# APPENDICES

## Appendix 1 ESP32 Pseudocode

**Algorithm 1** Supervisory Main Execution Loop and Task Scheduler

**Require:** System timestamps $t_{read}, t_{auto}, t_{pub}$, mode flag $M_{mode} \in \{\text{AUTO}, \text{MANUAL}\}$, Actuator states $P_{irr}, P_{drain}, V_{drain}$.
**Ensure:** Periodic telemetry, safety interlocks, and state machine progression.

**loop**
  $t_{now} \leftarrow$ CURRENTTIMEMILLIS
  CHECKNETWORKCONNECTIVITY ▷ *Maintain Wi-Fi and MQTT message broker*
  ▷ ***Safety Interlock: Actuator Mutual Exclusion***
  **if** $P_{irr} = \text{ON}$ **and** $P_{drain} = \text{ON}$ **then**
    $P_{drain} \leftarrow \text{OFF}$ ▷ *Hardware safety override: force drain pump off*
  ▷ ***Task 1: Sensor Acquisition Routine*** $(\Delta t = 5\ s)$
  **if** $t_{now} - t_{read} \geq 5000\,\text{ms}$ **then**
    $t_{read} \leftarrow t_{now}$
    $(\mathbf{x}, Q) \leftarrow$ ACQUIREANDCONDITIONDATA
  ▷ ***Task 2: Automation Evaluation Routine*** $(\Delta t = 30\ s)$
  **if** $t_{now} - t_{auto} \geq 30000\,\text{ms}$ **then**
    $t_{auto} \leftarrow t_{now}$
    **if** $M_{mode} = \text{AUTO}$ **and** $\neg$AutomationActive **then**
      EVALUATEAUTOMATIONTRIGGERS($\mathbf{x}$)
  ▷ ***Task 3: Continuous Actuation State Execution***
  **if** $M_{mode} = \text{AUTO}$ **and** AutomationActive **then**
    EXECUTEACTUATIONFSM
  ▷ ***Task 4: Telemetry Dispatch*** $(\Delta t = 60\ s)$
  **if** $t_{now} - t_{pub} \geq 60000\,\text{ms}$ **then**
    $t_{pub} \leftarrow t_{now}$
    PUBLISHTELEMETRY($\mathbf{x}, \mathbf{L}, Q$, SystemState)

UNIVERSITAS PENDIDIKAN GANESHA
UNDIKSHA

**Algorithm 2** Data Acquisition and Feature Conditioning

**Require:** 16-bit ADC channels $ADC_1, ADC_2, ADC_t$, flow pulse count $C_{pulse}$, sample window $\Delta t_{flow}$, remote node timestamps $t_{seenA}, t_{seenB}$.
**Ensure:** Calibrated feature vector $\mathbf{x} = [TDS_1, TDS_2, Turb, \Delta TDS]$, level array $\mathbf{L} = [L_1, L_2, L_3, L_4]$, flow rate $Q$.

**procedure** AcquireAndConditionData
  ▷ *1. Temperature Compensation for Dual TDS Probes*
  $k_{temp} \leftarrow 1.0 + 0.02 \times (T_{site} - 25.0)$ ▷ *Normalized to 28° C baseline*
  $TDS_1 \leftarrow \max\left(0, \frac{a_1 \cdot ADC_1 + b_1}{k_{temp}}\right)$
  $TDS_2 \leftarrow \max\left(0, \frac{a_2 \cdot ADC_2 + b_2}{k_{temp}}\right)$
  ▷ *2. Symmetrical Relative TDS Deviation*
  $TDS_{avg} \leftarrow (TDS_1 + TDS_2)/2.0$
  **if** $TDS_{avg} > 0$ **then**
    $\Delta TDS \leftarrow \left(\frac{|TDS_1 - TDS_2|}{TDS_{avg}}\right) \times 100\%$
  **else**
    $\Delta TDS \leftarrow 0.0\%$
  ▷ *3. Turbidity Conditioning and Volumetric Flow Rate*
  $Turb \leftarrow \max(0, a_t \cdot ADC_t + b_t)$ ▷ *Transfer function to NTU*
  $Q \leftarrow \frac{C_{pulse}}{\Delta t_{flow} \times 7.5}$ ▷ *Discharge in L/min, 7.5 pulse/L constant*
  $C_{pulse} \leftarrow 0$ ▷ *Reset hardware ISR counter*
  ▷ *4. Wireless Water Level Watchdog*
  $t_{now} \leftarrow$ CurrentTimeMillis
  FreshData $\leftarrow (t_{now} - t_{seenA} < 30000\,\text{ms})$ **and** $(t_{now} - t_{seenB} < 30000\,\text{ms})$
  **return** $\mathbf{x} \leftarrow [TDS_1, TDS_2, Turb, \Delta TDS]$, $\mathbf{L}$, $Q$, FreshData

