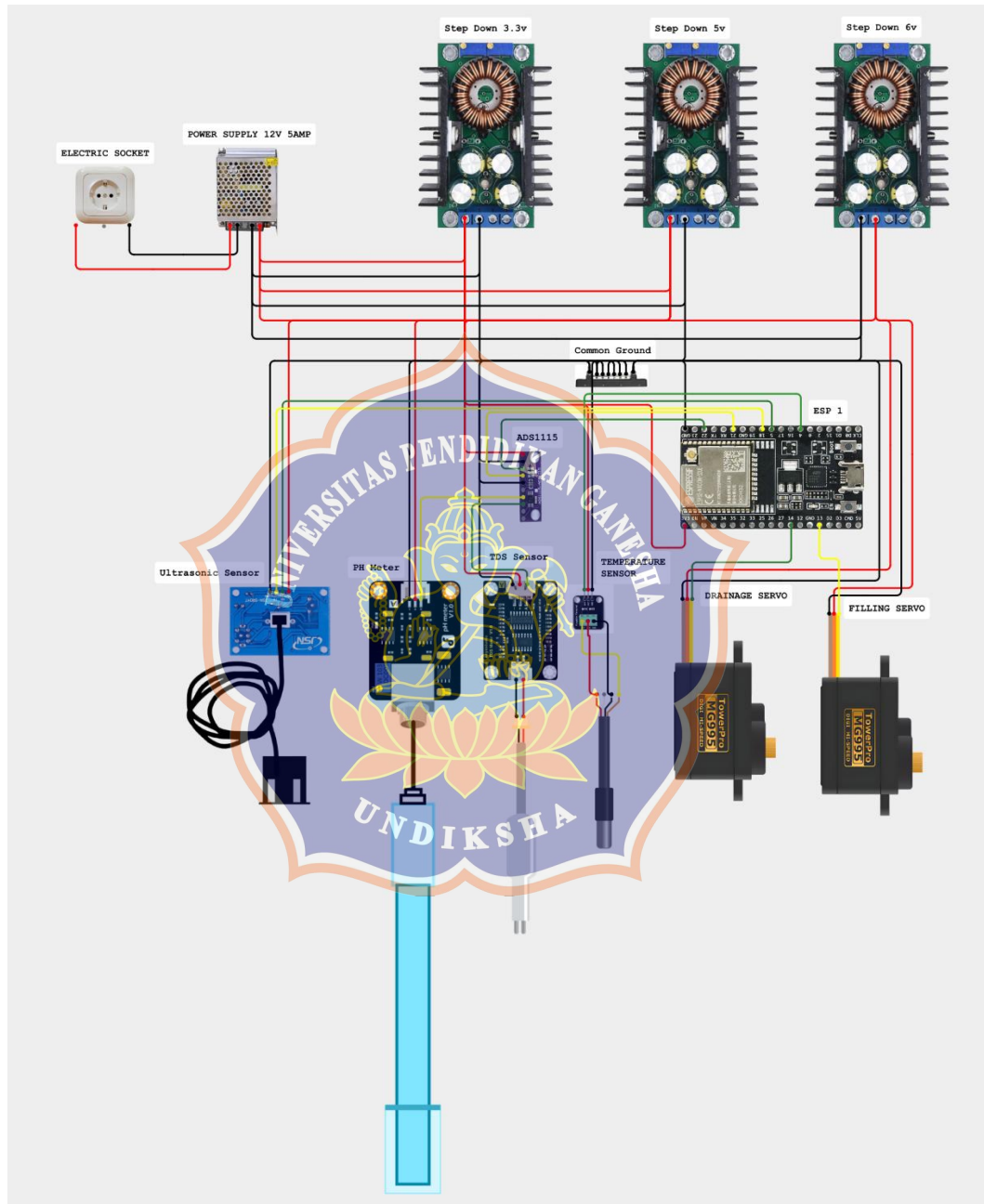
