มหาวิทยาลัยแม่ฟ้าหลวง
MAE FAH LUANG UNIVERSITY

1504201 Computer Based Control

Project Report

BY

6631502023	ARKAR PYAE PHYO
6631502028	SWAN HTET
6631502055	AUNG MYINT MYAT

3 May 2025

Table of Contents

ACKNOWLEDGEMENTS

We would like to express our sincere gratitude to all those who supported us throughout the completion of this project on *Computer Based Control*.

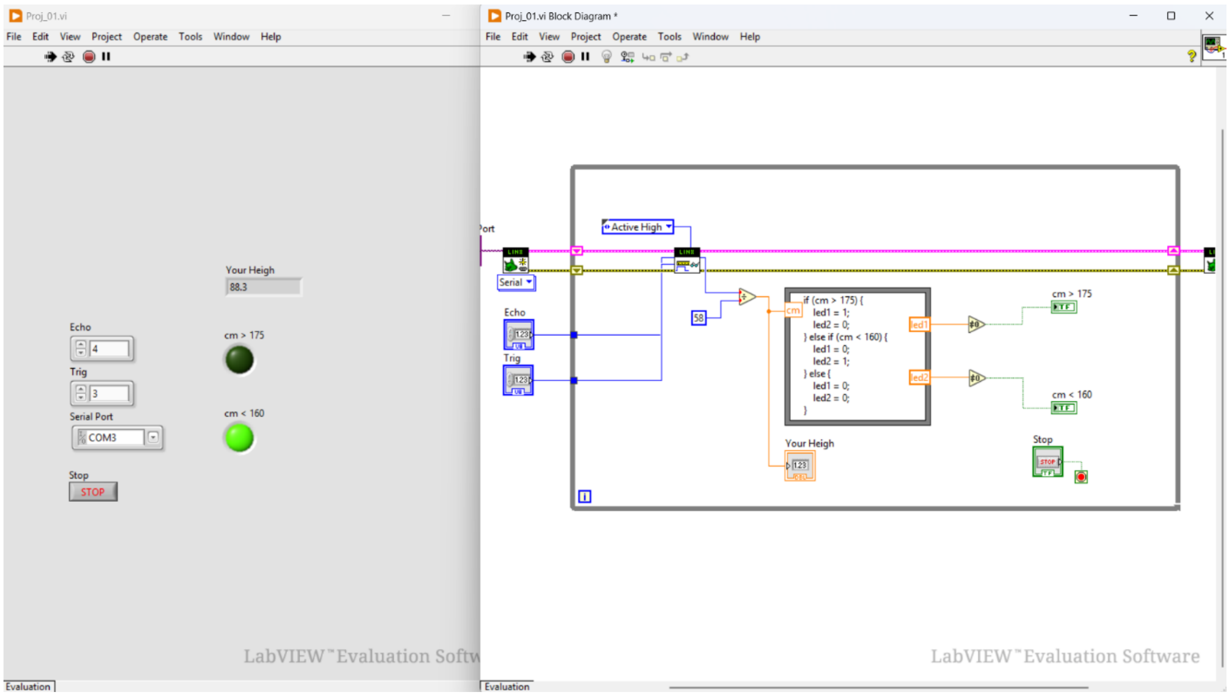
First, we would like to thank our professor, Teeravisit Laohapensaeng, for their invaluable guidance, constructive feedback, and continuous encouragement. Their expertise in the field enhanced our understanding of key concepts and helped shape the direction of this project.

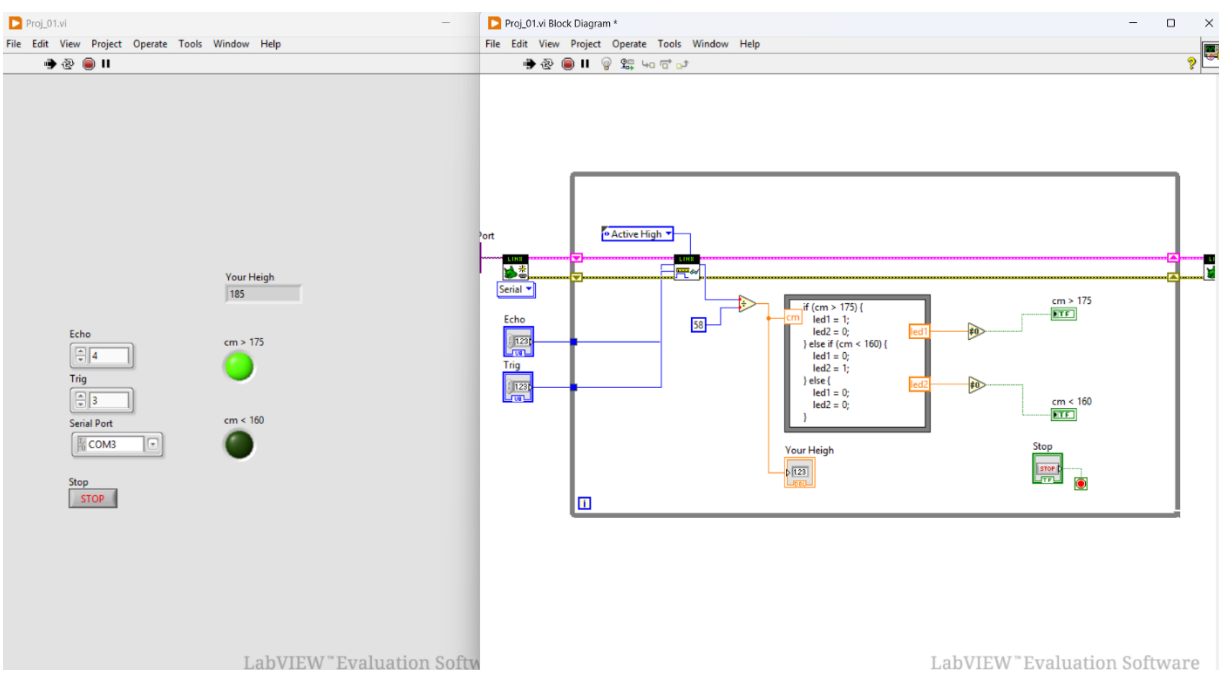
We are also grateful to our university, Mae Fah Luang University, for providing the necessary facilities and resources to conduct research and practical experiments. Special thanks to the Department of Applied Digital and Communication for offering a comprehensive curriculum that deepened our interest in networking systems.

We would also like to thank our classmates and friends for their helpful discussions and collaboration, which contributed to a better learning experience.

Thank you all.

1.HEIGH MEASUREMENT DEVICE USING ULTRASONIC SENSOR





This project presents a height measurement device using an ultrasonic sensor integrated with LabVIEW software. The ultrasonic sensor measures the distance from the sensor to the top of a person's head by sending and receiving ultrasonic pulses. The time taken for the echo to return is converted into centimeters, giving accurate reading.

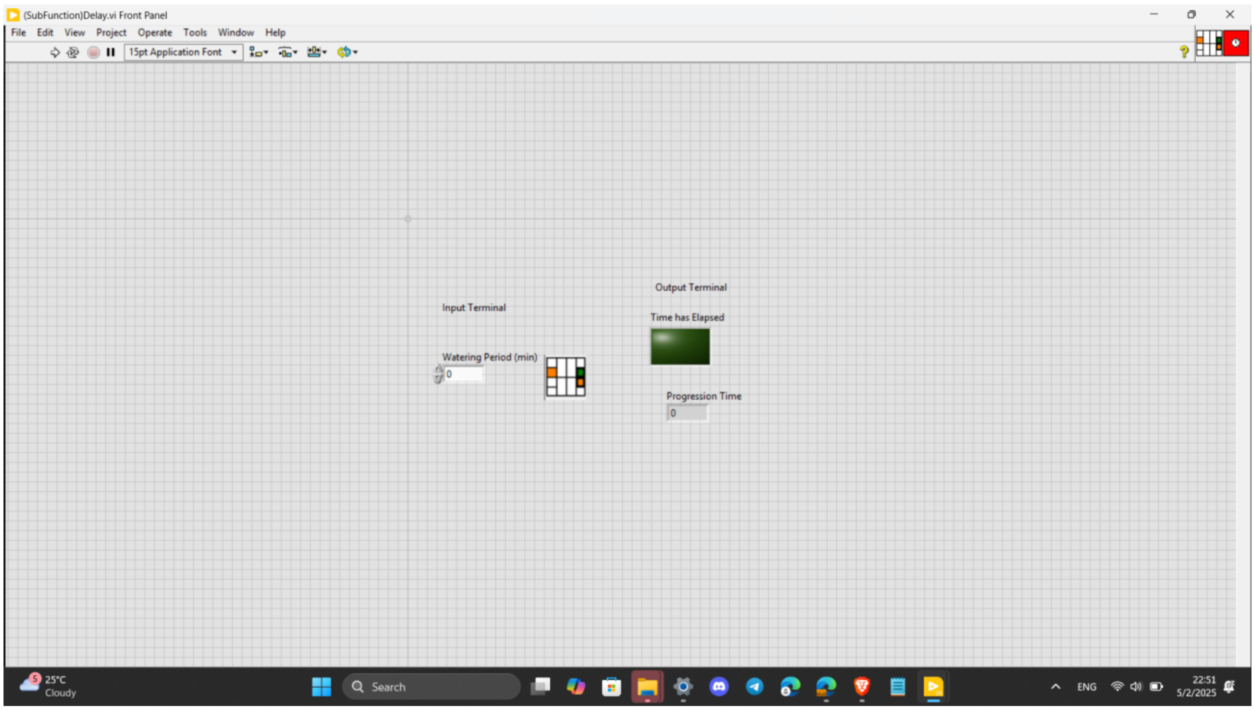
The LabVIEW interface allows users to set the Echo and Trigger pins and select the serial port for communication. The measured height is displayed in real-time, along with two LED indicators: one lights up if the height is greater than 175 cm, and another if the height is less than 160 cm. This provides a quick visual indication of height ranges.