**Algorithm 3** Decision Tree Water Quality Inference

**Require:** Feature vector $\mathbf{x} = [TDS_1, TDS_2, Turb, \Delta TDS]$.
**Ensure:** Operational state classification $C \in \{0, 1, 2, 3\}$.

**function** PredictWaterQuality($\mathbf{x}$)
  **if** $Turb \leq 25.0\,\text{NTU}$ **then**
    **if** $\Delta TDS \leq 20.0\%$ **then**
      **return** 2 ▷ *Class 2: Normal Quality (Optimal baseline)*
    **else**
      **return** 3 ▷ *Class 3: Nutrient Imbalance ($\Delta TDS$ anomaly)*
  **else** ▷ *$Turb > 25.0$ NTU (High Turbidity Domain)*
    **if** $TDS_1 \leq 1400.0\,\text{ppm}$ **then**
      **return** 1 ▷ *Class 1: Cloudy due to Sediment (Requires remediation)*
    **else**
      **return** 0 ▷ *Class 0: Cloudy due to Nutrient Precipitate*

**Algorithm 4** Closed-Loop Autonomous Actuation State Machine

**Require:** Feature vector $\mathbf{x}$, tank levels $\mathbf{L}$, state timers $t_{start}, t_{settle}, t_{cond}$, retry counts $N_{drain}, N_{topup}$.
**Ensure:** Relay switching signals for Irrigation Pump $P_{irr}$ and Tank Drain Valve $V_{drain}$.

**procedure** EVALUATEAUTOMATIONTRIGGERS($\mathbf{x}$)
  **if** ¬FreshData **then**
    **return** ▷ *Abort trigger evaluation if level telemetry is stale*
  ▷ ***Priority 1: Level Depletion Protection***
  **if** $\exists L_i \in \mathbf{L} : L_i < 30\%$ **then**
    **if** $t_{now} \geq t_{cooldown}$ **then**
      CurrentAction ← TOPUP, AutomationActive ← **true**
      $t_{start} \leftarrow t_{now}$, $\mathbf{L}_{baseline} \leftarrow \mathbf{L}$
    **return**
  ▷ ***Priority 2: Water Quality Remediation***
  $C \leftarrow$ PREDICTWATERQUALITY($\mathbf{x}$)
  **if** $C = 1$ **then** ▷ *Sediment contamination detected*
    CurrentAction ← DRAINREFILL, AutomationActive ← **true**
    DrainSettling ← **false**, $t_{start} \leftarrow t_{now}$
  **else if** $C \in \{0, 3\}$ **then**
    SENDNOTIFICATIONALERT($C$) ▷ *Notification only; actuators idle*

