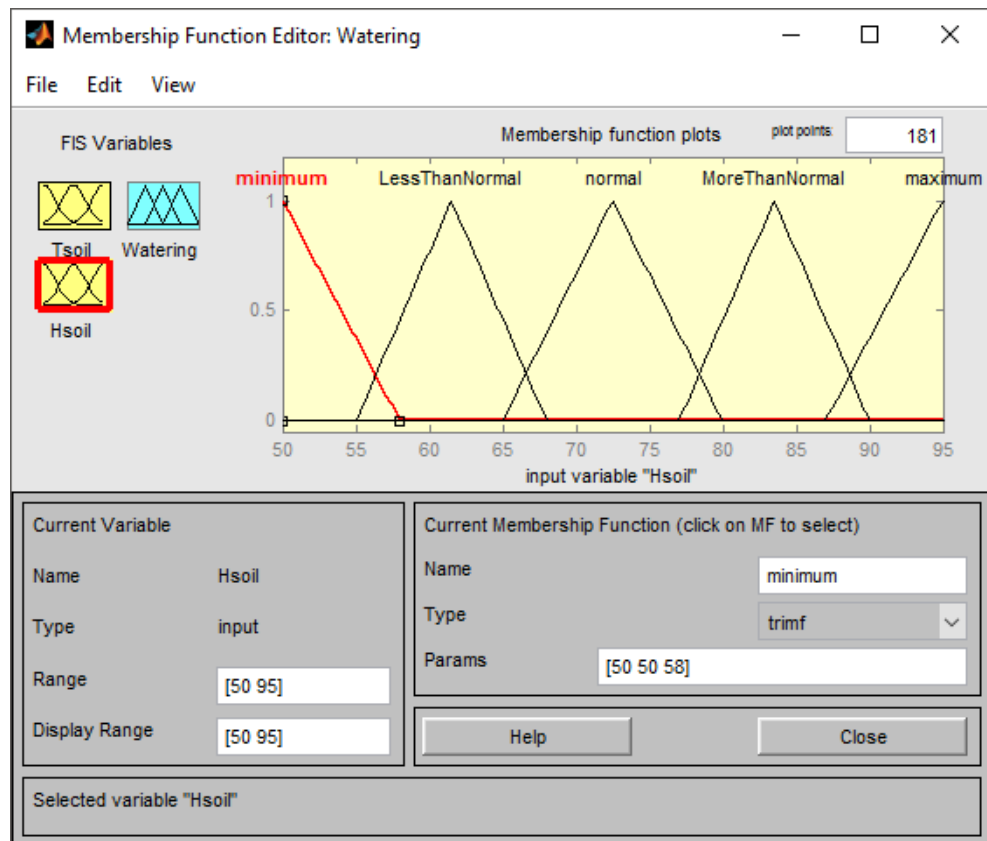


Рисунок 2.5 - Блок вхідних даних для датчика вологості

Таблиця 2.6 – Рівні вхідних сигналів

Рівень	Minimum	Less than normal	normal	More than normal	Maximum
Температура (T_{soil}), °C	10-14	12-19	17-24	22-28	26-30
Вологість (H_{soil}), 5	50-58	55-68	65-80	77-90	87-90

2.3.2. Блок вихідних даних

Блок вихідних даних складається з однієї частини. В ньому задаються функції керування вихідним сигналом, а саме напругою вихідного сигналу. За допомогою зміни напруги змінюється інтенсивність роботи насоса поливу ґрунту у теплиці.

Напруга змінюється в діапазоні від 0 до 5. В самому діапазоні, як і в блоці вхідних даних є п'ять рівнів, як показано в табл. 2.7.

Таблиця 2.7 –Рівні вихідного сигналу

Рівень	Off	low	medium	fast	intensely
Інтенсивність роботи	0-1	0,75-2	1,75-3,25	3-4,25	4-5

Для того, щоб забезпечити більш плавне переключення між рівнями було використане накладання крайніх значень відрізків. Наприклад, для того, щоб переключитися з рівня «low» на рівень «medium» вихідному достатньо бути в точці «1,86». Такі ж самі правила переходу між рівнями поширюються і на вхідні блоки.

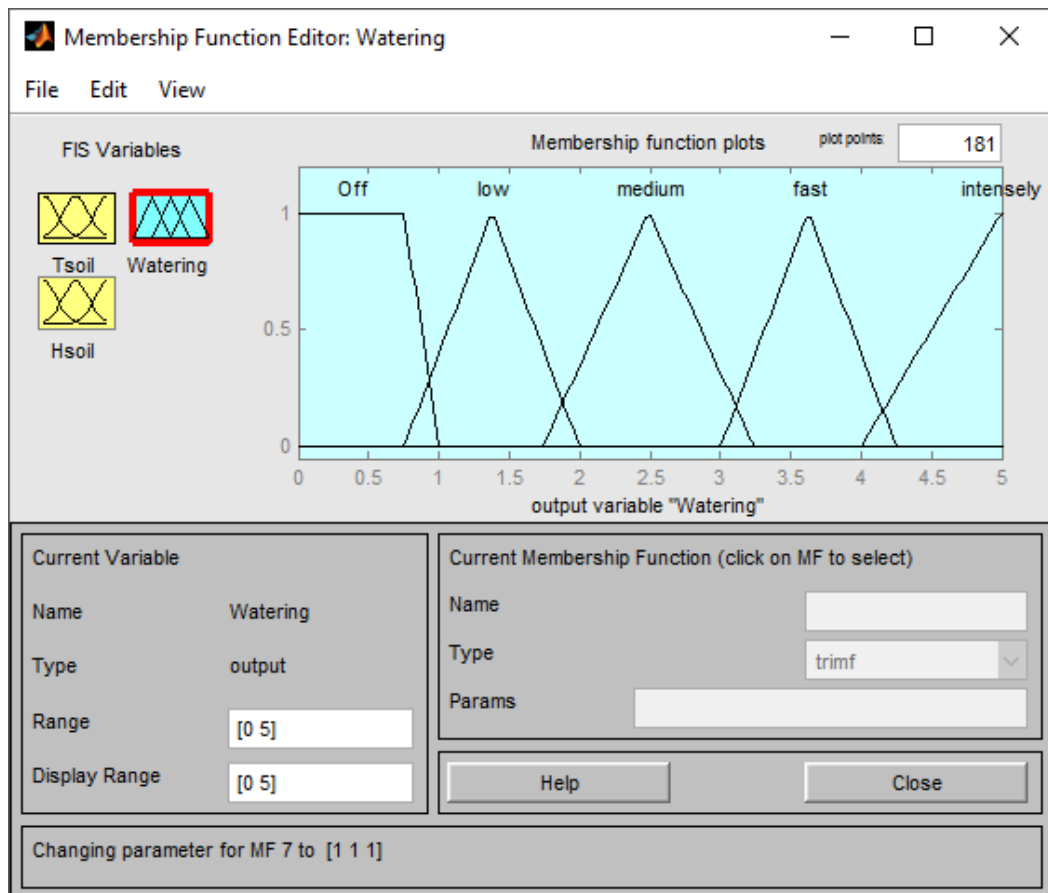


Рисунок 2.6 – Блок вихідного сигналу для керуванням насосом

2.3.3. Блок створення бази правил

Основним виконуючим блоком даної системи є блок бази правил, іншими словами «Редактор правил». За його допомогою інженер або проектувальник створює функції залежності вихідного сигналу/сигналів від вхідних сигналів.

