LAMPIRAN

Lampiran -1 Listing Program Esp 32

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <WiFi.h>
#include <FirebaseESP32.h>
#include <time.h>

#define WIFI_SSID "Vio"
#define WIFI_PASSWORD "adl220180020"

#define     FIREBASE_HOST     "https://panelsurya-f15ad-
default-rtdb.firebaseio.com"
#defineFIREBASE_AUTH"7ae7WV2viAk73oNbq5t1NuJXOZ0WMdT3fo
81Uc9c"
#define ANALOG_IN_PIN   34
#define RELAY_CHARGE    25
#define REF_VOLTAGE     3.3
#define ADC_RESOLUTION  4096.0
#define R1              30000.0
#define R2              7500.0

LiquidCrystal_I2C lcd(0x27, 16, 2);

// Firebase objects
FirebaseData fbdo;
FirebaseAuth auth;
FirebaseConfig config;

unsigned long lastSwitchTime = 0;
const unsigned long delayRelay = 10000;

bool charging = true;

const float teganganOff = 13.8;
const float teganganOn  = 12.2;

// Faktor koreksi linear
const float koreksi = 1.087;

// Konversi tegangan → kapasitas persen
int teganganToPersen(float tegangan) {
  if (tegangan >= 14.4) return 100;
  else if (tegangan >= 13.8)
    return map(tegangan * 100, 1380, 1440, 85, 100);
```

```
  else if (tegangan >= 12.2)
    return map(tegangan * 100, 1220, 1380, 30, 85);
  else if (tegangan >= 11.0)
    return map(tegangan * 100, 1100, 1220, 0, 30);
  else
    return 0;
}
float getAverageVoltage(int pin) {
  long total = 0;
  for (int i = 0; i < 25; i++) {
    total += analogRead(pin);
    delay(20);
  }
  floatvoltageADC = ((float)total / 25.0) * REF_VOLTAGE
/ ADC_RESOLUTION;
  float teganganBaterai = voltageADC * ((R1 + R2) / R2)
* koreksi;

  return teganganBaterai;
}
String getDateTime() {
  time_t now = time(nullptr);
  struct tm* t = localtime(&now);
  char buf[25];
  sprintf(buf, "%04d-%02d-%02d %02d:%02d:%02d",
    1900 + t->tm_year, 1 + t->tm_mon, t->tm_mday,
    t->tm_hour, t->tm_min, t->tm_sec);
  return String(buf);
}
void setup() {
  Serial.begin(9600);
  WiFi.mode(WIFI_STA);
  WiFi.disconnect();
  delay(1000);

  analogSetAttenuation(ADC_11db);

  pinMode(RELAY_CHARGE, OUTPUT);
  digitalWrite(RELAY_CHARGE, LOW);

  lcd.init     ();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("Sistem Kendali");

  Serial.println();
  Serial.println();
```

```
  Serial.print("Connecting to: ");
  lcd.print("Hubungkan ke Hotspot");

  Serial.println(WIFI_SSID);
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);

  while (WiFi.status() != WL_CONNECTED) {
    Serial.print("-");
    lcd.print("menghubungkan......");
    delay(500);
  }
  Serial.println();
  Serial.println("WiFi Connected");
  lcd.print("OK WiFi Terhubung");
  Serial.println();

  lcd.setCursor(0, 1);
  lcd.print("Pengisian bat.");
  delay(2000);
  lcd.clear();

  config.host = FIREBASE_HOST;
  config.signer.tokens.legacy_token = FIREBASE_AUTH;
  Firebase.begin(&config, &auth);
  Firebase.reconnectWiFi(true);
  configTime(8*3600,         0,         "pool.ntp.org",
"time.nist.gov");
  Serial.println("Awaiting NTP sync...");
  while (!time(nullptr)) {
    Serial.print(".");
    delay(500);
  }
  Serial.println("\nNTP synced");
}

void loop() {

  float                 teganganBaterai                 =
getAverageVoltage(ANALOG_IN_PIN);

  unsigned long now = millis();