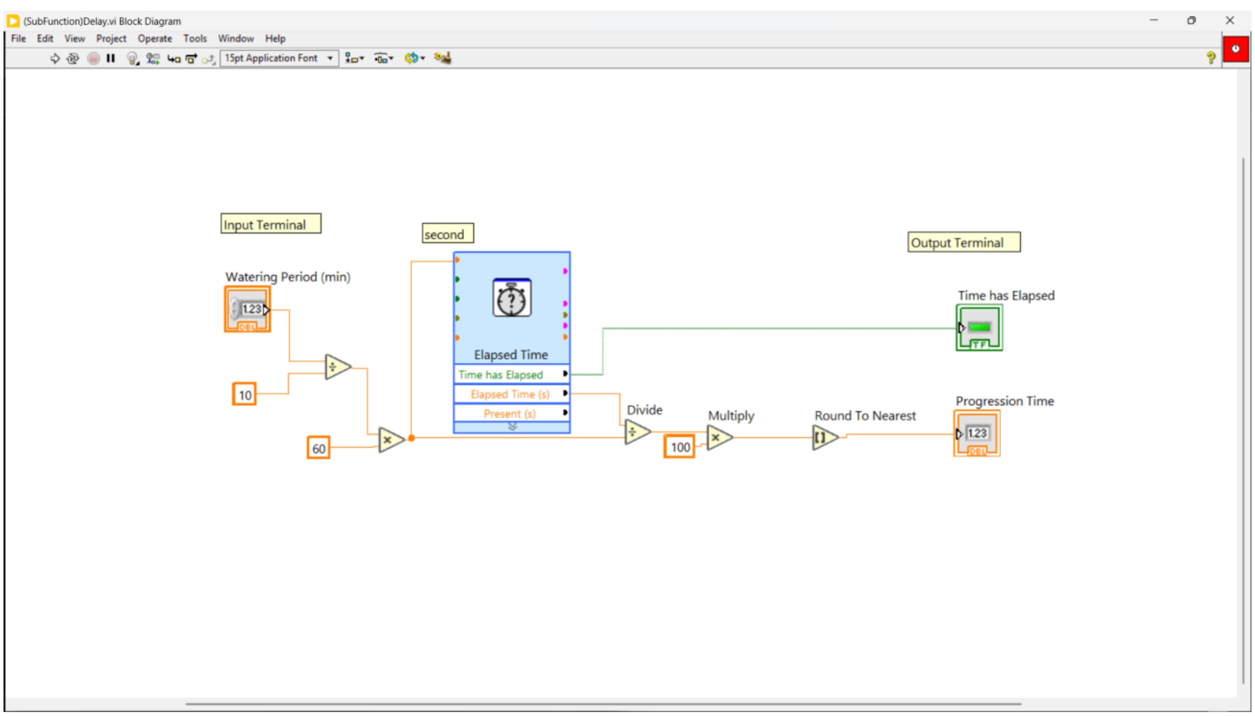
The block diagram includes serial communication for data reading, logical conditions to evaluate the height, and output controls for the LEDs. A stop button is provided for safe manual shutdown. The system is designed for continuous monitoring with user-friendly controls and clear outputs.

This device offers a simple and automated solution for height measurement. It can be used in health monitoring setups, entrance screening, or other scenarios where fast and reliable height detection is required.

2. SMART IRRIGATION SYSTEM USING AI MODEL

2.1 TIMER FUNCTION FOR WATERING CONTROL





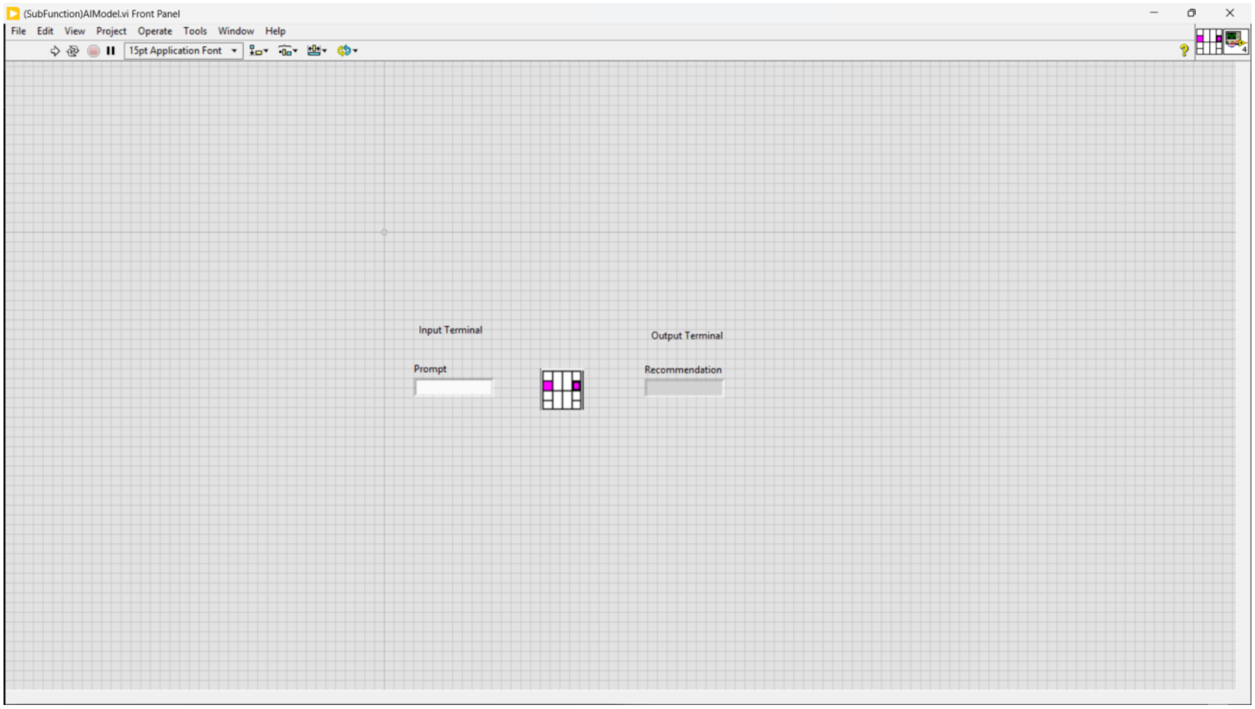
This section of the main system implements a timer function to manage and monitor the watering period within the overall irrigation control system. The user inputs the desired watering period in minutes, and the system tracks elapsed time in real-time to determine when watering should stop.