В залежності від миттєвих вхідних сигналів в редакторі правил (див. рис. 2.7) обирається відповідне рівняння функції керування. Для кращого уявлення цієї системи керування наведена матриця відповідності правил до функцій належності параметрів (див. табл. 2.8) [11].

1. If (Tsoil is minimum) and (Hsoil is minimum) then (Watering is fast) (1)
2. If (Tsoil is minimum) and (Hsoil is LessThanNormal) then (Watering is medium) (1)
3. If (Tsoil is minimum) and (Hsoil is maximum) then (Watering is Off) (1)
4. If (Tsoil is minimum) and (Hsoil is normal) then (Watering is Off) (1)
5. If (Tsoil is minimum) and (Hsoil is MoreThanNormal) then (Watering is Off) (1)
6. If (Tsoil is minimum) and (Hsoil is maximum) then (Watering is Off) (1)
7. If (Tsoil is LessThanNormal) and (Hsoil is minimum) then (Watering is intensely) (1)
8. If (Tsoil is LessThanNormal) and (Hsoil is LessThanNormal) then (Watering is fast) (1)
9. If (Tsoil is LessThanNormal) and (Hsoil is normal) then (Watering is Off) (1)
10. If (Tsoil is LessThanNormal) and (Hsoil is MoreThanNormal) then (Watering is Off) (1)
11. If (Tsoil is LessThanNormal) and (Hsoil is maximum) then (Watering is Off) (1)
12. If (Tsoil is normal) and (Hsoil is minimum) then (Watering is intensely) (1)
13. If (Tsoil is normal) and (Hsoil is LessThanNormal) then (Watering is fast) (1)
14. If (Tsoil is normal) and (Hsoil is normal) then (Watering is Off) (1)
15. If (Tsoil is normal) and (Hsoil is MoreThanNormal) then (Watering is Off) (1)
16. If (Tsoil is normal) and (Hsoil is maximum) then (Watering is Off) (1)
17. If (Tsoil is MoreThanNormal) and (Hsoil is minimum) then (Watering is intensely) (1)
18. If (Tsoil is MoreThanNormal) and (Hsoil is LessThanNormal) then (Watering is fast) (1)
19. If (Tsoil is MoreThanNormal) and (Hsoil is normal) then (Watering is Off) (1)
20. If (Tsoil is MoreThanNormal) and (Hsoil is MoreThanNormal) then (Watering is Off) (1)
21. If (Tsoil is MoreThanNormal) and (Hsoil is maximum) then (Watering is Off) (1)
22. If (Tsoil is maximum) and (Hsoil is minimum) then (Watering is intensely) (1)
23. If (Tsoil is maximum) and (Hsoil is LessThanNormal) then (Watering is intensely) (1)
24. If (Tsoil is maximum) and (Hsoil is normal) then (Watering is low) (1)
25. If (Tsoil is maximum) and (Hsoil is MoreThanNormal) then (Watering is low) (1)
26. If (Tsoil is maximum) and (Hsoil is maximum) then (Watering is Off) (1)

Рисунок 2.7 – Редактор правил обробки вхідних сигналів

Таблиця 2.8 – Матриця відповідності правил до функцій належності параметрів

T \ W	$T_1(10-14^{\circ}\text{C})$	$T_2(12 - 19^{\circ}\text{C})$	$T_3(17 - 24^{\circ}\text{C})$	$T_4(22 - 28^{\circ}\text{C})$	$T_5(26 - 30^{\circ}\text{C})$
$W_1(50 - 58\%)$	<i>fast</i>	<i>intensely</i>	<i>intensely</i>	<i>intensely</i>	<i>intensely</i>
$W_2(55 - 68\%)$	<i>medium</i>	<i>fast</i>	<i>fast</i>	<i>fast</i>	<i>intensely</i>
$W_3(65 - 80\%)$	<i>OFF</i>	<i>OFF</i>	<i>OFF</i>	<i>OFF</i>	<i>low</i>
$W_4(77 - 90\%)$	<i>OFF</i>	<i>OFF</i>	<i>OFF</i>	<i>OFF</i>	<i>low</i>

2.4. Створення теоретичної моделі системи автоматичного поливу

Для того, щоб протестувати створену модель скористаємося пакету програмного забезпечення MATLAB & SIMULINK.

У якості вхідних сигналів було використано генератори випадкового сигналу амплітуда, яких відповідає діапазону температур та відсотка вологості на відповідних датчиках. Для більшого приближення до реальних умов на кожен із сигналів додатково підключено генератор шуму сигнал, якого відповідає допустимій похибці на датчиках.

Далі вхідні сигнали надходять до мікроконтролера з встановленим програмним забезпеченням нечіткої логіки, що було описано в попередньому пункті роботи. Сигнали оброблюються та мікроконтролер надає відповідний вихідний сигнал який керує потужністю роботи насосу.

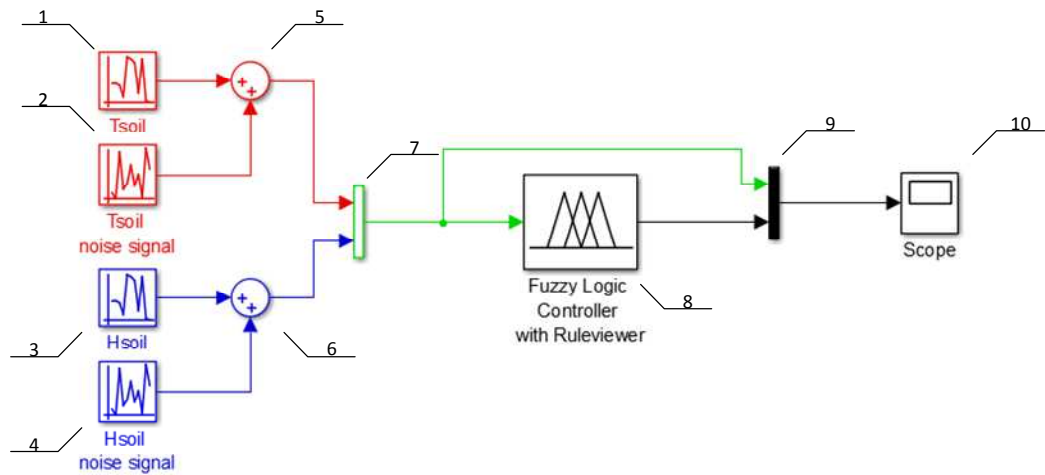


Рисунок 2.8 – Теоретична модель автоматичного поливу ґрунту

На рисунку можна побачити елементи що складають дану модель, як показано в табл. 2.9.