**procedure** EXECUTEACTUATIONFSM CurrentAction DRAINREFILL
  **if** ¬DrainSettling **then** ▷ ***Phase 1: Tank Drainage***
    $V_{drain} \leftarrow$ ON, $P_{irr} \leftarrow$ OFF
    $cond \leftarrow$ CHECKSTABLE($\forall L_i \in \mathbf{L}, L_i \leq 20\%, \tau_{stable} = 3\,\text{s}$)
    **if** $cond$ **or** $(t_{now} - t_{start} \geq 60\,\text{s})$ **then**
      $V_{drain} \leftarrow$ OFF, DrainSettling ← **true**, $t_{settle} \leftarrow t_{now}$
  **else** ▷ ***Phase 2: Refill and Verification***
    **if** $t_{now} - t_{settle} < 500\,\text{ms}$ **then**
      **return** ▷ *Hydraulic settling delay*
    $P_{irr} \leftarrow$ ON
    $cond \leftarrow$ CHECKSTABLE($\forall L_i \in \mathbf{L}, L_i \geq 80\%, \tau_{stable} = 3\,\text{s}$)
    **if** $cond$ **or** $(t_{now} - t_{start} \geq 180\,\text{s})$ **then**
      $P_{irr} \leftarrow$ OFF
      $(\mathbf{x}', \sim) \leftarrow$ ACQUIREANDCONDITIONDATA
      $C' \leftarrow$ PREDICTWATERQUALITY($\mathbf{x}'$) ▷ *Self-verification recheck*
      **if** $C' = 1$ **and** $N_{drain} < 2$ **then**
        $N_{drain} \leftarrow N_{drain} + 1$, DrainSettling ← **false**, $t_{start} \leftarrow t_{now}$
      **else**
        AutomationActive ← **false**, $N_{drain} \leftarrow 0$
  TOPUP
  $P_{irr} \leftarrow$ ON
  $cond \leftarrow$ CHECKSTABLE($\forall L_i \in \mathbf{L}, L_i \geq 80\%, \tau_{stable} = 3\,\text{s}$)
  **if** $cond$ **then**
    $P_{irr} \leftarrow$ OFF, AutomationActive ← **false**, $N_{topup} \leftarrow 0$
  **else if** $t_{now} - t_{start} \geq 120\,\text{s}$ **then** ▷ *Safety timeout reached*
    $P_{irr} \leftarrow$ OFF, AutomationActive ← **false**
    **if** $\forall L_i \in \mathbf{L}, (L_i - L_{baseline,i} < 3\%)$ **then** ▷ *Stall / pump fault detected*
      $N_{topup} \leftarrow N_{topup} + 1$
      **if** $N_{topup} \geq 2$ **then**
        $t_{cooldown} \leftarrow t_{now} + 1800\,\text{s}$, $N_{topup} \leftarrow 0$ ▷ *Lockout for 30 minutes*

UNIVERSITAS PENDIDIKAN GANESHA
UNDIKSHA

## Appendix 2 WEMOS D1 Mini

**Algorithm 1** WEMOS A - Water Level 1-2

**function** SAMPLELEVELSENSOR($Pin_{pwr}, Ch_{adc}, ADC_{min}, ADC_{max}$)
  Set $Pin_{pwr} \leftarrow$ HIGH ▷ Energize sensor probe to prevent galvanic corrosion
  Wait 150 ms ▷ Transient stabilization / warmup delay
  $total \leftarrow 0$
  **for** $k \leftarrow 1$ **to** $N$ **do**
    $total \leftarrow total +$ READADC($Ch_{adc}$)
    Wait 10 ms
  **end for**
  Set $Pin_{pwr} \leftarrow$ LOW ▷ De-energize sensor probe immediately
  $ADC_{avg} \leftarrow total/N$
  $Range \leftarrow ADC_{max} - ADC_{min}$
  $Deadband \leftarrow Range/10$ ▷ 10% noise threshold margin
  ▷ **Deadband and Zero-Level Rejection**
  **if** $Range \leq 0$ **or** $ADC_{avg} < (ADC_{min} + Deadband)$ **then**
    **return** (0%, 0.0 cm)
  **end if**
  ▷ **Linear Depth and Percentage Mapping**
  $pct \leftarrow \min\left(100, \max\left(0, \frac{ADC_{avg} - ADC_{min}}{Range} \times 100\right)\right)$
  $h_{mm} \leftarrow \min\left(L_{mm}, \max\left(0, \frac{ADC_{avg} - ADC_{min}}{Range} \times L_{mm}\right)\right)$
  $h_{cm} \leftarrow h_{mm}/10.0$
  **return** $(pct, h_{cm})$
**end function**