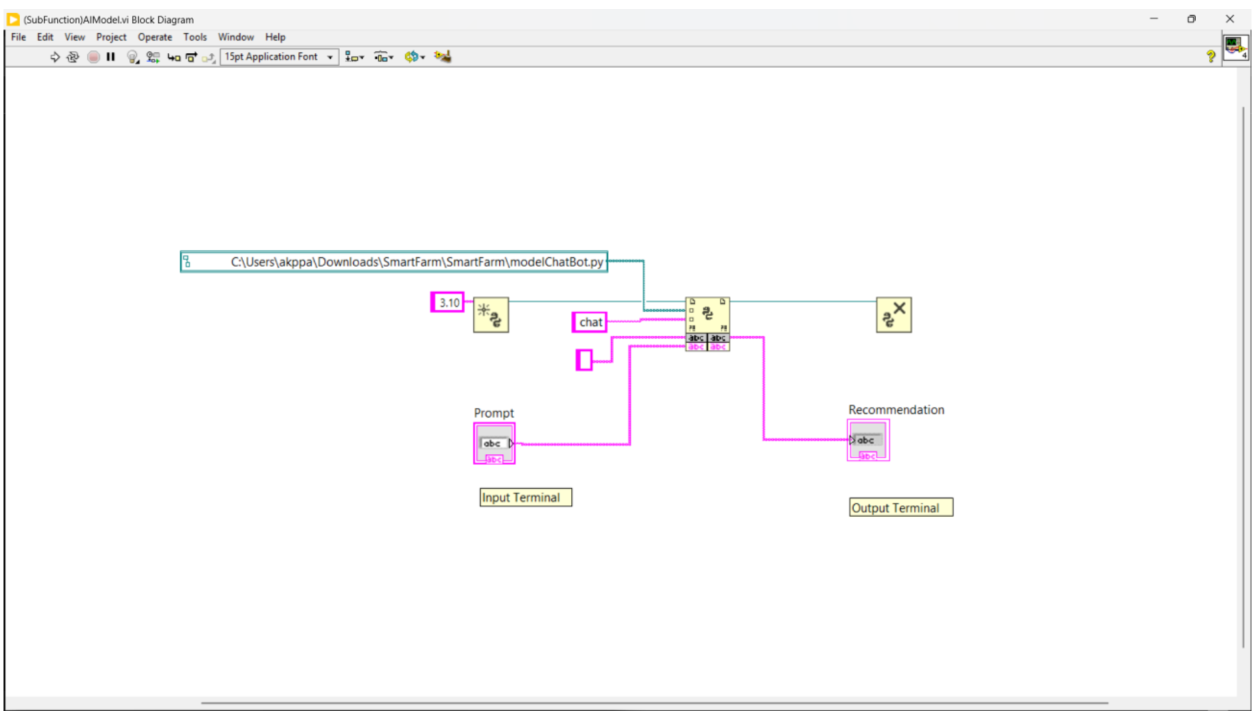
The front panel provides an input terminal for the watering period and two output indicators: one shows whether the full watering period has elapsed, and the other displays the progression time as a percentage. This allows the main system to clearly communicate the watering cycle's status to the user.

The block diagram utilizes LabVIEW's Elapsed Time function to monitor time accurately, converting the input period from minutes to seconds. The elapsed time is divided by the total period and multiplied by 100 to calculate the percentage of completion, which is rounded for clarity. This percentage is useful for visual feedback within the main system.

As a critical part of the main irrigation system, this timer function ensures precise control of watering cycles, contributing to efficient and automated plant care.

2.2 AI CHATBOT INTEGRATION MODULE



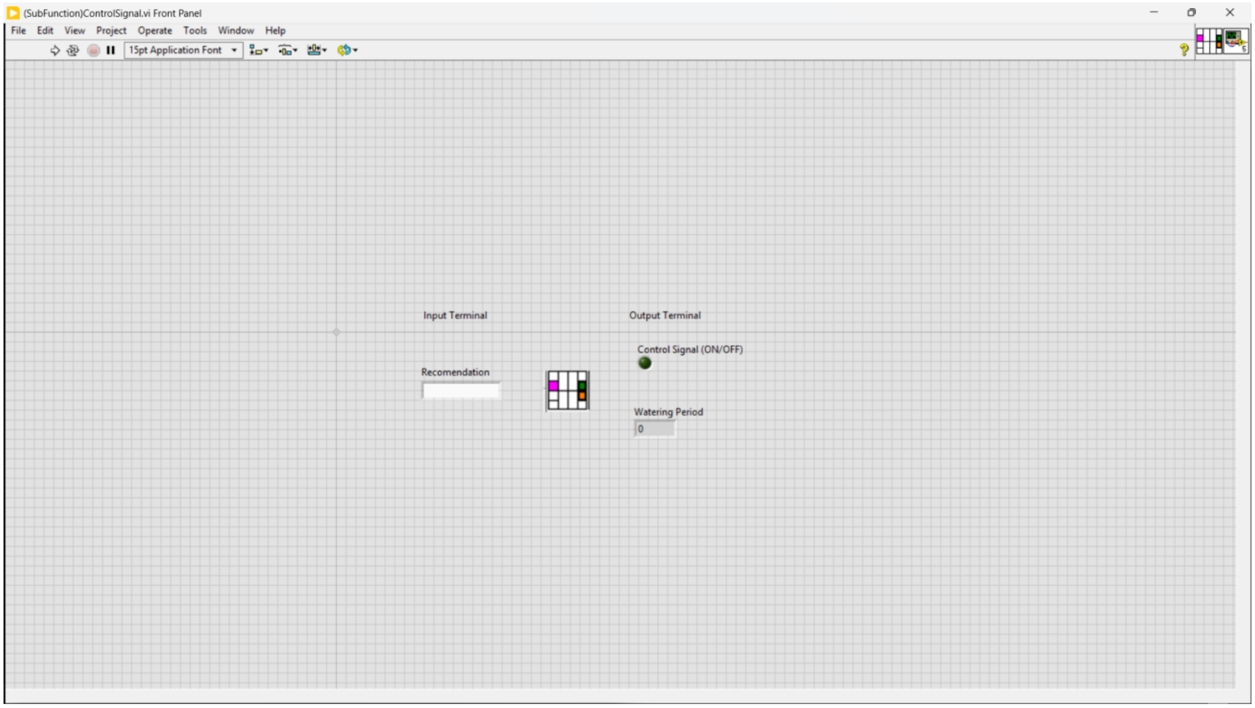


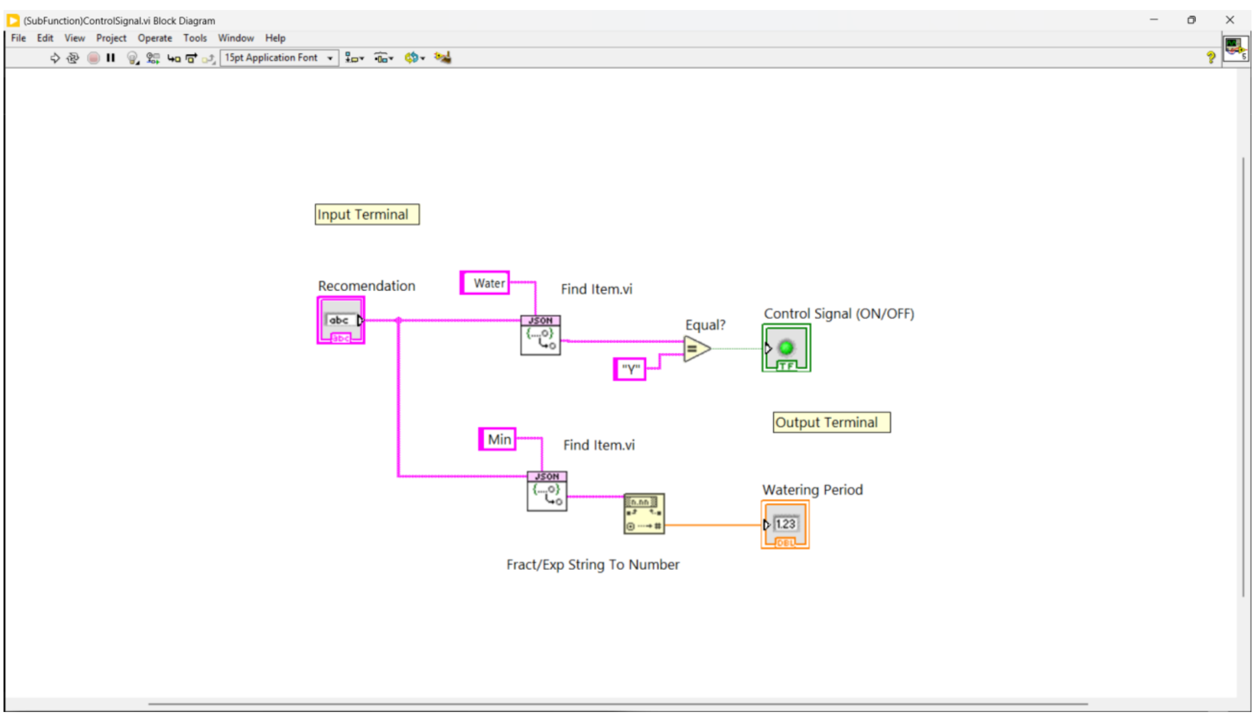
This module is part of the main system and integrates an AI-powered chatbot to provide real-time recommendations and responses based on user input. The system uses a Python script (ChatBot.py) executed via LabVIEW to handle AI processing, making the solution both flexible and scalable.