Таблиця 2.9 – Складові теоретичної моделі

Номер позначення	Пояснення
1	Генератор випадкового сигналу (датчик температури)
2	Генератор шуму
3	Генератор випадкового сигналу (датчик вологості)
4	Генератор шуму
5	Суматор сигналів, накладає шум на «чистий» сигнал з датчика температури
6	Суматор сигналів, накладає шум на «чистий» сигнал з датчика вологості
7	Мультиплексор вхідних сигналів
8	Контролер нечіткої логіки, в ньому оброблюються вхідні сигнали та видають керуючий сигнал
9	Мультиплексор вихідного сигналу та вхідного для кращого порівняння результату
10	Осцилограф, наглядно демонструє залежність сигналів та принцип роботи нечіткої логіки

В даній моделі насос замінено на осцилограф який відображує вихідний сигнал, а також додатково до осцилографа під'єднано вхідні сигнали від генераторів, це допоможе наглядно побачити залежність вихідного сигналу від вхідного.

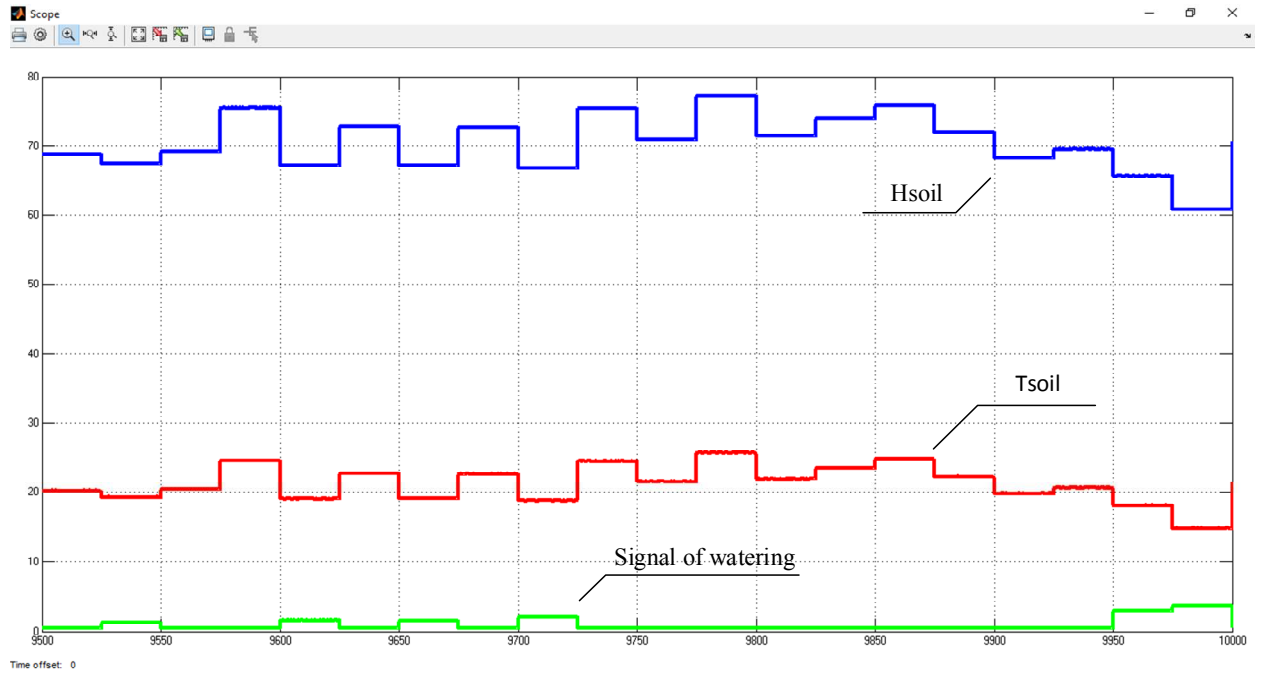
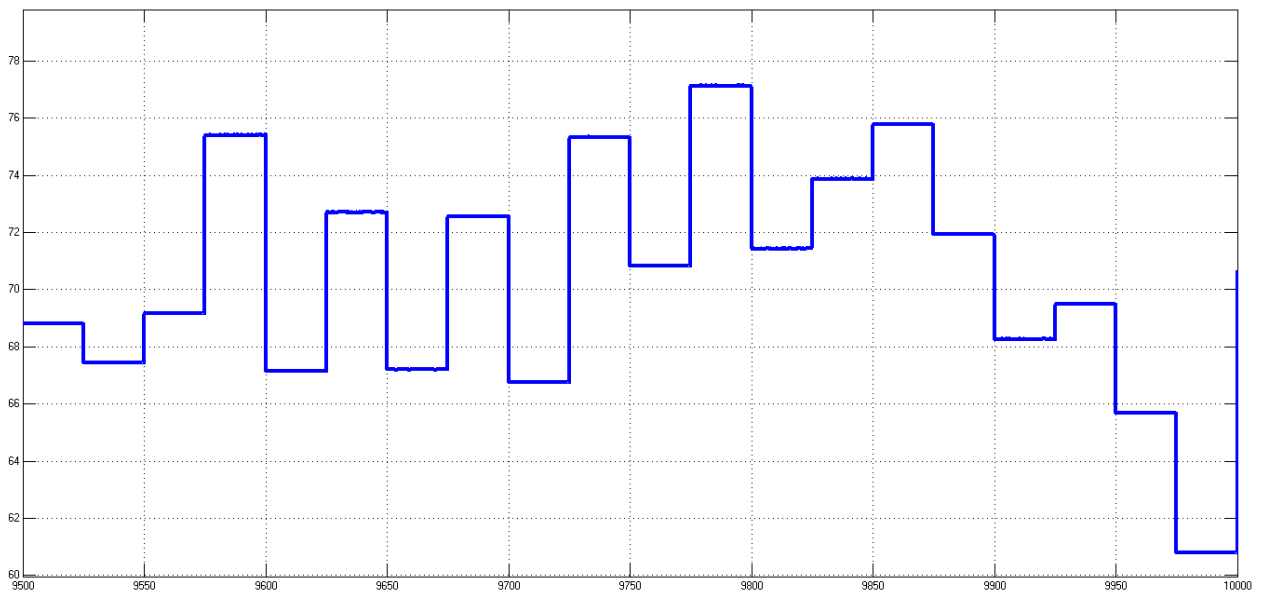
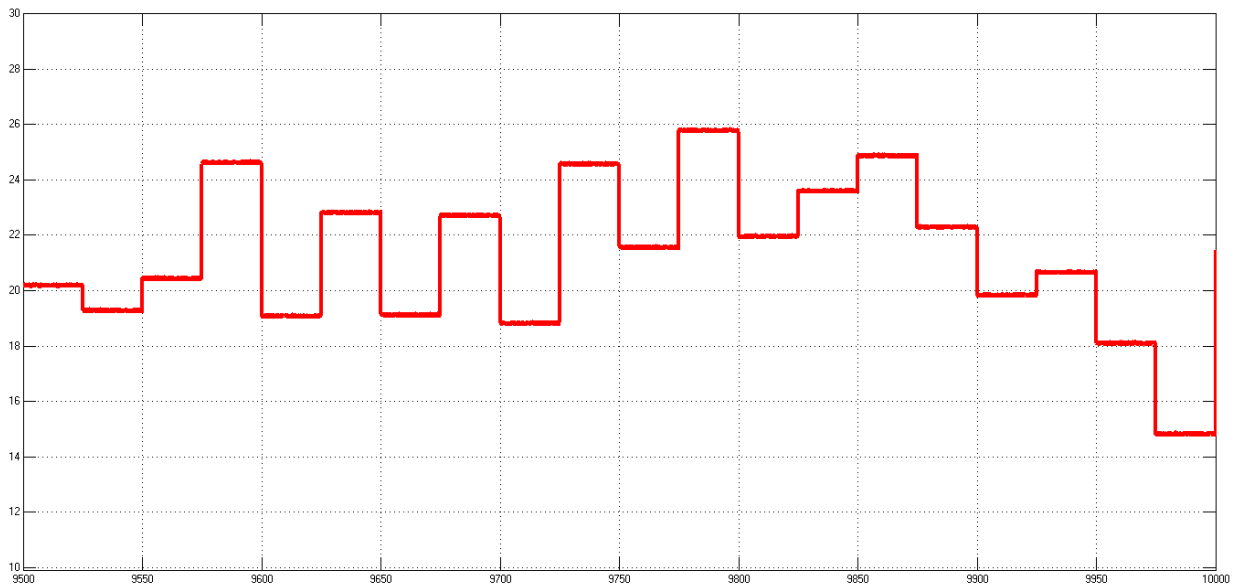