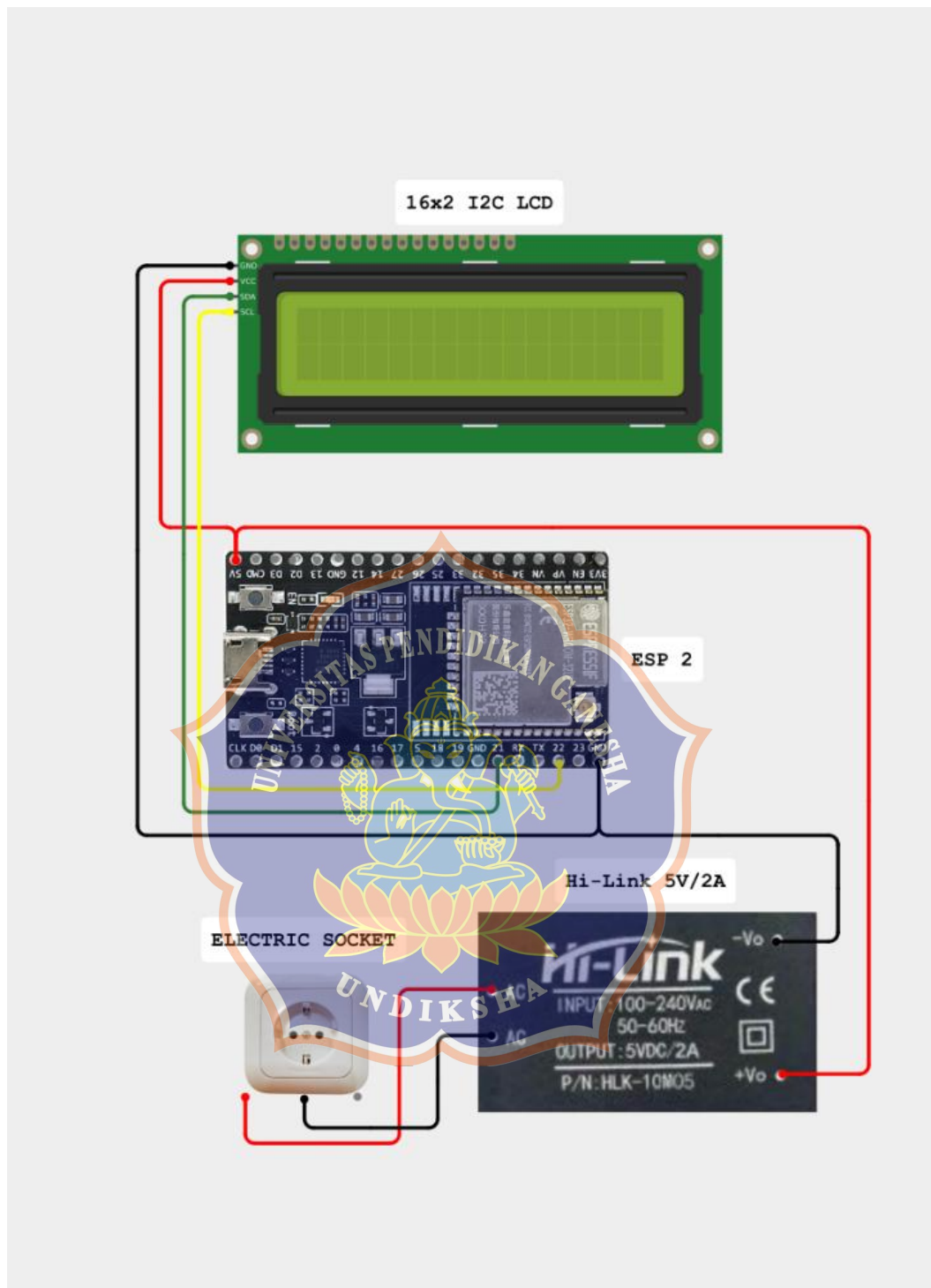