The block diagram shows the connection between the input terminal, where the user provides a text prompt, and the output terminal, which displays the AI-generated recommendation. The module initializes the Python environment (version 3.10), runs the chatbot script, and processes the response seamlessly within the LabVIEW interface.

This chatbot module enhances the main system by offering smart advice or guidance, such as farming tips, system status explanations, or troubleshooting help, making the system more interactive and user-friendly.

2.3 WATERING CONTROL SUBSYSTEM REPORT





This module is part of an automated irrigation control system, designed to manage plant watering recommendations based on predefined conditions. It interprets textual commands and activates a control signal to initiate watering when specific criteria are met.

The block diagram illustrates the logical flow from the Input Terminal, where a recommendation string (e.g., "Water for 10 minutes") is received, to the Output Terminal, which triggers the system's control signal (ON/OFF). The system uses LabVIEW's string processing functions to locate and extract key information such as the presence of a "Water" command and the associated watering duration (in minutes).

Two instances of the Find Item.vi functions are employed to locate:

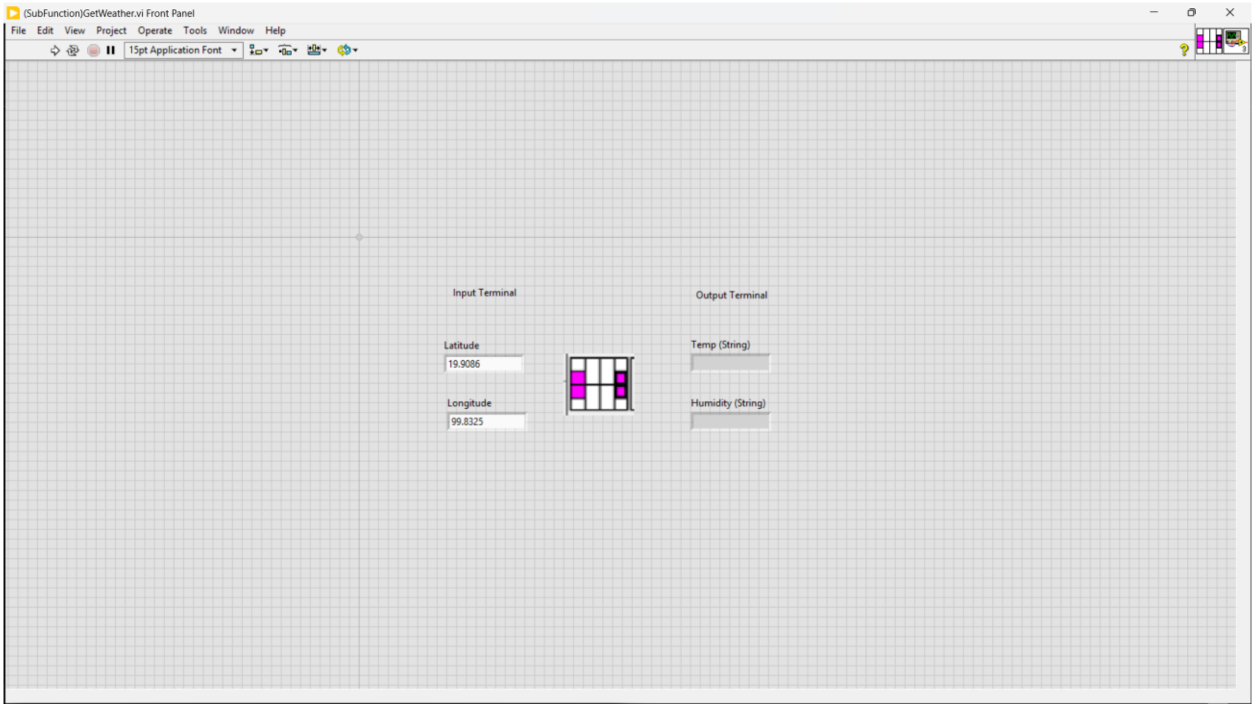
The keyword "Water" — confirming whether watering is needed.

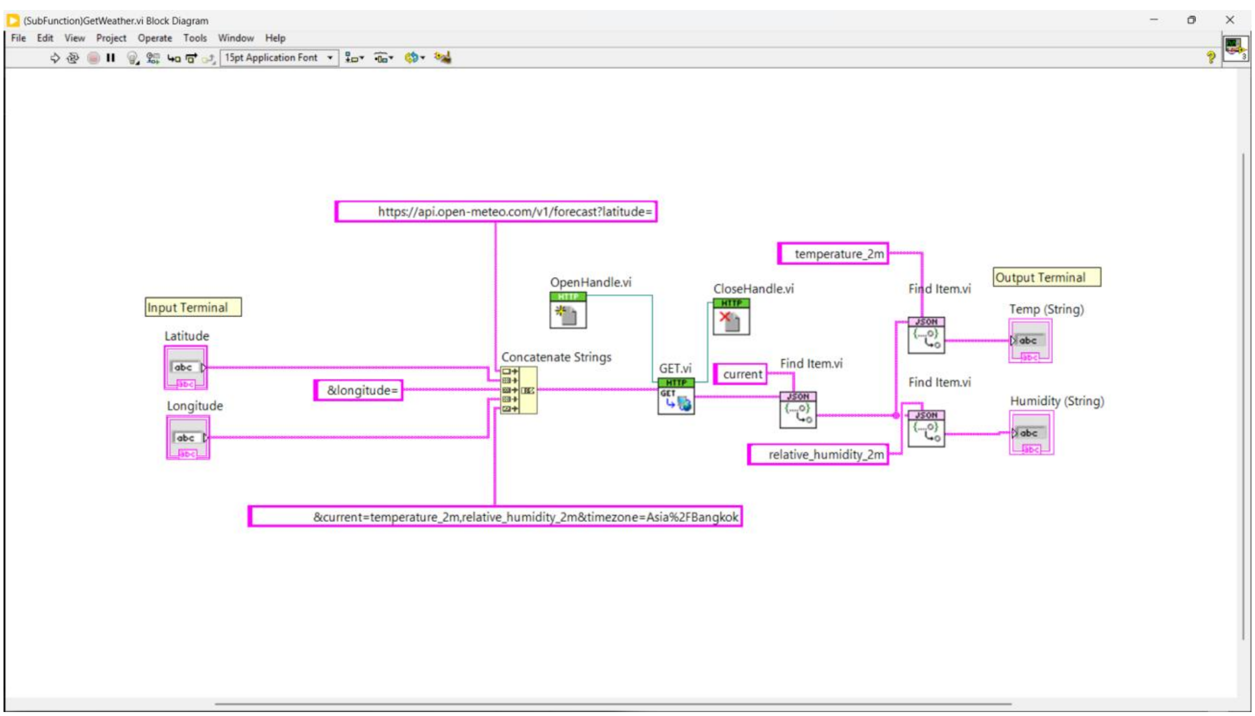
The keyword "Min" — used to identify the time duration value.

A comparison block checks if the word "Water" is present in the recommendation. If true, it activates the Control Signal, enabling the watering mechanism. Simultaneously, the text associated with the time duration is converted using the Fract/Exp String to Number function and passed to the Watering Period indicator, representing the number of minutes for which the watering should be executed.

This watering control module contributes to the larger system by enabling smart, conditional actuation based on textual input. It supports automation and intelligent control, allowing integration with higher-level decision-making systems such as AI-based recommendation engines.