  String waktu = getDateTime();

  int kapasitas = teganganToPersen(teganganBaterai);


  if (teganganBaterai < 1.0) {
```

```
    lcd.setCursor(0, 0);
    lcd.print("Teg:0.00V       ");
    lcd.setCursor(0, 1);
    lcd.print("No Input        ");
    Serial.println("Tegangan: 0.00 V | No Input");
    delay(1000);
    return;
  }

  if (teganganBaterai >= teganganOff && charging) {
    static unsigned long highVoltageStart = 0;
    if (highVoltageStart == 0) highVoltageStart = now;
    if ((now - highVoltageStart) >= delayRelay) {
      digitalWrite(RELAY_CHARGE, HIGH);
      charging = false;
      lastSwitchTime = now;
      Serial.println("Charging   OFF   (tegangan   tinggi
stabil)");
    }
  } else if (teganganBaterai <= teganganOn && !charging
&& (now - lastSwitchTime > delayRelay)) {
    digitalWrite(RELAY_CHARGE, LOW);
    charging = true;
    lastSwitchTime = now;
    Serial.println("Charging ON (tegangan rendah)");
  } else {
    static unsigned long highVoltageStart = 0;
    highVoltageStart = 0;
  }
```

```
  // LCD display
  lcd.setCursor(0, 0);
  lcd.print("Teg:");
  lcd.print(teganganBaterai, 2);
  lcd.print("V     ");

   lcd.setCursor(0, 1);
  lcd.print(charging ? "Chg:ON " : "Chg:OFF");
  lcd.setCursor(9, 1);
  lcd.print(kapasitas);
  lcd.print("% ");

  // Serial debug
  Serial.print("Tegangan: ");
  Serial.print(teganganBaterai, 2);
  Serial.print(" V | Charging: ");
  Serial.print(charging ? "ON" : "OFF");
  Serial.print(" | Kapasitas: ");
  Serial.print(kapasitas);
  Serial.println("%");

  // Simpan data realtime (update)
  Firebase.setString(fbdo,
"/solar_system/realtime_data/kapasitas_baterai",
String(kapasitas));
  Firebase.setString(fbdo,
"/solar_system/realtime_data/status_charging",  charging
? "on" : "off");

  // Simpan data historis (push)
  FirebaseJson json;
```

```
  json.set("waktu", String(waktu ));
  json.set("tegangan_baterai",   String(teganganBaterai,
2));
  json.set("kapasitas_baterai", String(kapasitas));
  json.set("status_charging", charging ? "on" : "off");


  if    (Firebase.pushJSON(fbdo,    "/solar_system/logs",
json)) {
Serial.println("✓     Data     berhasil     dipush     ke
/solar_system/logs");
  } else {
    Serial.print("✗ Gagal push Firebase: ");
    Serial.println(fbdo.errorReason());
  }
  delay(1000);
 }
```

Lampiran -2 Listing Program HTML

```
<!DOCTYPE html>
<html lang="id">
<head>
  <meta charset="UTF-8" />
  <meta  name="viewport"  content="width=device-width,
initial-scale=1.0" />
  <title>Monitoring Panel Surya IoT</title>
  <link
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.0.0-beta3/css/all.min.css" rel="stylesheet">
  <script
src="https://www.gstatic.com/firebasejs/5.0.4/firebase.
js"></script>

  <style>
    * {
      margin: 0;
      padding: 0;
      box-sizing: border-box;
      font-family: sans-serif;
    }
    body {
      background: #f1f1f1;
      padding: 20px;
    }
    h1 {
      text-align: center;
      margin-bottom: 20px;
      color: #4CAF50;
      font-size: 2em;
    }
    .grid-row {
      display: grid;
      grid-template-columns:          repeat(auto-fit,
minmax(250px, 1fr));
      gap: 20px;
      max-width: 1200px;
      margin: 0 auto 20px auto;
    }
    .card {
      background-color: #fff;
      border-radius: 10px;
```