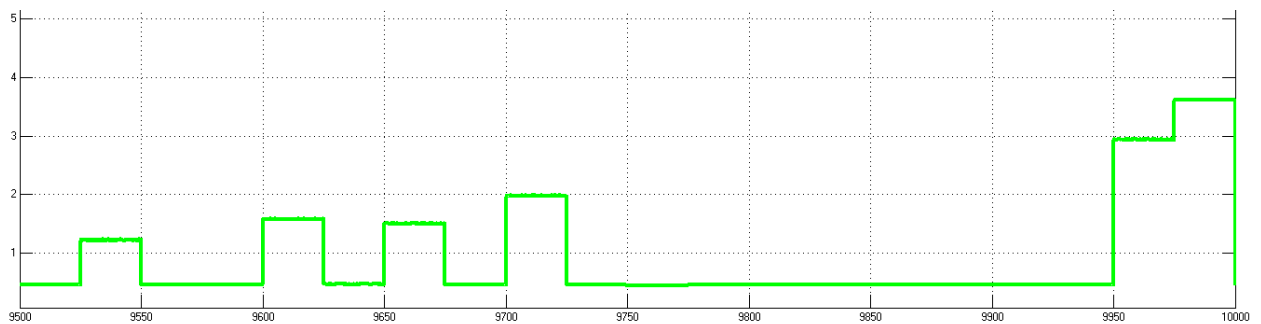
Рисунок 2.9 – Осцилограма залежності сигналів створеної системи



a)



б)



в)

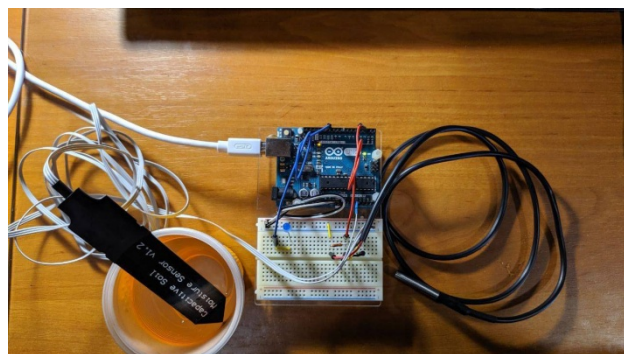
Рисунок 2.9 – Окремо відображені осцилограми сигналів створеної системи: а) сигнал з датчика вологості, б) сигнал з датчика температури, в) вихідний сигнал

2.5. Створення реальної моделі на базі мікроконтролера Arduino

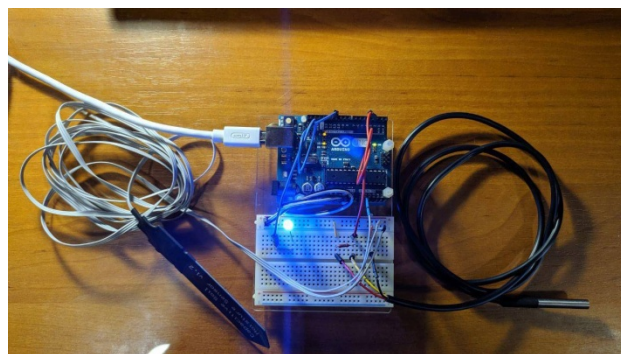
Перш за все необхідно створити програмний код, який буде основою обробки сигналів в даній системі. Код було створено за допомогою онлайн сервісу для конвертації файлів типу «.fis» в «.ino». Цей конвертер допомагає майже миттєво отримати робочу програму здатну працювати в програмі «Arduino.ide» [12]. Після того, як було отримано конвертований, за допомогою сервісу, код змінюємо його відповідно до необхідних для використання

датчиків та виводів мікропроцесора, що використовуються в контролері. Лістинг програми наведено в Додатку А.

В лабораторних умовах було створено експериментальну модель системи, робочий вигляд, якої наведено на рисунку 2.10. У якості насоса було використано світлодіод потужність та яскравість, якого залежить від керуючого сигналу контролера. Діапазон потужностей було описано в пункті 2.3.2.



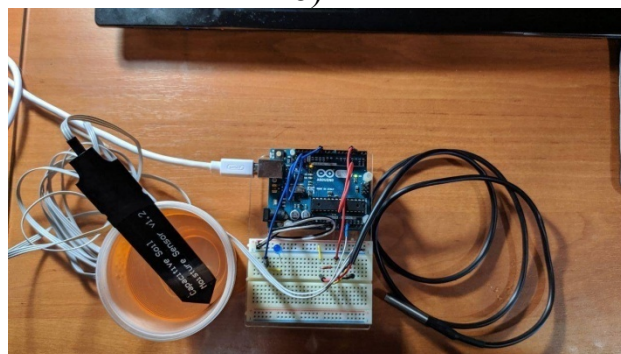
а)



б)



в)



г)

Рисунок 2.10 – Експериментальна модель автоматичної системи поливу:
а, г – світлодіод (полив) вимкнено, так як вологість та температура в нормі;
б, в – світлодіод ввімкнено, так як недостатньо вологості або висока температура

Окрім відображення режиму роботи за допомогою світлодіода також можна спостерігати за допомогою послідовного порту. На ньому відображується значення температури та вологості з датчиків, напруга на світлодіоді та режим роботи, який відповідає функції системи нечіткої логіки, як показано на рис. 2.11.