2.4 WEATHER DATA ACQUISITION MODULE

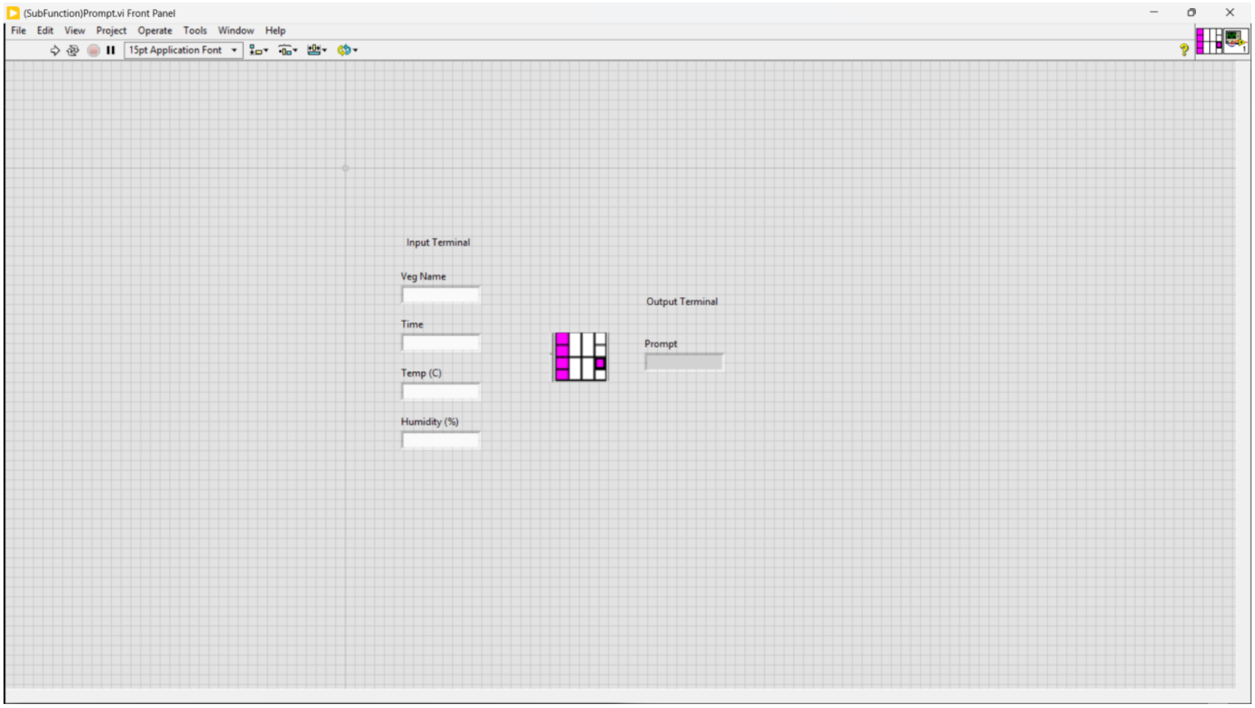


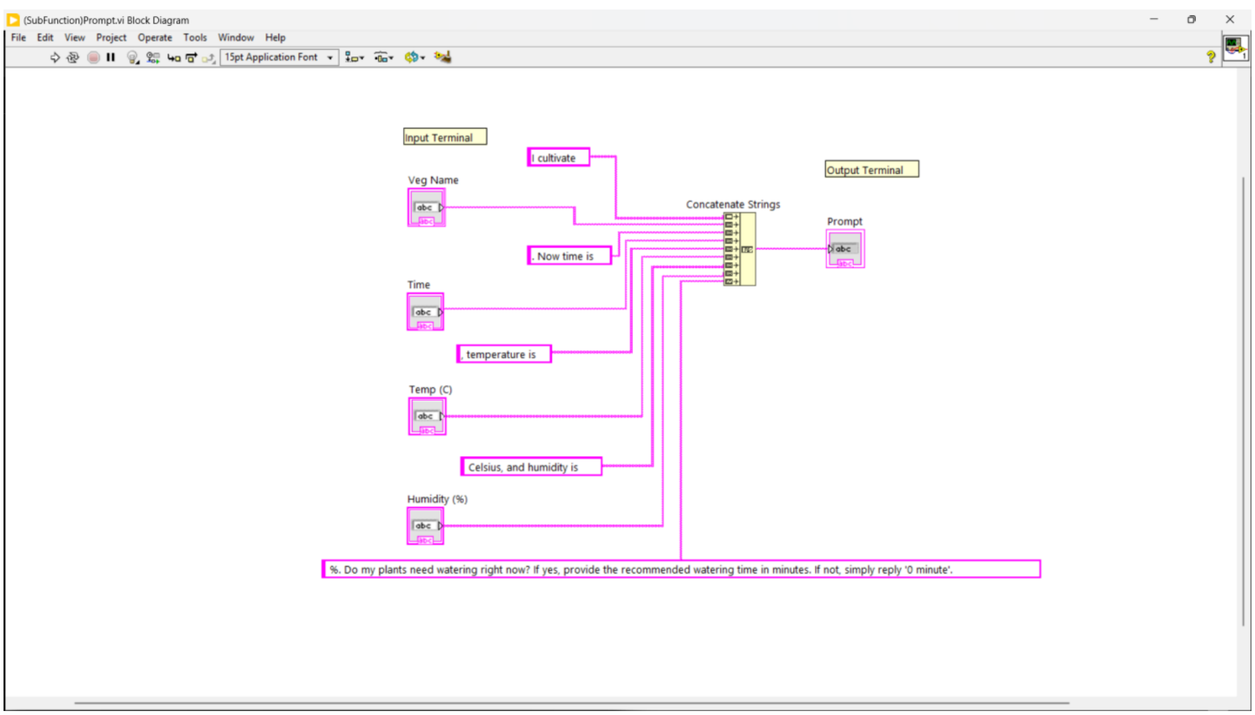


This LabVIEW module is designed to fetch real-time weather data—specifically temperature and humidity—using the Open-Meteo API. It takes latitude and longitude as inputs from the user, constructs a URL by concatenating the necessary parameters, and performs an HTTP GET request using built-in VI functions (OpenHandle.vi, GET.vi, and CloseHandle.vi) to retrieve live weather information.

The response is parsed using Find Item.vi to extract "temperature_2m" and "relative_humidity_2m" values, which are then sent to the output terminals as string data. This enables the larger system to make location-specific, weather-aware decisions, such as optimizing irrigation or environmental controls based on current atmospheric conditions.

