LAMPIRAN - LAMPIRAN

A. Datasheet Sensor

1. Datasheet Wemos D1 R1

No	Pin	Function	ESP-8266 Pin
1	TX	TXD	TXD
2	RX	RXD	RXD
3	A0	Analog input, max 3.3V input	A0
4	D0	IO	GPIO16
5	D1	IO, SCL	GPIO5
6	D2	IO, SDA	GPIO4
7	D3	IO, 10k Pull-up	GPIO0
8	D4	IO, 10k Pull-up, BUILTIN_LED	GPIO2
9	D5	IO, SCK	GPIO14
10	D6	IO, MISO	GPIO12
11	D7	IO, MOSI	GPIO13
12	D8	IO, 10k Pull-down, SS	GPIO15
13	G	Ground	GND
14	5V	5V	–
15	3V3	3.3V	3.3V
16	RST	Reset	RST

2. Datasheet Sensor PH Air 4502C

No.	Nama Pin	Deskripsi
1	TO (Temperature Output)	Output suhu
2	DO (3.3V pH Limit Trigger)	Pemicu batas pH 3.3V
3	PO (PH Analog Output)	Output Analog pH
4	Gnd (Gnd for PH Probe)	Ground untuk probe pH
5	Gnd (Gnd for Board)	Ground untuk papan
6	VCC (5V DC)	Daya 5V DC
7	POT 1 (Analog Reading Offset)	Penyetelan bacaan analog offset (Terdekat dengan konektor BNC)
8	POT 2 (PH Limit Setting)	Penyetelan batas pH

3. Datasheet sensor loadcell

Mekanikal:		
No.	**Fitur**	**Deskripsi**
1	Material Housing	Aluminium Alloy
2	Tipe Load Cell	Strain Gauge
3	Kapasitas	20kg
4	Dimensi	55.25x12.7x12.7mm
5	Lubang Pemasangan	M5 (Ukuran Sekrup)
6	Panjang Kabel	550mm
7	Ukuran Kabel	30 AWG (0.2mm)
8	Jumlah Kabel	4
Elektrikal:		
No.	**Parameter**	**Nilai**
1	Presisi	0.05%
2	Rated Output (Keluaran Terukur)	1.0±0.15 mv/V
3	Non-Linearity (Nonlinearitas)	0.05% FS
4	Hysteresis (Histereisis)	0.05% FS
5	Non-Repeatability (Tidak Konsisten)	0.05% FS
6	Creep (per 30 menit)	0.1% FS
7	Efek Suhu pada Nol (per 10°C)	0.05% FS
8	Efek Suhu pada Rentang (per 10°C)	0.05% FS
9	Zero Balance (Keseimbangan Nol)	±1.5% FS

10	Impedansi Input	1130±10 Ohm
11	Impedansi Output	1000±10 Ohm
12	Resistansi Insulasi (di bawah 50VDC)	≥5000 MOhm
13	Voltage Eksitasi	5 VDC
14	Rentang Temperatur Terkompensasi	-10 to ~+40°C
15	Rentang Temperatur Operasional	-20 to ~+55°C
16	Beban Aman (Safe Overload)	120% Kapasitas
17	Beban Maksimum (Ultimate Overload)	150% Kapasitas

4. Datasheet HX711

No.	Nama Pin	Fungsi	Deskripsi
1	VSUP	Power Regulator	Pasokan Regulator: 2.7 ~ 5.5V
2	BASE	Analog Output Regulator	Kendali output regulator (NC ketika tidak digunakan)
3	AVDD	Power Analog	Pasokan Analog: 2.6 ~ 5.5V
4	VFB	Analog Input	Masukan kendali regulator (hubungkan ke AGND ketika tidak digunakan)
5	AGND	Ground Analog	Ground Analog
6	VBG	Analog Output	Output lewat referensi
7	INA-	Analog Input	Saluran A input negatif
8	INA+	Analog Input	Saluran A input positif
9	INB-	Analog Input	Saluran B input negatif
10	INB+	Analog Input	Saluran B input positif
11	PD_SCK	Digital Input	Kontrol daya mati (aktif tinggi) dan input clock serial
12	DOUT	Digital Output	Keluaran data serial
13	XO	Digital I/O	I/O kristal (NC ketika tidak digunakan)
14	XI	Digital Input	I/O kristal atau input clock eksternal, 0: gunakan osilator chip
15	RATE	Digital Input	Kontrol laju data keluar, 0: 10Hz; 1: 80Hz
16	DVDD	Power Digital	Pasokan Digital: 2.6 ~ 5.5V

5. Datasheet Motor Servo MG966

No.	Parameter	Nilai
1	Berat	55 g
2	Dimensi	40.7 x 19.7 x 42.9 mm (sekitar)
3	Torsi Tertahan	9.4 kgf·cm (4.8 V), 11 kgf·cm (6 V)
4	Kecepatan Operasi	0.17 s/60º (4.8 V), 0.14 s/60º (6 V)
5	Tegangan Operasi	4.8 V hingga 7.2 V
6	Arus Berjalan	500 mA
7	Arus Tertahan	2.5 A (6 V)
8	Lebar Dead Band	5 µs
9	Desain Bantalan Bola	Ganda, Stabil, Tahan Goncangan
10	Rentang Suhu	0 ºC hingga 55 ºC