T: 16.56	H:79.00	Out = 0.47	I = 0	Voltage = 0.00	T: 19.19	H:72.00	Out = 0.45	I = 0	Voltage = 0.00
T: 16.62	H:79.00	Out = 0.47	I = 0	Voltage = 0.00	T: 19.19	H:72.00	Out = 0.45	I = 0	Voltage = 0.00
T: 16.69	H:79.00	Out = 0.47	I = 0	Voltage = 0.00	T: 19.25	H:72.00	Out = 0.45	I = 0	Voltage = 0.00
T: 16.69	H:79.00	Out = 0.47	I = 0	Voltage = 0.00	T: 19.31	H:72.00	Out = 0.45	I = 0	Voltage = 0.00
T: 16.75	H:79.00	Out = 0.47	I = 0	Voltage = 0.00	T: 19.25	H:72.00	Out = 0.45	I = 0	Voltage = 0.00
T: 16.81	H:79.00	Out = 0.47	I = 0	Voltage = 0.00	T: 19.25	H:72.00	Out = 0.45	I = 0	Voltage = 0.00
T: 16.87	H:79.00	Out = 0.47	I = 0	Voltage = 0.00	T: 19.31	H:72.00	Out = 0.45	I = 0	Voltage = 0.00
T: 16.87	H:79.00	Out = 0.47	I = 0	Voltage = 0.00	T: 19.44	H:72.00	Out = 0.44	I = 0	Voltage = 0.00
T: 16.94	H:79.00	Out = 0.47	I = 0	Voltage = 0.00	T: 19.37	H:72.00	Out = 0.45	I = 0	Voltage = 0.00
T: 17.00	H:79.00	Out = 0.47	I = 0	Voltage = 0.00	T: 19.37	H:72.00	Out = 0.45	I = 0	Voltage = 0.00
T: 17.00	H:79.00	Out = 0.47	I = 0	Voltage = 0.00	T: 19.44	H:72.00	Out = 0.44	I = 0	Voltage = 0.00
T: 17.06	H:79.00	Out = 0.47	I = 0	Voltage = 0.00	T: 19.37	H:72.00	Out = 0.45	I = 0	Voltage = 0.00
T: 17.12	H:79.00	Out = 0.47	I = 0	Voltage = 0.00	T: 19.37	H:72.00	Out = 0.45	I = 0	Voltage = 0.00
T: 17.19	H:79.00	Out = 0.47	I = 0	Voltage = 0.00	T: 19.44	H:72.00	Out = 0.44	I = 0	Voltage = 0.00
T: 17.12	H:79.00	Out = 0.47	I = 0	Voltage = 0.00	T: 19.44	H:72.00	Out = 0.44	I = 0	Voltage = 0.00
T: 17.25	H:79.00	Out = 0.47	I = 0	Voltage = 0.00	T: 19.44	H:72.00	Out = 0.44	I = 0	Voltage = 0.00
T: 17.25	H:79.00	Out = 0.47	I = 0	Voltage = 0.00	T: 19.50	H:71.00	Out = 0.44	I = 0	Voltage = 0.00
T: 17.25	H:79.00	Out = 0.47	I = 0	Voltage = 0.00	T: 19.56	H:71.00	Out = 0.44	I = 0	Voltage = 0.00
T: 27.69	H:57.00	Out = 4.29	I = 204	Voltage = 4.00	T: 14.00	H:94.00	Out = 0.45	I = 0	Voltage = 0.00
T: 27.56	H:56.00	Out = 4.17	I = 204	Voltage = 4.00	T: 14.06	H:94.00	Out = 0.45	I = 0	Voltage = 0.00
T: 27.56	H:55.00	Out = 4.60	I = 204	Voltage = 4.00	T: 14.31	H:94.00	Out = 0.45	I = 0	Voltage = 0.00
T: 27.37	H:55.00	Out = 4.59	I = 204	Voltage = 4.00	T: 14.50	H:94.00	Out = 0.44	I = 0	Voltage = 0.00
T: 27.44	H:55.00	Out = 4.60	I = 204	Voltage = 4.00	T: 14.56	H:95.00	Out = 0.44	I = 0	Voltage = 0.00
T: 27.50	H:55.00	Out = 4.60	I = 204	Voltage = 4.00	T: 14.63	H:95.00	Out = 0.44	I = 0	Voltage = 0.00
T: 27.87	H:55.00	Out = 4.60	I = 204	Voltage = 4.00	T: 14.75	H:95.00	Out = 0.44	I = 0	Voltage = 0.00
T: 28.75	H:56.00	Out = 4.57	I = 204	Voltage = 4.00	T: 14.88	H:95.00	Out = 0.44	I = 0	Voltage = 0.00
T: 29.12	H:57.00	Out = 4.59	I = 204	Voltage = 4.00	T: 15.00	H:95.00	Out = 0.44	I = 0	Voltage = 0.00
T: 29.12	H:55.00	Out = 4.60	I = 204	Voltage = 4.00	T: 15.06	H:95.00	Out = 0.43	I = 0	Voltage = 0.00
T: 28.94	H:55.00	Out = 4.60	I = 204	Voltage = 4.00	T: 15.13	H:95.00	Out = 0.43	I = 0	Voltage = 0.00
T: 28.75	H:55.00	Out = 4.60	I = 204	Voltage = 4.00	T: 15.25	H:95.00	Out = 0.43	I = 0	Voltage = 0.00
T: 28.37	H:55.00	Out = 4.60	I = 204	Voltage = 4.00	T: 15.44	H:95.00	Out = 0.43	I = 0	Voltage = 0.00
T: 28.25	H:55.00	Out = 4.60	I = 204	Voltage = 4.00	T: 15.50	H:95.00	Out = 0.43	I = 0	Voltage = 0.00
T: 28.06	H:57.00	Out = 4.59	I = 204	Voltage = 4.00	T: 15.56	H:95.00	Out = 0.43	I = 0	Voltage = 0.00
T: 28.00	H:57.00	Out = 4.59	I = 204	Voltage = 4.00	T: 15.63	H:95.00	Out = 0.43	I = 0	Voltage = 0.00
T: 28.00	H:55.00	Out = 4.60	I = 204	Voltage = 4.00	T: 15.69	H:95.00	Out = 0.43	I = 0	Voltage = 0.00

T: 12.25	H:55.00	Out = 3.76	I = 153	Voltage = 3.00
T: 12.19	H:57.00	Out = 2.93	I = 102	Voltage = 2.00
T: 12.13	H:57.00	Out = 2.89	I = 102	Voltage = 2.00
T: 12.06	H:57.00	Out = 2.84	I = 102	Voltage = 2.00
T: 11.94	H:55.00	Out = 3.63	I = 153	Voltage = 3.00
T: 11.88	H:55.00	Out = 3.63	I = 153	Voltage = 3.00
T: 11.88	H:57.00	Out = 2.79	I = 102	Voltage = 2.00
T: 12.00	H:56.00	Out = 3.13	I = 153	Voltage = 3.00
T: 12.06	H:55.00	Out = 3.66	I = 153	Voltage = 3.00
T: 12.31	H:57.00	Out = 3.02	I = 153	Voltage = 3.00
T: 12.63	H:57.00	Out = 3.14	I = 153	Voltage = 3.00
T: 12.75	H:55.00	Out = 3.97	I = 153	Voltage = 3.00
T: 12.94	H:55.00	Out = 4.05	I = 204	Voltage = 4.00
T: 13.19	H:55.00	Out = 4.17	I = 204	Voltage = 4.00
T: 13.31	H:57.00	Out = 3.39	I = 153	Voltage = 3.00
T: 13.50	H:55.00	Out = 4.30	I = 204	Voltage = 4.00
T: 13.69	H:55.00	Out = 4.38	I = 204	Voltage = 4.00
T: 13.81	H:55.00	Out = 4.46	I = 204	Voltage = 4.00
T: 13.94	H:56.00	Out = 4.07	I = 204	Voltage = 4.00