2.5 PROMPT GENERATION MODULE

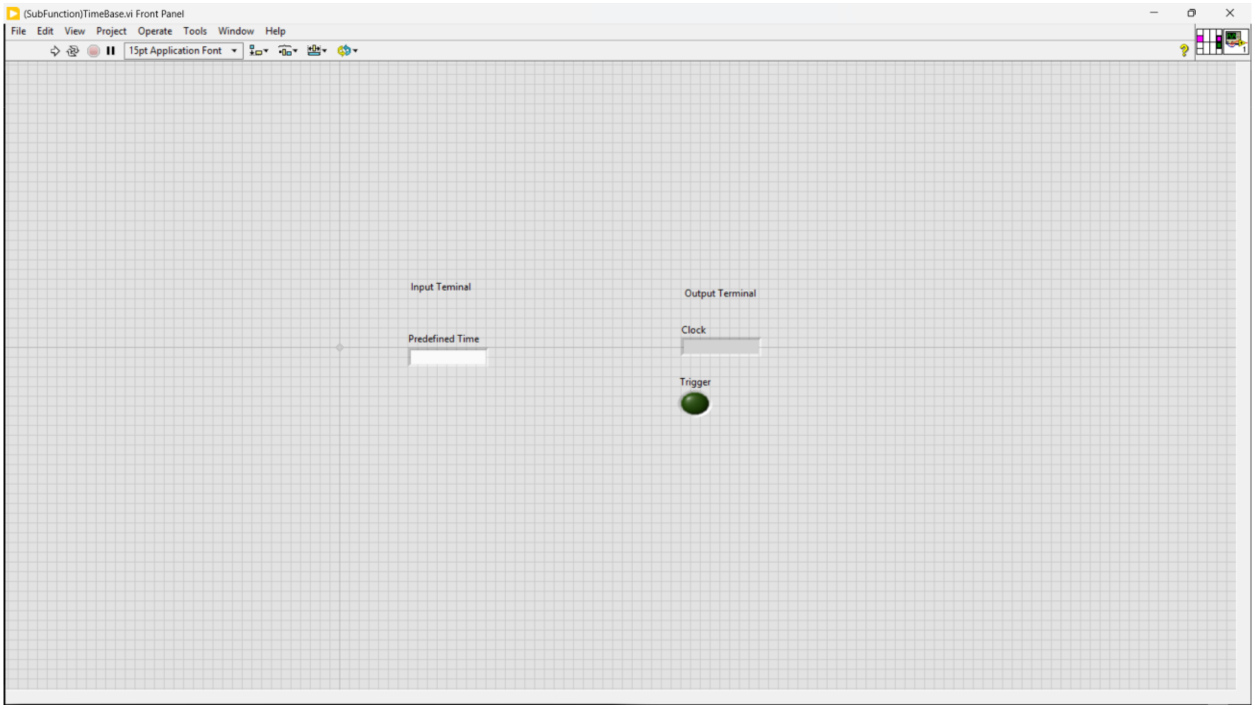


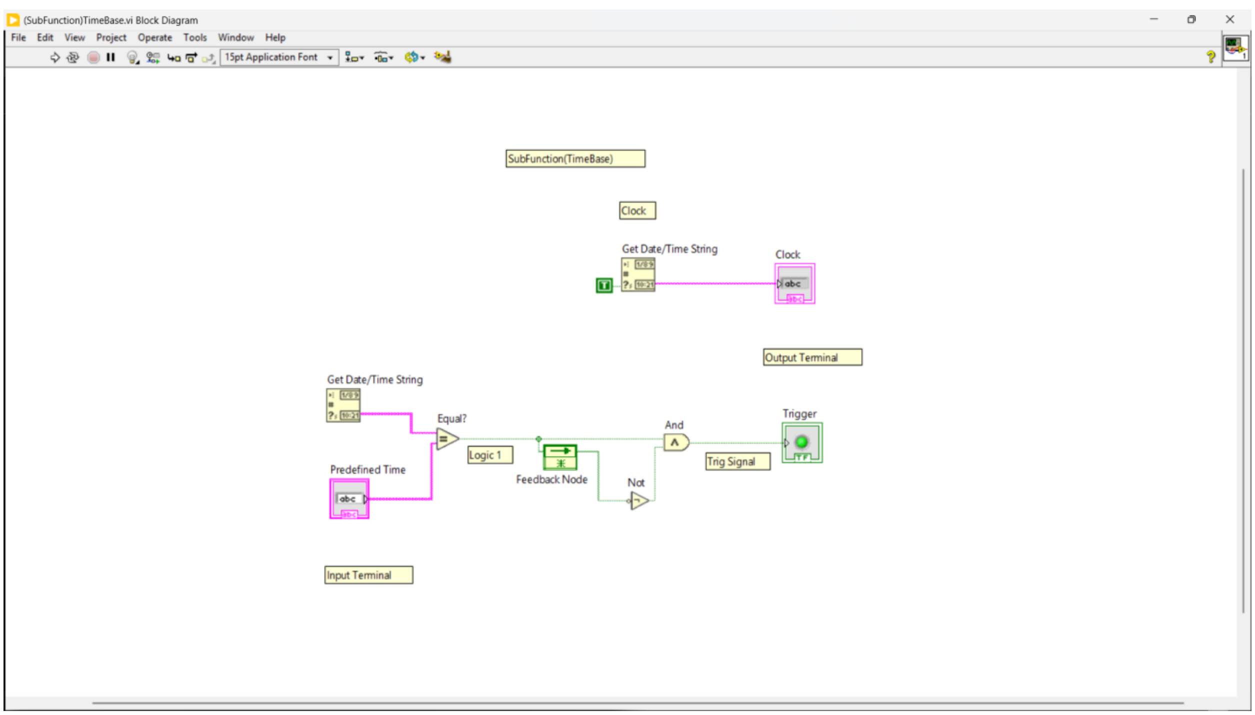


This LabVIEW module generates a dynamic text prompt based on current environmental and crop-specific inputs to assist in automated decision-making, typically by an AI or chatbot system. It takes inputs such as the vegetable name, current time, temperature, and humidity, and uses the Concatenate Strings function to assemble them into a complete, context-rich sentence.

The generated string forms a natural-language query asking whether the plant needs watering, including all relevant conditions like time, temperature in Celsius, and humidity percentage. The final prompt is output as a single string, which can be passed to an AI model to determine the appropriate watering recommendation. This enhances system interactivity and supports precision farming decisions.

