# Intelligent Vertical Farm Project Report

**Project Phase:** Final Integration & Documentation

**Controller:** Raspberry Pi 5

**Architecture:** Centralized Control Box (Breadboard-Free)

## 1. Introduction

This project aims to design and construct an automated, intelligent vertical farming system for the cultivation of *Capsicum annuum* (Chilli). The system utilizes the Nutrient Film Technique (NFT) / Drip Irrigation method, controlled by a Raspberry Pi 5. The core objective is to minimize human intervention by automating irrigation cycles, environmental climate control, and nutrient solution chemistry (pH balancing).

Unlike prototyping setups, this system rejects temporary breadboards in favor of a robust **Industrial Control Box** architecture, ensuring reliability and safety in a wet environment.

## 2. Mechanical Design & Structure

### 2.1 The Frame

* **Material:** 40x40mm Aluminum Extrusion Profiles (Slot 8).
* **Dimensions:** \* **Height:** 1543.5 mm (~1.5 meters).
  + **Width:** 680 mm (Outer) / 600 mm (Inner Growing Width).
  + **Depth:** 480 mm.
* **Mounting:** The frame utilizes angle brackets and T-nuts for modular assembly. Adjustable feet (Stellfuß D40) ensure stability on uneven floors.

### 2.2 The Planting Station (Hydroponics)

* **Channels:** Rigid PVC pipes (20mm supply lines / 100mm growing channels) mounted horizontally with a 1-2° slope for drainage.
* **Reservoir:** A light-proof black container positioned at the base (Sump).
* **Irrigation Plumbing:** \* **Pump:** 12V Brushless DC Submersible Pump (240L/H).
  + **Manifold:** 20mm PVC Main Line rising vertically, branching into 4mm Micro-Tubing.
  + **Delivery:** Pressure Compensating (PC) Drippers ensure equal water distribution to every plant regardless of height.

## 3. Electrical Architecture (The Control Box)

The system is centralized around a protective IP54-rated Control Box. This box houses the logic and power distribution, eliminating loose wires and breadboards.

### 3.1 Power Distribution

* **Primary Source:** 12V DC Power Supply Unit (PSU) - 5A+.
* **Voltage Regulation:** A Step-Down Buck Converter (12V to 5V) provides clean power to the Raspberry Pi and Sensors.
* **The Bus System:** Instead of a breadboard, the box uses **Wago Connectors / Terminal Blocks** to create internal power rails:
  + **12V Bus:** Powers Pumps, Fans, Lights.
  + **5V Bus:** Powers Logic, Sensors, Relay Coils.
  + **Common Ground Bus:** Essential for unifying 12V and 5V circuits.

### 3.2 The Actuator Interface (8-Channel Relay)

The Relay Module acts as the bridge between the Raspberry Pi's logic (3.3V) and the mechanical components (12V).

| **Relay Channel** | **GPIO Pin** | **Device Connected** | **Function** |
| --- | --- | --- | --- |
| **Relay 1** | GPIO 5 | **Main Water Pump** | Cyclic Irrigation (5 mins/hour). |
| **Relay 2** | GPIO 6 | **Grow Lights** | Day/Night Cycle (06:00-20:00). |
| **Relay 3** | GPIO 13 | **Air Fan** | Temperature Control (> 25°C). |
| **Relay 4** | GPIO 19 | **pH Down Pump** | Acid Dosing (High pH Correction). |
| **Relay 5** | GPIO 26 | **pH Up Pump** | Base Dosing (Low pH Correction). |
| **Relay 6-8** | Various | *(Expansion)* | Reserved for Nutrients A/B. |

### 3.3 The Sensor Network

Sensors are connected via specific protocols to the Control Box's external terminals.

* **I2C Bus (SDA/SCL):** Shared communication line for digital sensors.
  + **BME280:** Measures Air Temperature & Humidity.
  + **ADS1115 (ADC):** Converts analog signals from water sensors.
  + **LCD Display:** 16x2 Screen for status output.
* **Analog Inputs (via ADS1115):**
  + **Port A0:** pH Sensor Probe.
  + **Port A1:** EC (Electrical Conductivity) Sensor.
  + **Port A2:** Resistive Water Level Sensor (Safety Cutoff).