Рисунок 2.11 – Параметри, що надходять до послідовного порту від датчиків та вихідний сигнал що надходить до світлодіоду

ВИСНОВКИ ТА ПЕРСПЕКТИВНІ НАПРЯМКИ ДОСЛІДЖЕНЬ

За експериментальними та теоретичними випробуваннями видно, що поставлені на початку роботи завдання виконані. Система автоматично обирає найближчий варіант роботи, що відповідає заздалегідь створеній функції. Для того, щоб система працювала більш плавно – необхідно додати більше рівнів у блоках входу та виходу, відповідно до цього необхідно створити більшу кількість функцій залежності в блоці редагування.

Дану систему можна поширювати не тільки для автоматизації поливу ґрунту в теплиці, також автоматичне керування за допомогою нечіткого управління можна реалізовувати в багатьох сферах та галузях виробництва.

Подальшими планами розробки цієї системи є експериментальні випробування системи на довгочасному вирощуванні культурних рослин, дослідження градієнту розповсюдження вологи в ґрунті з різним відсотком вологості. Крім того дану систему можна вдосконалити та позбутися дротового способу передавання сигналу, замінивши його на бездротовий зв'язок за допомогою модулів ZigBee.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Автоматизація керування крапельним поливом тепличних культур / (Горячев С. В., Козловцев А. Л., Сорокін А. А. та ін.); - К.: Вістник Оренбургського державного аграрного університету, 2017 – 133 с.
2. Farid Touati, M Al-Hitmi, Kamel Benhmed (2012) Sistema upravlinnya zroshennyam na osnovi nechitkoyi lohiky v posushlyvykh rehionakh Derzhavy Katar [A fuzzy logic based irrigation management system in arid regions applied to the State of Qatar]. WIT Transactions on Ecology and the Environment (electronic journal), pp. 189-199. Available at: https://www.researchgate.net/publication/253234402_A_fuzzy_logic_based_irrigation_management_system_in_arid_regions_applied_to_the_State_of_Qatar (accessed 21 October 2019).
3. Методи вимірювання вологості ґрунту[електронний ресурс] /Богдан Малиновський // – 2018. – Режим доступу до ресурсу: <https://propozitsiya.com/ua/metody-vymiryuvannya-vologosti-gruntu>
4. Методи і засоби агрометеорологічних вимірювань параметрів ґрунтів[електронний ресурс] // Режим доступу до ресурсу: <https://uhmi.org.ua/rozz/agro/index.php#Z1>
5. Larry Pitts (2016) Monitorynh volohosti gruntu dlya optimal'noho zrostannya vrozhayu [Monitoring Soil Moisture for Optimal Crop Growth] (Electronic resource), Available at: <https://observant.zendesk.com/hc/en-us/articles/208067926-Monitoring-Soil-Moisture-for-Optimal-Crop-Growth>(accessed 10 November 2019).
6. Вимірювання вологості [електронний ресурс] // Режим доступу до ресурсу: https://studopedia.com.ua/1_125406_vimiryuvannya-vologosti.html
7. Лактіонов, І.С. Комп'ютеризована система комплексного моніторингу й керування мікрокліматом промислових теплиць на базі нечіткої логіки /

Лактіонов І.С., Вовна О.В., Бережний М.О., Лебедєв В.А.. – Кременчук: Вістник КрНУ імені Михайла Остроградського. Випуск 3/2019 (116). – 120 с.

8. Produkty vumiryuvannya volohosti - z-hidno do rehionu [Soils Moisture Products - Select Region] (Electronic resource), Available at: <http://www.ictinternational.com/products/soils/moisture-sensors/> (accessed 15 November 2019).

9. Sensor volohosti hruntu SM150T [SM150T Soil Moisture Sensor] (Electronic resource), Available at: <https://www.deltat.co.uk/product/sm150t/> (accessed 15 November 2019).

10. Ємнісний датчик вологості від DFRobot [електронний ресурс] // Режим доступу до ресурсу: <https://arduino.ua/prod3255-emkostnii-datchik-vlajnosti-pochvi-ot-dfrobot>.

11. Вовна, О.В. Комп'ютеризована технологія моніторингу й керування системою автоматичного поливу рослин у теплицях на базі нечіткої логіки / О.В. Вовна, І.С. Лактіонов, М.О. Бережний, В.А. Лебедєв // Перспективні напрямки сучасної електроніки, інформаційних і комп'ютерних систем: тези доповідей на Всеукр. наук.-практ. конф. MEICS-2019. – К. : м. Дніпро 2019 – С. 77 – 78.

12. Arduino FIST: Systema nechitkoji lohiky MATLAB dlya peretvorennya v Arduino C [Arduino FIST: MATLAB Fuzzy Inference System to Arduino C Converter] (Electronic resource), Available at: http://www.makeproto.com/projects/fuzzy/matlab_arduino_FIST/index.php (accessed 20 November 2019).