2.6 TIME-BASED TRIGGER MODULE

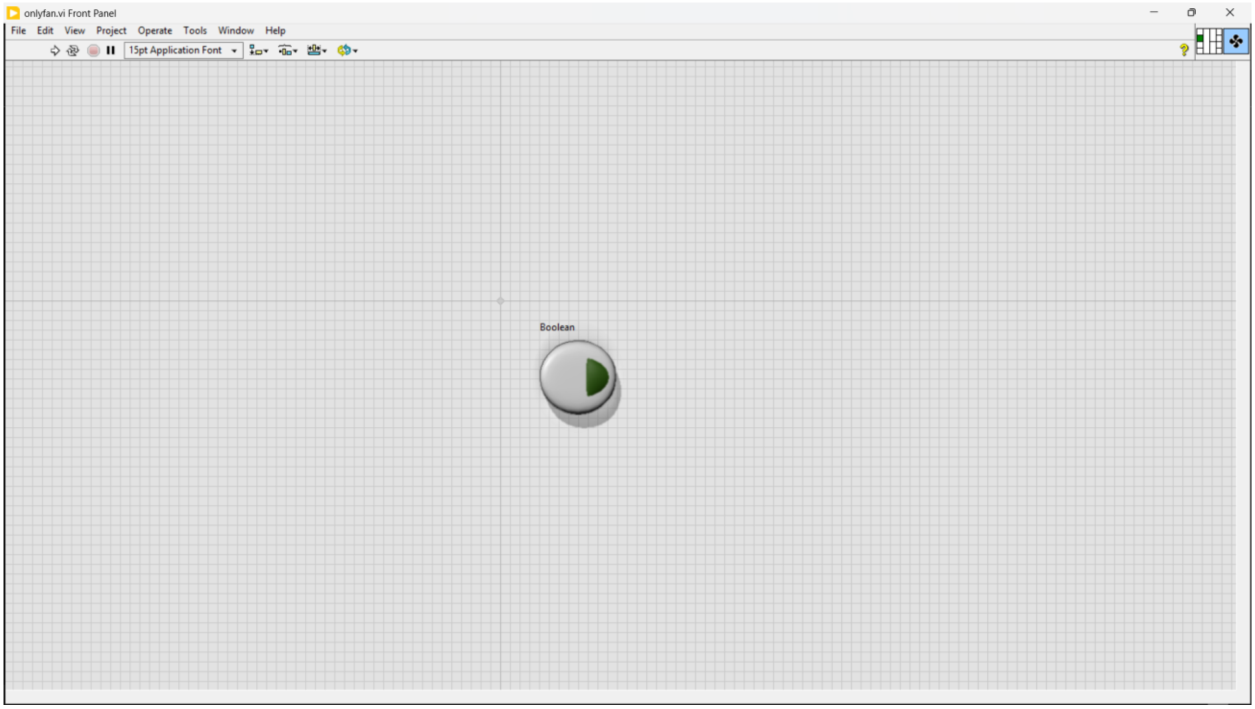


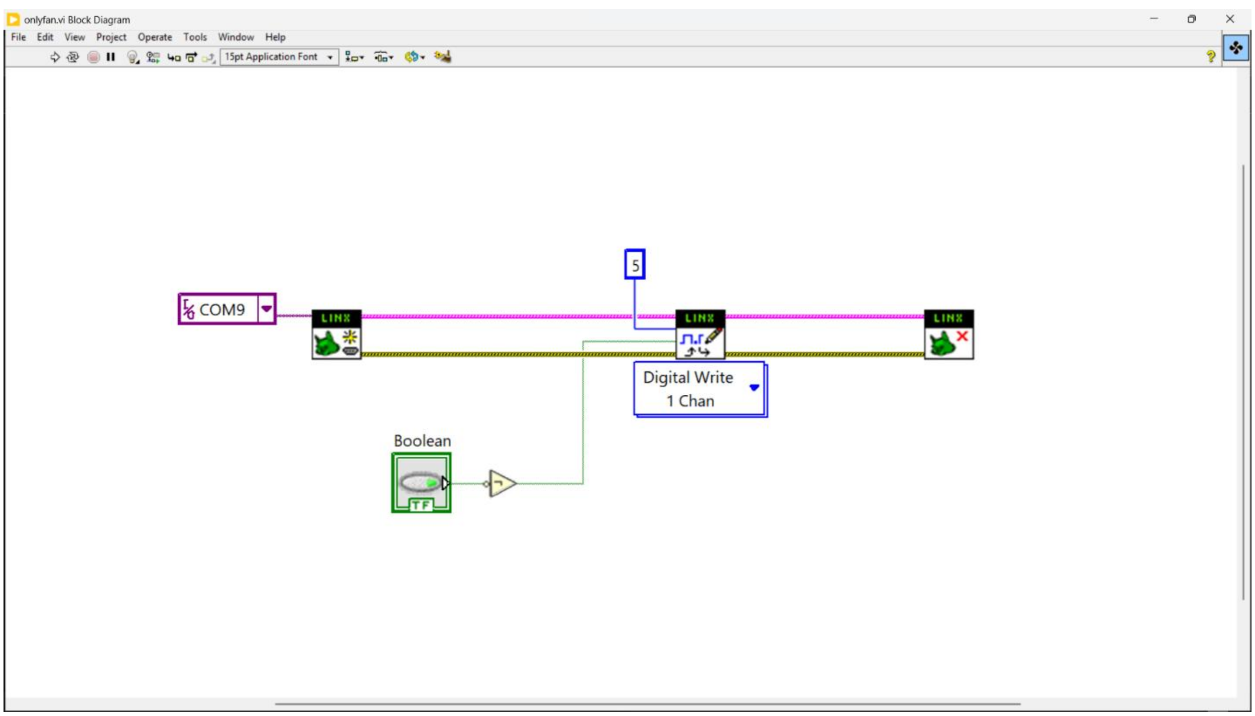


This module implements a time-based trigger mechanism that activates specific actions at a predefined time. It continuously reads the system's current time using the Get Date/Time String function and compares it with a user-defined time input. If the current time matches the predefined value, a logic block evaluates the condition, and a trigger signal is generated accordingly.

A feedback node and logical operations (AND, NOT) ensure the trigger is activated only once per match to prevent repeated triggering. The output is sent through a digital indicator labeled Trigger, signaling other parts of the system to proceed. This module is essential for scheduling tasks such as automated irrigation or data logging at specific intervals.