**procedure** MAINTELEMETRYLOOP
  **loop**
    $t_{now} \leftarrow$ CURRENTTIMEMILLIS
    MAINTAINMQTTCONNECTION
    **if** $t_{now} - t_{last} \geq \Delta t_{pub}$ **then**
      $t_{last} \leftarrow t_{now}$
      $(pct_1, h_1) \leftarrow$ SAMPLELEVELSENSOR($D_5, Ch_0, S_{1,min}, S_{1,max}$)
      $(pct_2, h_2) \leftarrow$ SAMPLELEVELSENSOR($D_6, Ch_1, S_{2,min}, S_{2,max}$)
      PUBLISHMQTTTELEMETRY($pct_1, h_1, pct_2, h_2$)
    **end if**
  **end loop**
**end procedure**

**Algorithm 1** WEMOS B - Water Level 3-4

**function** MEASURELEVELPROBE($Pin_{pwr}, Ch_{adc}, ADC_{min}, ADC_{max}$)
  Set $Pin_{pwr} \leftarrow$ HIGH ▷ Power-gate probe ON to prevent galvanic corrosion
  Wait 150 ms ▷ Warmup and charge dissipation delay
  $total \leftarrow 0$
  **for** $k \leftarrow 1$ **to** $N$ **do**
    $total \leftarrow total +$ READADC($Ch_{adc}$)
    Wait 10 ms
  **end for**
  Set $Pin_{pwr} \leftarrow$ LOW ▷ De-energize probe immediately
  $ADC_{avg} \leftarrow total/N$
  $Range \leftarrow ADC_{max} - ADC_{min}$
  $Deadband \leftarrow Range/10$ ▷ 10% noise threshold margin
  ▷ **Zero-Level and Noise Rejection**
  **if** $Range \leq 0$ **or** $ADC_{avg} < (ADC_{min} + Deadband)$ **then**
    **return** (0%, 0.0 cm)
  **end if**
  ▷ **Linear Scaling and Normalization**
  $pct \leftarrow \min\left(100, \max\left(0, \frac{ADC_{avg} - ADC_{min}}{Range} \times 100\right)\right)$
  $h_{cm} \leftarrow \frac{1}{10.0} \times \min\left(L_{mm}, \max\left(0, \frac{ADC_{avg} - ADC_{min}}{Range} \times L_{mm}\right)\right)$
  **return** $(pct, h_{cm})$.
**end function**

**procedure** NODEBTELEMETRYLOOP
  **loop**
    $t_{now} \leftarrow$ CURRENTTIMEMILLIS
    MAINTAINMQTTCONNECTION
    **if** $t_{now} - t_{last} \geq \Delta t_{pub}$ **then**
      $t_{last} \leftarrow t_{now}$
      $(pct_3, h_3) \leftarrow$ MEASURELEVELPROBE($D_5, Ch_0, S_{3,min}, S_{3,max}$)
      $(pct_4, h_4) \leftarrow$ MEASURELEVELPROBE($D_6, Ch_1, S_{4,min}, S_{4,max}$)
      PUBLISHMQTT("wemosB", $pct_3, h_3, pct_4, h_4$)
    **end if**
  **end loop**
**end procedure**

UNIVERSITAS PENDIDIKAN GANESHA
UNDIKSHA

## Appendix 3 Telegram Bot Pseudocode

**Algorithm 1:** Smart Garden IoT Gateway and Telegram Controller

**Input:** MQTT telemetry packets, User Telegram interactions
**Output:** Actuator commands, Database records, User alerts

**State:** Telemetry $\mathcal{D}$, Relays $\mathcal{S}$, Mode $M \in \{\text{AUTO}, \text{MANUAL}\}$, Queue $\mathcal{Q}$, Sessions $\mathcal{T}$