## 4. Software & Logic (Automation)

The system runs a Python script (main.py) as a systemd background service, launching automatically on boot.

### 4.1 The Control Loop

The software operates on a continuous feedback loop:

1. **READ:** Gather data from all sensors (Temp, pH, EC, Water Level).
2. **CHECK SAFETY:** \* If Water Level < Minimum: **EMERGENCY STOP** (Protect Pump).
   * If Temp > 30°C: **SAFETY CUTOFF** (Turn off Lights).
3. **DECIDE:**
   * *Is it daytime?* -> Lights ON.
   * *Is it hot?* -> Fan ON.
   * *Is it time to water?* -> Run Pump for 300 seconds.
   * *Is pH incorrect?* -> Pulse Dosing Pump for 1.5s -> **LOCKOUT** dosing for 15 mins to allow mixing.
4. **LOG & DISPLAY:** Save data to CSV for ML training and update LCD screen.

### 4.2 Machine Learning Integration

An AnomalyDetector class (using the Isolation Forest algorithm) runs parallel to the control logic. It analyzes the correlation between Temperature, Humidity, and pH. If the combination of values deviates from the "learned normal," it flags an alert, potentially identifying complex issues like sensor drift or root rot before they become visible.

## 5. Wiring Diagram (Schematic)

The following diagram illustrates the "Hub-and-Spoke" wiring architecture of the Control Box.

graph TD  
 %% --- INPUT SENSORS ---  
 subgraph Sensors ["SENSING LAYER"]  
 direction LR  
 BME["BME280 (Temp/Humid)"]  
 PH["pH Probe"]  
 EC["EC Probe"]  
 Level["Water Level Sensor"]  
 end  
  
 %% --- CONTROL BOX ---  
 subgraph Box ["CONTROL BOX (IP54)"]  
 direction TB  
 Pi["Raspberry Pi 5"]  
 Relay["8-Ch Relay Module"]  
 ADC["ADS1115 ADC"]  
 PSU12["12V PSU"]  
 PSU5["5V Buck Converter"]  
   
 %% Power Distribution Blocks  
 Terminal12V["12V Terminal Block"]  
 Terminal5V["5V Terminal Block"]  
 TerminalGND["GND Terminal Block"]  
   
 PSU12 ==> Terminal12V  
 Terminal12V --> PSU5  
 PSU5 --> Terminal5V  
   
 Terminal5V --> Pi  
 Terminal5V --> ADC  
 Terminal5V --> Relay  
   
 TerminalGND --> Pi & Relay & ADC & PSU12  
 end  
  
 %% --- ACTUATORS ---  
 subgraph Actuators ["ACTION LAYER"]  
 direction TB  
 PumpMain["Main Water Pump"]  
 Light["Grow Lights (25W)"]  
 Fan["Cooling Fan"]  
 PumpDown["pH Down Pump"]  
 PumpUp["pH Up Pump"]  
 end  
  
 %% --- CONNECTIONS ---  
   
 %% Sensor Signals  
 BME -- "I2C" --> Pi  
 PH -- "Analog A0" --> ADC  
 EC -- "Analog A1" --> ADC  
 Level -- "Analog A2" --> ADC  
 ADC -- "I2C" --> Pi  
   
 %% Relay Signals (GPIO)  
 Pi -- "GPIO 5-26" --> Relay  
   
 %% Power Output  
 Terminal12V == "12V Line" ==> PumpMain  
 Terminal12V == "12V Line" ==> Light  
 Terminal12V == "12V Line" ==> Fan  
 Terminal12V == "12V Line" ==> PumpDown  
 Terminal12V == "12V Line" ==> PumpUp  
   
 %% Relay Switching  
 Relay -.-> PumpMain  
 Relay -.-> Light  
 Relay -.-> Fan  
 Relay -.-> PumpDown  
 Relay -.-> PumpUp

## 6. Conclusion

This project successfully demonstrates the application of IoT and embedded systems in modern agriculture. By transitioning from a prototype breadboard layout to a centralized Control Box with industrial relay logic, the system achieves the reliability required for continuous, unmanned operation. The integration of sensors for pH, EC, and climate, combined with Machine Learning anomaly detection, creates a closed-loop system capable of optimizing plant growth conditions autonomously.