2.7 FAN CONTROL USING LINX AND DIGITAL WRITE

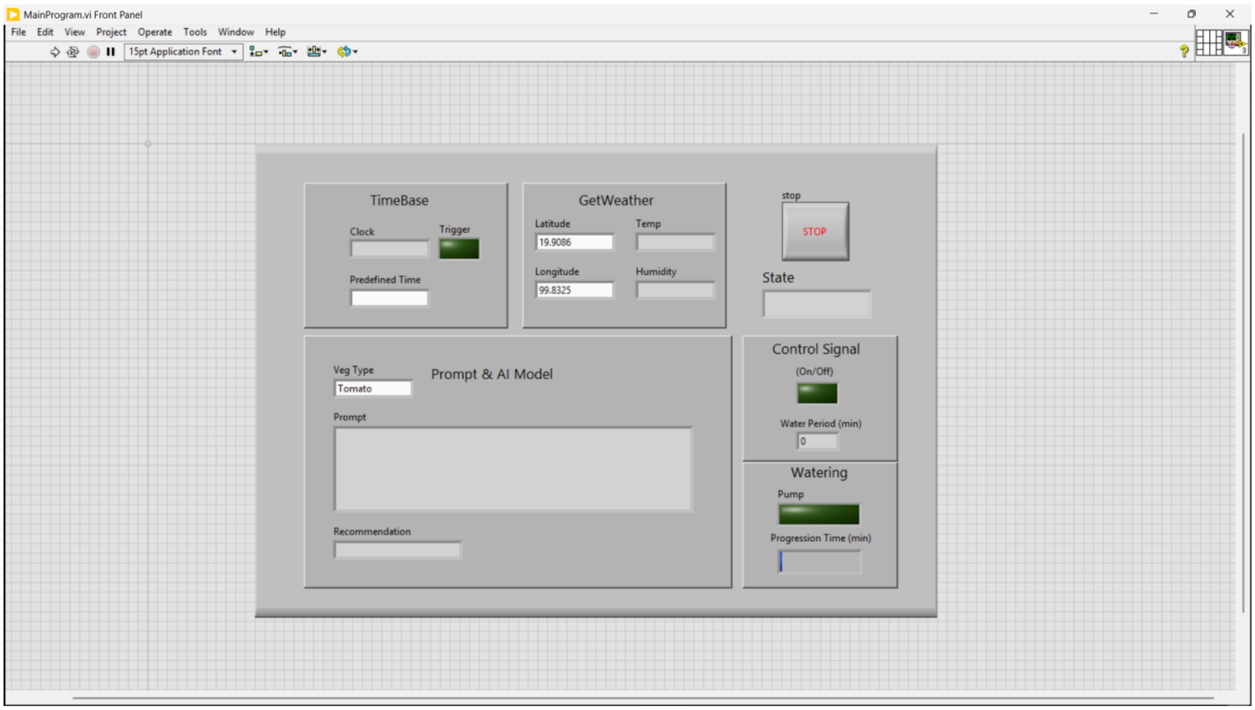


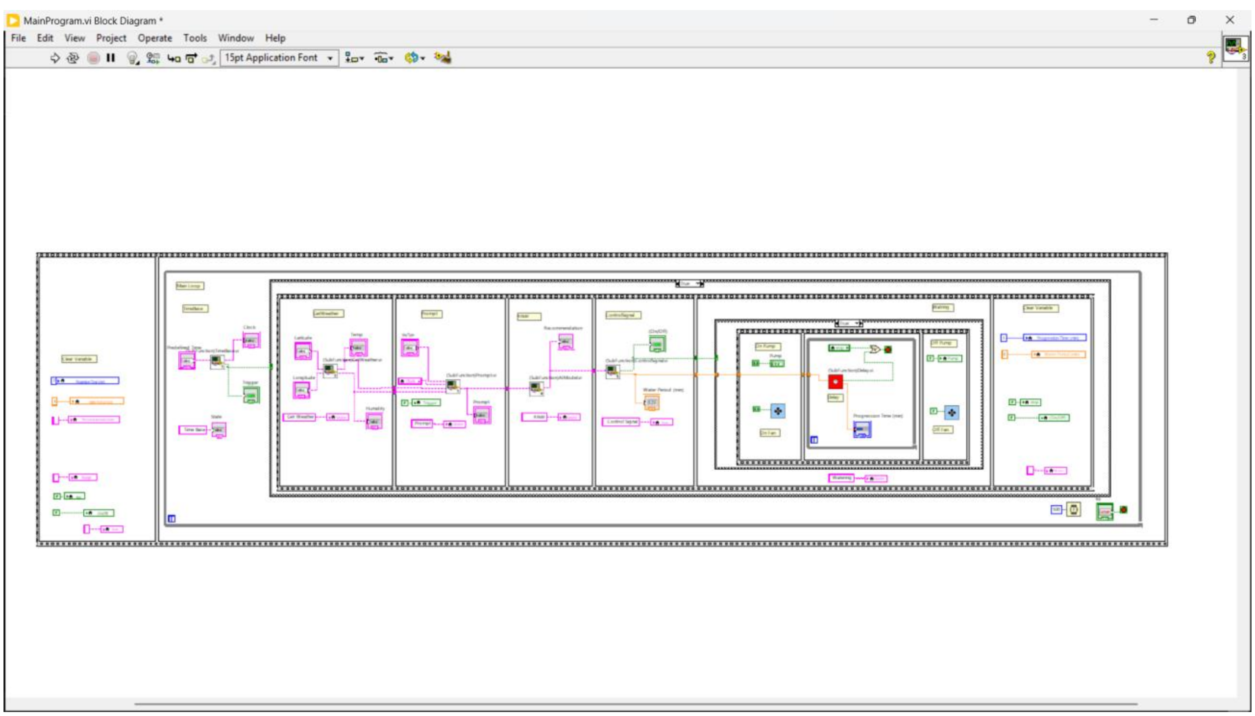


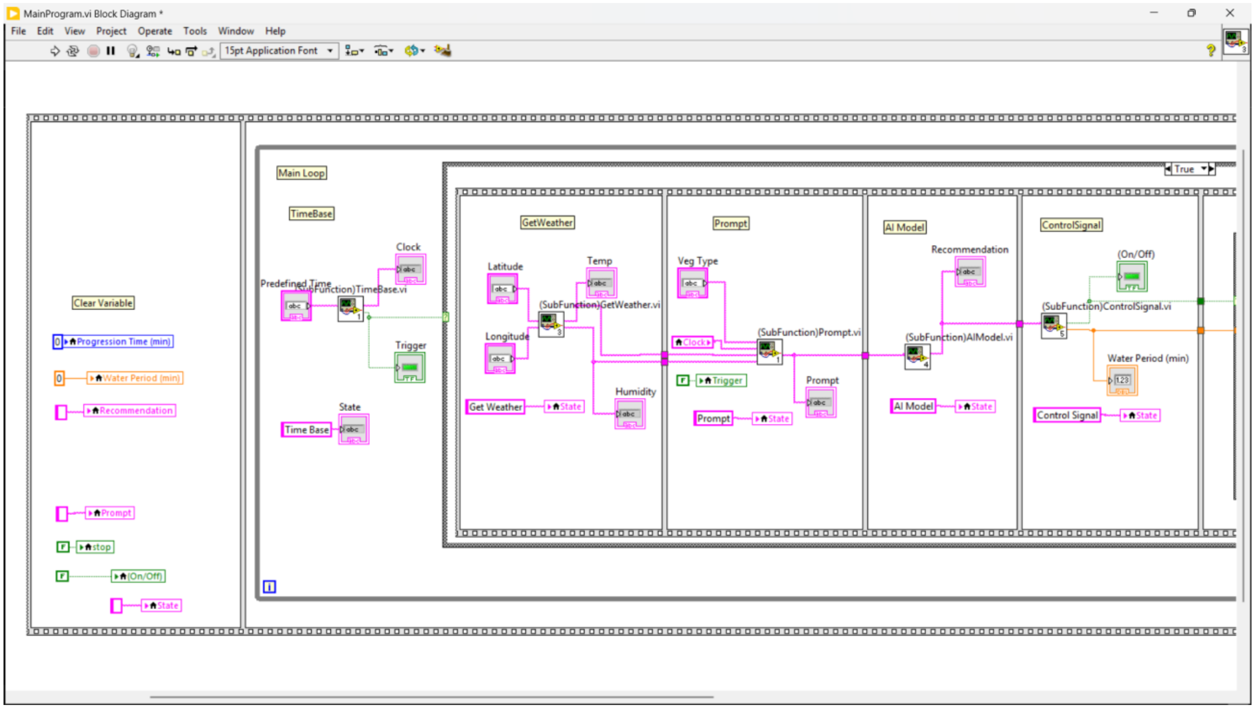
This VI establishes communication with an external microcontroller via the COM9 serial port using the LINX toolkit. It uses a digital output channel (pin 5) to control a fan. When the Boolean switch is toggled to True, the logic signal is inverted and sent to the Digital Write block, activating the fan by setting the pin LOW (assuming active-low logic).

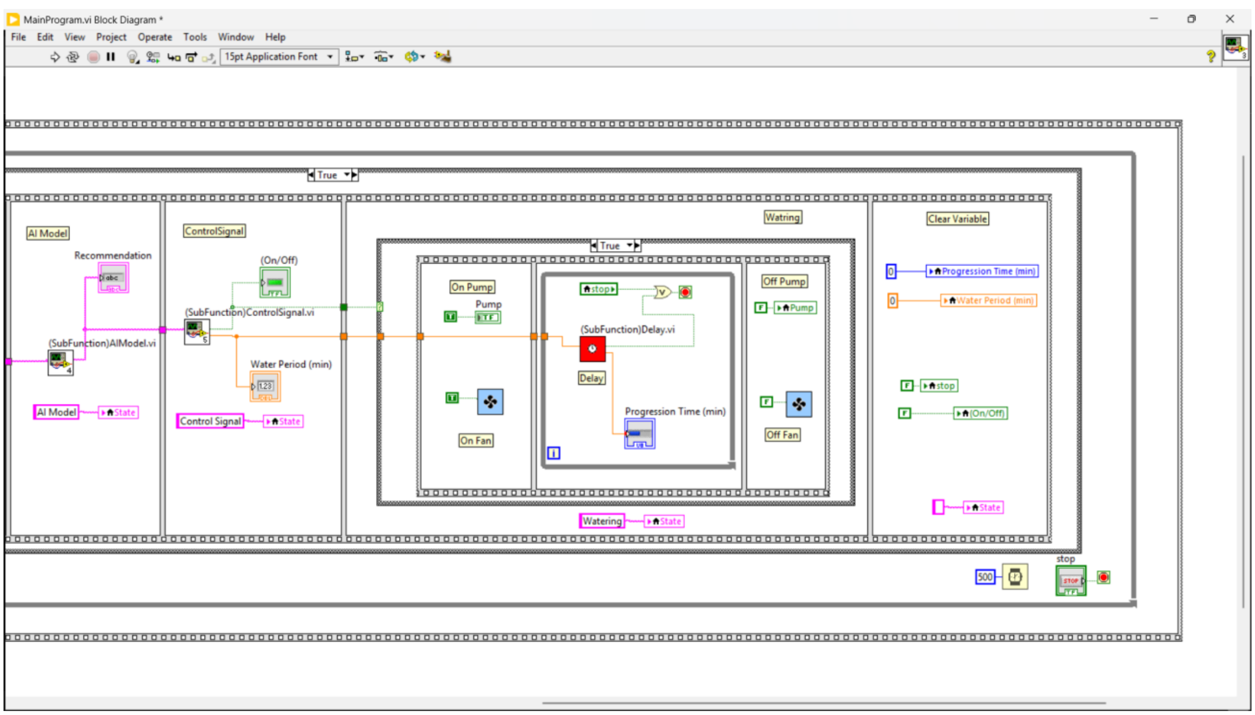
The VI structure also includes device initialization and closure to properly manage hardware resources. This setup is typically used for basic ON/OFF control of actuators like fans, relays, or LEDs, often integrated into automation systems such as smart irrigation or environmental control.

2.8 MAIN SMART IRRIGATION PROGRAM OVERVIEW






MainProgram.vi Block Diagram *
File Edit View Project Operate Tools Window Help
15pt Application Font
Clear Variable
Progression Time (min)
Water Period (min)
Recommendation
Prompt
stop
(On/Off)
State
Main Loop
TimeBase
Predefined Time
(SubFunction)TimeBase.vi
Clock
Trigger
State
Time Base
True
GetWeather
Latitude
Longitude
Temp
(SubFunction)GetWeather.vi
Humidity
Get Weather
State
Prompt
Veg Type
Clock
Trigger
(SubFunction)Prompt.vi
Prompt
Prompt
State
AI Model
Recommendation
(SubFunction)AIModel.vi
AI Model
State
ControlSignal
(On/Off)
(SubFunction)ControlSignal.vi
Water Period (min)
Control Signal
State


MainProgram.vi Block Diagram *
File Edit View Project Operate Tools Window Help
15pt Application Font
True
AI Model
Recommendation
(SubFunction)AIModel.vi
AI Model
State
ControlSignal
(On/Off)
(SubFunction)ControlSignal.vi
Water Period (min)
Control Signal
State
Watring
True
On Pump
Pump
On Fan
stop
(SubFunction)Delay.vi
Delay
Progression Time (min)
Off Pump
Pump
Off Fan
Watering
State
Clear Variable
Progression Time (min)
Water Period (min)
stop
(On/Off)
State
500
stop

MainProgram VI – Functional Steps

1. Set Preferred Time
   - The user inputs a predefined watering time in the “TimeBase” section.
   - The system continuously checks the current time.
2. Trigger Activation
   - When the system clock matches the predefined time:
     - The Trigger signal is activated.
     - A green LED lights up to indicate the trigger event.
3. Weather & AI Decision
   - The system fetches real-time weather data (Temperature and Humidity) using the provided Latitude and Longitude.
   - The user selects the Plant Type (e.g., Tomato).
   - An AI model generates a decision:
     - If watering is not needed: It replies, “0 minute.”
     - If watering is needed: It provides the recommended watering duration (e.g., “30 minutes”).
4. Watering & Fan Activation
   - If AI recommends watering:
     - The Pump is turned ON.
     - The Fan (via digital pin control) is also activated.
5. Auto Stop After Progress Time
   - A countdown begins based on the AI-recommended Progression Time.
   - Once the countdown completes:
     - The Pump and Fan turn OFF automatically.
     - The system resets for the next schedule.

CONCLUSIONS

Both projects demonstrate practical and intelligent use of LabVIEW for real-world automation. The smart irrigation system leverages AI, weather data, and sensor inputs to control watering and ventilation, optimizing plant care while conserving resources. The height measurement system offers a reliable and user-friendly method for quickly determining a person's height using ultrasonic sensing and visual indicators. Together, these projects showcase the versatility of LabVIEW in integrating hardware and logic to solve everyday problems efficiently and intelligently.