Додаток А – Лістинг програмного коду автоматичної системи на базі

Fuzzylogic

```
/**
 *
 */
#include "fis_header.h"
#include <OneWire.h>
#include <DallasTemperature.h>
#define ONE_WIRE_BUS 2
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
// Number of inputs to the fuzzy inference system
const int fis_gcI = 2;
// Number of outputs to the fuzzy inference system
const int fis_gcO = 1;
// Number of rules to the fuzzy inference system
const int fis_gcR = 26;
const int led = 11;
FIS_TYPE g_fisInput[fis_gcI];
FIS_TYPE g_fisOutput[fis_gcO];
// Setup routine runs once when you press reset:
void setup()
{
  Serial.begin(9600);
  // initialize the Analog pins for input.
  // Pin mode for Input: Tsoil
  sensors.begin();
  // Pin mode for Input: Hsoil
  pinMode(A1 , INPUT);
  // initialize the Analog pins for output.
  // Pin mode for Output: Poliv
  pinMode(led , OUTPUT);
}
void loop()
{
  sensors.requestTemperatures();
  g_fisInput[0] = sensors.getTempCByIndex(0);
  int H = analogRead(A1);
  g_fisInput[1] = map(H, 300, 584, 95, 50);
  g_fisOutput[0] = 0;
  fis_evaluate();
  int I = map(g_fisOutput[0], 5, 0, 255, 0);
  analogWrite(led,I);
  Serial.print("T: ");
  Serial.print(g_fisInput[0]);
  Serial.print(" H:");
  Serial.print(g_fisInput[1]);
  Serial.print(" Out = ");
}
```

```

Serial.print(g_fisOutput[0]);
Serial.print(" I = ");
Serial.print(I);
Serial.print(" Voltage = ");
Serial.println(I*0.0196);
delay(500);
// Set output vlaue: Poliv
analogWrite(2 , g_fisOutput[0]);
}
//*****
// Support functions for Fuzzy Inference System
//*****
// Triangular Member Function
FIS_TYPE fis_trimf(FIS_TYPE x, FIS_TYPE* p)
{
    FIS_TYPE a = p[0], b = p[1], c = p[2];
    FIS_TYPE t1 = (x - a) / (b - a);
    FIS_TYPE t2 = (c - x) / (c - b);
    if ((a == b) && (b == c)) return (FIS_TYPE) (x == a);
    if (a == b) return (FIS_TYPE) (t2*(b <= x)*(x <= c));
    if (b == c) return (FIS_TYPE) (t1*(a <= x)*(x <= b));
    t1 = min(t1, t2);
    return (FIS_TYPE) max(t1, 0);
}
// Trapezoidal Member Function
FIS_TYPE fis_trapmf(FIS_TYPE x, FIS_TYPE* p)
{
    FIS_TYPE a = p[0], b = p[1], c = p[2], d = p[3];
    FIS_TYPE t1 = ((x <= c) ? 1 : ((d < x) ? 0 : ((c != d) ? ((d - x) / (d - c)) : 0)));
    FIS_TYPE t2 = ((b <= x) ? 1 : ((x < a) ? 0 : ((a != b) ? ((x - a) / (b - a)) : 0)));
    return (FIS_TYPE) min(t1, t2);
}
FIS_TYPE fis_min(FIS_TYPE a, FIS_TYPE b)
{
    return min(a, b);
}
FIS_TYPE fis_max(FIS_TYPE a, FIS_TYPE b)
{
    return max(a, b);
}
FIS_TYPE fis_array_operation(FIS_TYPE *array, int size, _FIS_ARR_OP pfnOp)
{
    int i;
    FIS_TYPE ret = 0;
    if (size == 0) return ret;
    if (size == 1) return array[0];
    ret = array[0];
    for (i = 1; i < size; i++)

```

```

    {
        ret = (*pfnOp)(ret, array[i]);
    }
    return ret;
}
//*****
// Data for Fuzzy Inference System
//*****
// Pointers to the implementations of member functions
_FIS_MF fis_gMF[] =
{
    fis_trimf, fis_trapmf
};
// Count of member function for each Input
int fis_gIMFCount[] = { 5, 5 };
// Count of member function for each Output
int fis_gOMFCount[] = { 5 };
// Coefficients for the Input Member Functions
FIS_TYPE fis_gMFI0Coeff1[] = { 10, 10, 14 };
FIS_TYPE fis_gMFI0Coeff2[] = { 17, 20.5, 24 };
FIS_TYPE fis_gMFI0Coeff3[] = { 26, 30, 30 };
FIS_TYPE fis_gMFI0Coeff4[] = { 12, 15.5, 19 };
FIS_TYPE fis_gMFI0Coeff5[] = { 22, 25, 28 };
FIS_TYPE* fis_gMFI0Coeff[] = { fis_gMFI0Coeff1, fis_gMFI0Coeff2, fis_gMFI0Coeff3,
    fis_gMFI0Coeff4, fis_gMFI0Coeff5 };
FIS_TYPE fis_gMFI1Coeff1[] = { 50, 50, 58 };
FIS_TYPE fis_gMFI1Coeff2[] = { 65, 72.5, 80 };
FIS_TYPE fis_gMFI1Coeff3[] = { 87, 95, 95 };
FIS_TYPE fis_gMFI1Coeff4[] = { 55, 61.5, 68 };
FIS_TYPE fis_gMFI1Coeff5[] = { 77, 83.5, 90 };
FIS_TYPE* fis_gMFI1Coeff[] = { fis_gMFI1Coeff1, fis_gMFI1Coeff2, fis_gMFI1Coeff3,
    fis_gMFI1Coeff4, fis_gMFI1Coeff5 };
FIS_TYPE** fis_gMFI0Coeff[] = { fis_gMFI0Coeff, fis_gMFI1Coeff };
// Coefficients for the Output Member Functions
FIS_TYPE fis_gMFO0Coeff1[] = { 0, 0, 0.75, 1 };
FIS_TYPE fis_gMFO0Coeff2[] = { 0.75, 1.375, 2 };
FIS_TYPE fis_gMFO0Coeff3[] = { 1.75, 2.5, 3.25 };
FIS_TYPE fis_gMFO0Coeff4[] = { 3, 3.625, 4.25 };
FIS_TYPE fis_gMFO0Coeff5[] = { 4, 5, 5 };
FIS_TYPE* fis_gMFO0Coeff[] = { fis_gMFO0Coeff1, fis_gMFO0Coeff2,
    fis_gMFO0Coeff3, fis_gMFO0Coeff4, fis_gMFO0Coeff5 };
FIS_TYPE** fis_gMFO0Coeff[] = { fis_gMFO0Coeff };
// Input membership function set
int fis_gMFI0[] = { 0, 0, 0, 0, 0 };
int fis_gMFI1[] = { 0, 0, 0, 0, 0 };
int* fis_gMFI[] = { fis_gMFI0, fis_gMFI1 };
// Output membership function set
int fis_gMFO0[] = { 1, 0, 0, 0, 0 };

```

```

int* fis_gMFO[] = { fis_gMFO0};
// Rule Weights
FIS_TYPE fis_gRWeight[] = { 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 };
// Rule Type
int fis_gRType[] = { 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 };
// Rule Inputs
int fis_gRI0[] = { 1, 1 };
int fis_gRI1[] = { 1, 4 };
int fis_gRI2[] = { 1, 3 };
int fis_gRI3[] = { 1, 2 };
int fis_gRI4[] = { 1, 5 };
int fis_gRI5[] = { 1, 3 };
int fis_gRI6[] = { 4, 1 };
int fis_gRI7[] = { 4, 4 };
int fis_gRI8[] = { 4, 2 };
int fis_gRI9[] = { 4, 5 };
int fis_gRI10[] = { 4, 3 };
int fis_gRI11[] = { 2, 1 };
int fis_gRI12[] = { 2, 4 };
int fis_gRI13[] = { 2, 2 };
int fis_gRI14[] = { 2, 5 };
int fis_gRI15[] = { 2, 3 };
int fis_gRI16[] = { 5, 1 };
int fis_gRI17[] = { 5, 4 };
int fis_gRI18[] = { 5, 2 };
int fis_gRI19[] = { 5, 5 };
int fis_gRI20[] = { 5, 3 };
int fis_gRI21[] = { 3, 1 };
int fis_gRI22[] = { 3, 4 };
int fis_gRI23[] = { 3, 2 };
int fis_gRI24[] = { 3, 5 };
int fis_gRI25[] = { 3, 3 };
int* fis_gRI[] = { fis_gRI0, fis_gRI1, fis_gRI2, fis_gRI3, fis_gRI4, fis_gRI5, fis_gRI6,
fis_gRI7, fis_gRI8, fis_gRI9, fis_gRI10, fis_gRI11, fis_gRI12, fis_gRI13, fis_gRI14,
fis_gRI15, fis_gRI16, fis_gRI17, fis_gRI18, fis_gRI19, fis_gRI20, fis_gRI21, fis_gRI22,
fis_gRI23, fis_gRI24, fis_gRI25 };
// Rule Outputs
int fis_gRO0[] = { 4 };
int fis_gRO1[] = { 3 };
int fis_gRO2[] = { 1 };
int fis_gRO3[] = { 1 };
int fis_gRO4[] = { 1 };
int fis_gRO5[] = { 1 };
int fis_gRO6[] = { 5 };
int fis_gRO7[] = { 4 };
int fis_gRO8[] = { 1 };
int fis_gRO9[] = { 1 };