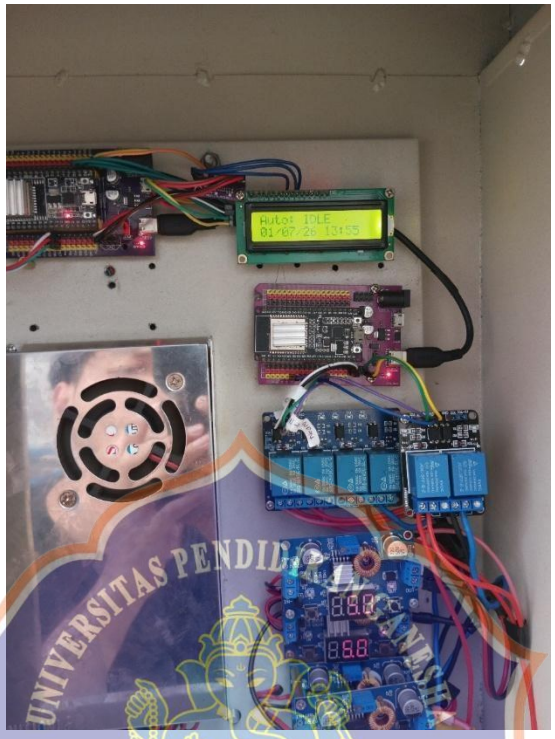
**Procedure** `Main()`
- `InitDatabaseConnection()`
- `StartMQTTBackgroundSubscriber()`
- `RegisterTelegramHandlers()`
- `StartPeriodicScheduler`(telemetry: 300s, alerts: 5s)
- `RunTelegramPollingLoop()`

**Procedure** `OnMQTTMessage`*(topic, payload)*
- **if** *topic* == *TOPIC_DATA* **then**
  - $\mathcal{D} \leftarrow$ `ParseJSON(CleanNaN(`*payload*`))`
  - $\mathcal{D}$["quality"] $\leftarrow$ `ClassifyWaterQuality(`$\mathcal{D}$`)`
- **else if** *topic* $\in$ *RELAY_TOPICS* **then**
  - $(r, s) \leftarrow$ `(ResolveRelay(`*topic*`)`, *payload*$)$
  - $\mathcal{S}[r] \leftarrow s$
  - `LogRelayTransitionToDB(`$r, s, M$`)`
- **else if** *topic* == *TOPIC_NOTIF* **then**
  - $\mathcal{Q}$`.Enqueue(ParseJSON(`*payload*`))`
- **else if** *topic* == *TOPIC_MODE* **then**
  - $M \leftarrow$ *payload*

**Procedure** `ClassifyWaterQuality`*(d)*
- **if** $\exists i \in \{1, \ldots, 4\}$ *s.t.* $d[level_i] \leq 30$ **then**
  - **return** Low Level
- **else if** $d[turbidity] > 50$ **then**
  - **return** $(d[\Delta \text{tds}] \leq 15)$ ? Sediment : Nutrient Cloudiness
- **else if** $d[\Delta tds] > 20$ **then**
  - **return** Nutrient Imbalance
- **return** Normal

**Procedure** `HandleUserCommand`*(user, cmd, args)*
- **if** *cmd* == *LOGIN* **then**
  - **if** *args*[0] == *PASSWORD* **then**
    - $\mathcal{T}[user] \leftarrow$ `CurrentTime()`
  - **return**
- **if** $user \notin \mathcal{T} \vee$ (`CurrentTime()` $- \mathcal{T}[user] > 3600$) **then**
  - **return** `Reply(`*user*, "Unauthorized"`)`
- $\mathcal{T}[user] \leftarrow$ `CurrentTime()`
- **if** *cmd* $\in$ {*STATUS*, *ALL_DATA*} **then**
  - `SendDashboard(`*user*, D, S, M`)`
- **else if** *cmd* == *SET_RELAY* **then**
  - **if** $M$ == *AUTO* **then**
    - `Reply(`*user*, "Error: Manual Mode Required"`)`
  - **else**
    - `MQTTPublish(Topic(`*args*[relay]`)`, *args*[state]`)`
- **else if** *cmd* == *EXPORT* **then**
  - *file* $\leftarrow$ `ExportToCSV(`*args*[table], *args*[days]`)`
  - `SendDocument(`*user*, *file*`)`
  - `Delete(`*file*`)`

## Appendix 4 Other Documentation



UNIVERSITAS PENDIDIKAN GANESHA
UNDIKSHA