6. Datasheet buzzer

No	**Parameter**	**Nilai**
1	Min. Sound Pressure Level at 12V DC /30cm (dB(A))	85 dB(A)
2	Rated Voltage (VDC)	12 VDC
3	Operating Voltage Range (VDC)	3 ~ 24 VDC
4	Max. Consumption (mA)	12 mA/12V DC
5	Oscillating Frequency (kHz)	3.5 ± 0.5 kHz
6	Tone Nature	Continuous
7	Operating Temperature (°C)	-20 ~ +70
8	Case Material/Color	PC/Black
9	Weight (g)	4 g

B. Kodingan Keseluruhan

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClient.h>
#include <HX711.h>
#include <Servo.h>
#include <ph4502c_sensor.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#include <ezTime.h>

//Your Domain name with URL path or IP address with path
const char* ssid = "Rahmad";
const char* password = "12345678";
const char* serverSend = "http://djmukti.darmajaya.ac.id/ikan/api/store.php";
const char* jadwalCheck = "http://djmukti.darmajaya.ac.id/ikan/api/cek_jadwal.php";
String payload, berat, ph, currentTime, Info;
boolean feed = false;

// Init Pin PH probe
#define PH4502C_PH_LEVEL_PIN A0
#define PH4502C_TEMP_PIN     D2
#define LOADCELL_DOUT_PIN    D7
#define LOADCELL_SCK_PIN     D8
#define LED                  D4
#define BUZZER               D5
#define SERVO_PIN            D9

PH4502C_Sensor ph4502(PH4502C_PH_LEVEL_PIN, PH4502C_TEMP_PIN);
```

```
HX711 scale;
Servo servo;
WiFiClient client;
Timezone timezone;

const float SCALE_VAL = 192.34;
const int   LOW_FEED = 500;
const int   FEED_TIME = 4000;
const int   OPEN_SERVO = 90;
const int   CLOSE_SERVO = 0;
float adc_resolution = 1023.0;

void setup() {
  Serial.begin(9600);
  scale.begin(LOADCELL_DOUT_PIN, LOADCELL_SCK_PIN);
  scale.set_scale(SCALE_VAL);
  scale.tare();
  servo.attach(SERVO_PIN);
  servo.write(CLOSE_SERVO);
  ph4502.init();
  pinMode(LED, OUTPUT);
  digitalWrite(LED, HIGH);
  WiFi.begin(ssid, password);
  Serial.println("Connecting");
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    digitalWrite(LED, !digitalRead(LED));
  }
  Serial.println("");
  Serial.print("Connected to WiFi network with IP Address: ");
```

```
  Serial.println(WiFi.localIP());
  waitForSync();
  timezone.setLocation(F("Asia/Jakarta"));
  pinMode(BUZZER, OUTPUT);
  digitalWrite(BUZZER, LOW);
}

void loop() {
  currentTime = timezone.dateTime("H:i"); // Sync Waktu
  Serial.print("Waktu saat ini => ");
  Serial.println(currentTime);

  getWeight(); // Get data berat pakan
  getPh(); // Get pH
  sendServer(); // Kirim ke server
  getServer(); // Cek Jadwal

  // Control Servo & Motor DC
  if (payload.indexOf(currentTime) != -1 && feed == false) {
    servo.write(OPEN_SERVO); // Buka Pakan
    Serial.println("Memberi Pakan");
    feed = true;
    delay(FEED_TIME); // Lama Pakan Yang Keluar
    servo.write(CLOSE_SERVO); // Tutup Pakan
    Serial.println("Selesai Memberi Pakan");
    delay (6000);
  }
  if (payload.indexOf(currentTime) == -1 && feed == true) {
    feed = false;
    Serial.println("Menunggu Waktu Pakan Selanjutnya");
```

```
  }
  delay(3000);
}

void getWeight() {
  if (scale.is_ready()) {
    Serial.print("Place a known weight on the scale...");
    long reading = scale.get_units(10);
    if (reading < LOW_FEED) {
      Info = "Pakan akan habis, Segera isi ulang.\n";
      digitalWrite(BUZZER, HIGH);
      delay(100);
      digitalWrite(BUZZER, LOW);
      delay(100);
      digitalWrite(BUZZER, HIGH);
      delay(100);
      digitalWrite(BUZZER, LOW);
    } else {
      digitalWrite(BUZZER, LOW);
      Info = "";
    }
    Serial.print("Result: ");
    Serial.println(reading);
    berat = (String)reading;
  } else {
    Serial.println("HX711 not found.");
    Info = "Sensor HX711 Tidak Terdeteksi.\n";
  }
}
```

```
void getPh() {
  ph = String(ph4502.read_ph_level());
  Serial.println("pH Level Reading: " + ph);
  if (ph4502.read_ph_level() < 6.5) Info += "pH dibawah pH ideal.\n";
  if (ph4502.read_ph_level() > 7.5) Info += "pH melebihi pH ideal.\n";
}

void sendServer() {
  HTTPClient http;
  http.begin(client, serverSend);
  http.addHeader("Content-Type", "application/x-www-form-urlencoded");
  String httpRequestData = "berat=" + berat + "&ph=" + ph + "&info=" + Info;
  int responseHttp = http.POST(httpRequestData);
  Serial.print("Sending Code Response => ");
  Serial.println(responseHttp);
  http.end();
}

void getServer() {
  HTTPClient http;
  http.begin(client, jadwalCheck);
  int httpResponseCode = http.GET();
  Serial.print("Getting Code Response => ");
  Serial.println(httpResponseCode);
  payload = (httpResponseCode == 200) ? http.getString() : payload;
  Serial.print("Jadwal Pakan => ");
  Serial.println(payload);
  http.end();
}
```