```



```

int fis_gRO10[] = { 1 };
int fis_gRO11[] = { 5 };
int fis_gRO12[] = { 4 };
int fis_gRO13[] = { 1 };
int fis_gRO14[] = { 1 };
int fis_gRO15[] = { 1 };
int fis_gRO16[] = { 5 };
int fis_gRO17[] = { 4 };
int fis_gRO18[] = { 1 };
int fis_gRO19[] = { 1 };
int fis_gRO20[] = { 1 };
int fis_gRO21[] = { 5 };
int fis_gRO22[] = { 5 };
int fis_gRO23[] = { 2 };
int fis_gRO24[] = { 2 };
int fis_gRO25[] = { 1 };
int* fis_gRO[] = { fis_gRO0, fis_gRO1, fis_gRO2, fis_gRO3, fis_gRO4, fis_gRO5,
fis_gRO6, fis_gRO7, fis_gRO8, fis_gRO9, fis_gRO10, fis_gRO11, fis_gRO12, fis_gRO13,
fis_gRO14, fis_gRO15, fis_gRO16, fis_gRO17, fis_gRO18, fis_gRO19, fis_gRO20,
fis_gRO21, fis_gRO22, fis_gRO23, fis_gRO24, fis_gRO25 };
// Input range Min
FIS_TYPE fis_gIMin[] = { 10, 50 };
// Input range Max
FIS_TYPE fis_gIMax[] = { 30, 95 };
// Output range Min
FIS_TYPE fis_gOMin[] = { 0 };
// Output range Max
FIS_TYPE fis_gOMax[] = { 5 };
//*****
// Data dependent support functions for Fuzzy Inference System
//*****
FIS_TYPE fis_MF_out(FIS_TYPE** fuzzyRuleSet, FIS_TYPE x, int o)
{
    FIS_TYPE mfOut;
    int r;
    for (r = 0; r < fis_gcR; ++r)
    {
        int index = fis_gRO[r][o];
        if (index > 0)
        {
            index = index - 1;
            mfOut = (fis_gMF[fis_gMFO[o][index]])(x, fis_gMFOCoeff[o][index]);
        }
        else if (index < 0)
        {
            index = -index - 1;
            mfOut = 1 - (fis_gMF[fis_gMFO[o][index]])(x, fis_gMFOCoeff[o][index]);
        }
    }
}

```

```

        else
        {
            mfOut = 0;
        }
        fuzzyRuleSet[0][r] = fis_min(mfOut, fuzzyRuleSet[1][r]);
    }
    return fis_array_operation(fuzzyRuleSet[0], fis_gcR, fis_max);
}
FIS_TYPE fis_defuzz_centroid(FIS_TYPE** fuzzyRuleSet, int o)
{
    FIS_TYPE step = (fis_gOMax[o] - fis_gOMin[o]) / (FIS_RESOLUTION - 1);
    FIS_TYPE area = 0;
    FIS_TYPE momentum = 0;
    FIS_TYPE dist, slice;
    int i;

    // calculate the area under the curve formed by the MF outputs
    for (i = 0; i < FIS_RESOLUTION; ++i){
        dist = fis_gOMin[o] + (step * i);
        slice = step * fis_MF_out(fuzzyRuleSet, dist, o);
        area += slice;
        momentum += slice*dist;
    }
    return ((area == 0) ? ((fis_gOMax[o] + fis_gOMin[o]) / 2) : (momentum / area));
}

/**
// Fuzzy Inference System
**/

void fis_evaluate()
{
    FIS_TYPE fuzzyInput0[] = { 0, 0, 0, 0, 0 };
    FIS_TYPE fuzzyInput1[] = { 0, 0, 0, 0, 0 };
    FIS_TYPE* fuzzyInput[fis_gcI] = { fuzzyInput0, fuzzyInput1, };
    FIS_TYPE fuzzyOutput0[] = { 0, 0, 0, 0, 0 };
    FIS_TYPE* fuzzyOutput[fis_gcO] = { fuzzyOutput0, };
    FIS_TYPE fuzzyRules[fis_gcR] = { 0 };
    FIS_TYPE fuzzyFires[fis_gcR] = { 0 };
    FIS_TYPE* fuzzyRuleSet[] = { fuzzyRules, fuzzyFires };
    FIS_TYPE sW = 0;

```

```

// Transforming input to fuzzy Input
int i, j, r, o;
for (i = 0; i < fis_gcI; ++i)
{
    for (j = 0; j < fis_gIMFCount[i]; ++j)
    {
        fuzzyInput[i][j] =
            (fis_gMF[fis_gMFI[i][j]])(g_fisInput[i], fis_gMFICoeff[i][j]);
    }
}
int index = 0;
for (r = 0; r < fis_gcR; ++r)
{
    if (fis_gRType[r] == 1)
    {
        fuzzyFires[r] = FIS_MAX;
        for (i = 0; i < fis_gcI; ++i)
        {
            index = fis_gRI[r][i];
            if (index > 0)
                fuzzyFires[r] = fis_min(fuzzyFires[r], fuzzyInput[i][index - 1]);
            else if (index < 0)
                fuzzyFires[r] = fis_min(fuzzyFires[r], 1 - fuzzyInput[i][-index - 1]);
            else
                fuzzyFires[r] = fis_min(fuzzyFires[r], 1);
        }
    }
}

else
{
    fuzzyFires[r] = FIS_MIN;
}

```

```

for (i = 0; i < fis_gcI; ++i)
{
    index = fis_gRI[r][i];
    if (index > 0)
        fuzzyFires[r] = fis_max(fuzzyFires[r], fuzzyInput[i][index - 1]);
    else if (index < 0)
        fuzzyFires[r] = fis_max(fuzzyFires[r], 1 - fuzzyInput[i][-index - 1]);
    else
        fuzzyFires[r] = fis_max(fuzzyFires[r], 0);
}
}
fuzzyFires[r] = fis_gRWeight[r] * fuzzyFires[r];
sW += fuzzyFires[r];
}
if (sW == 0)
{
    for (o = 0; o < fis_gcO; ++o)
    {
        g_fisOutput[o] = ((fis_gOMax[o] + fis_gOMin[o]) / 2);
    }
}
else
{
    for (o = 0; o < fis_gcO; ++o)
    {
        g_fisOutput[o] = fis_defuzz_centroid(fuzzyRuleSet, o);
    }
}
}
}
}

```