APPENDICES

Appendices 1 Device Wiring Diagram





Appendices 2 Image of the draining and filling actuator



Appendices 3 Image of ESP 1 and ESP 2 panel boxes



Appendices 4 Image of a code snippet

```

1 #include <esp_now.h>
2 #include <WiFi.h>
3 #include <esp_wifi.h>
4 #include <Wire.h>
5 #include <Adafruit_ADS1X15.h>
6 #include <OneWire.h>
7 #include <DallasTemperature.h>
8 #include <ESP32Servo.h>
9 #include <time.h>
10 #include <sys/time.h>
11 #include <Preferences.h>
12
13 // =====
14 // KONFIGURASI PIN & JARINGAN
15 // =====
16 #define PIN_Suhu 4
17 #define PIN_SDA 21
18 #define PIN_SCL 22
19 #define PIN_SERVO_ISI 13
20 #define PIN_SERVO_KURAS 14
21 #define PIN_TRIG 5
22 #define PIN_ECHO 18
23
24 // PIN RELAY UNTUK ISOLASI SENSOR
25 #define PIN_RELAY_PH 26
26 #define PIN_RELAY_TDS 27
27
28 #define MAX_CHANNEL 13
29 #define MAX_FAIL 3
30
31 uint8_t gatewayAddress[] = { 0x00, 0x70, 0x07, 0x26, 0x47, 0xFC };
32
33 // =====
34 // KALIBRASI DASAR PABRIK
35 // =====
36 const float TDS_M = 445.7;
37 const float TDS_C = -31.6;
38
39 const float PH_M = 2.87184;
40 const float PH_C = 2.99850;
41
42 const float OFFSET_SUHU = 0.43;
43
44 // =====
45 // KONFIGURASI SUDUT SERVO INDEPENDEN
46 // =====
47 const int SERVO_ISI_BUKA = 75;
48 const int SERVO_ISI_TUTUP = 175;
49 const int SERVO_KURAS_BUKA = 75;
50 const int SERVO_KURAS_TUTUP = 175;
51
52 const int BATAS_ATAS = 20;
53 const int BATAS_BAWAH_SEDANG = 28;
54 const int BATAS_BAWAH_BURUK = 36;
55
56 const unsigned long INTERVAL_SIRKULASI = 2000;
57 const unsigned long DURASI_JEDA = 5000;
58
59 // =====
60 // OBJECT & STRUCT (DIPERBARUI)
61 // =====

```

```

1 #include <esp_now.h>
2 #include <WiFi.h>
3 #include <LiquidCrystal_I2C.h>
4 #include <WiFiClientSecure.h>
5 #include <UniversalTelegramBot.h>
6 #include <time.h>
7 #include <HTTPClient.h>
8 #include <LittleFS.h>
9 #include <queue>
10
11 #define WIFI_SSID "DWP HARMONI"
12 #define WIFI_PASSWORD "harmoni2024"
13
14 #define BOT_TOKEN "8080608932:AAG5cH6eRXiT9roLXe21OG8uPXab5sjaVno"
15 #define CHAT_ID "870018227"
16 const String URL_GOOGLE_SHEETS = "https://script.google.com/macros/s/
    AKfycbwteRkQOQFk9jAtFyscAcsCvwx7PVKQDijQp0XN4AM6ErWZ0cOfdg4vPDD3HKIDVOU
    /exec";
17
18 WiFiClientSecure secured_client; UniversalTelegramBot bot(BOT_TOKEN,
    secured_client);
19 const long gmtOffset_sec = 8 * 3600; const int daylightOffset_sec =
    0;
20 LiquidCrystal_I2C lcd(0x27, 16, 2); const int TINGGI_TOTAL_SENSOR =
    100;
21
22 typedef struct struct_message {
23     float suhu;
24     float ph;
25     int tds;
26     int tinggi_air;
27     bool fuzzy_active;
28     bool otomasi_active;
29     int command_servo;
30     float skor_fuzzy;
31     char status[32];
32     char type[16];
33     unsigned long epoch;
34     int command_kalibrasi;
35     float nilai_kalibrasi;
36 } struct_message;
37
38 struct_message sensorData; struct_message commandData;
39 uint8_t macESP1[6];
40
41 bool esp1_terhubung = false, wasWiFiConnected = false, isEspNowLost =
    false;
42 unsigned long lastEspNowData = 0, lastTelegramSent = 0, lastLCDUpdate
    = 0, lastTimeUpdate = 0, lastBotCheck = 0, lastWiFiCheck = 0,
    lastQueueProcess = 0, lastSystemCheck = 0,
    lastWiFiReconnectAttempt = 0;
43 int lcdState = 0;
44 std::queue<String> telegramQueue; const int MAX_TELEGRAM_QUEUE = 50;
45
46 void pushTelegram(String msg) { if (telegramQueue.size() <
    MAX_TELEGRAM_QUEUE) telegramQueue.push(msg); }
47
48 void writeToQueue() {
49     struct tm timeinfo; if (!getLocalTime(&timeinfo)) return;

```