```
      box-shadow: 0 4px 10px rgba(0, 0, 0, 0.15);
      padding: 20px;
      text-align: center;
      background:   linear-gradient(135deg,   #e0f7fa,
#fce4ec);
      transition: transform 0.2s;
    }
    .card:hover {
      transform: scale(1.05);
    }
    .card i {
      font-size: 3em;
      color: #00796B;
      margin-bottom: 10px;
    }
    h2 {
      font-size: 1.5em;
      color: #333;
      margin-bottom: 10px;
    }
    .sensor {
      display: flex;
      flex-wrap: wrap;
      justify-content: center;
      gap: 10px;
      margin-top: 10px;
    }
    .sensor div {
      background-color: #fff;
      color: #333;
      padding: 10px;
      border-radius: 10px;
      width: 45%;
      box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
      text-align: center;
    }
    .tegangan  {  background-color:  #e0f7fa;  color:
#00796B; }
    .suhu { background-color: #ffeef2; color: #e91e63;
}
    .lux { background-color: #ffe0b2; color: #f57c00; }
    .status { background-color: #f3e5f5; color: #6A1B9A;
}
```

```
    .battery { background-color: #fff3e0; color:
#ff9800; }

    p {
      font-size: 1.5em;
      margin: 0;
    }

    footer {
      text-align: center;
      margin-top: 30px;
      color: #888;
      font-size: 0.9em;
    }
  </style>

  <script>
    var config = {
      apiKey:
"AIzaSyAiBrr9yTwIdIyKxy337PdkGkjXZVM_lNs",
      databaseURL: "https://panelsurya-f15ad-default-
rtdb.firebaseio.com/"
    };
    firebase.initializeApp(config);
    var f_database = firebase.database();


f_database.ref("solar_system/realtime_data").on("value"
, function(snapshot) {
      var p1 = snapshot.child("panel_1");

document.getElementById("teganganPanel").innerHTML =
p1.child("tegangan_panel").val() + " V";
      document.getElementById("suhuPanel").innerHTML =
p1.child("suhu_panel").val() + " °C";
      document.getElementById("luxPanel").innerHTML =
p1.child("lux_panel").val() + " lux";

      var p2 = snapshot.child("panel_2");

document.getElementById("teganganPanel2").innerHTML =
p2.child("tegangan_panel").val() + " V";
```

```
      document.getElementById("suhuPanel2").innerHTML =
p2.child("suhu_panel").val() + " °C";
      document.getElementById("luxPanel2").innerHTML  =
p2.child("lux_panel").val() + " lux";

      var p3 = snapshot.child("panel_3");

document.getElementById("teganganPanel3").innerHTML   =
p3.child("tegangan_panel").val() + " V";
      document.getElementById("suhuPanel3").innerHTML =
p3.child("suhu_panel").val() + " °C";
      document.getElementById("luxPanel3").innerHTML  =
p3.child("lux_panel").val() + " lux";

      var                   kapasitas                   =
snapshot.child("kapasitas_baterai").val();

document.getElementById("kapasitasBaterai").innerHTML =
kapasitas + " %";

      var                     status                    =
snapshot.child("status_charging").val();

document.getElementById("statusCharging").innerHTML   =
status;
    });
  </script>
</head>

<body>
  <h1>Sistem Monitoring & Kendali Panel Surya</h1>

  <div class="grid-row">
    <div class="card">
      <h2>Monitoring Panel 1</h2>
      <i class="fas fa-solar-panel"></i>
      <div class="sensor">
        <div                          class="tegangan"><p
id="teganganPanel">...</p></div>
        <div                              class="suhu"><p
id="suhuPanel">...</p></div>
        <div class="lux"><p id="luxPanel">...</p></div>
      </div>
```

```
    </div>

    <div class="card">
      <h2>Monitoring Panel 2</h2>
      <i class="fas fa-solar-panel"></i>
      <div class="sensor">
        <div                          class="tegangan"><p
id="teganganPanel2">...</p></div>
        <div                              class="suhu"><p
id="suhuPanel2">...</p></div>
        <div                               class="lux"><p
id="luxPanel2">...</p></div>
      </div>
    </div>

    <div class="card">
      <h2>Monitoring Panel 3</h2>
      <i class="fas fa-solar-panel"></i>
      <div class="sensor">
        <div                          class="tegangan"><p
id="teganganPanel3">...</p></div>
        <div                              class="suhu"><p
id="suhuPanel3">...</p></div>
        <div                               class="lux"><p
id="luxPanel3">...</p></div>
      </div>
    </div>
  </div>

  <div class="grid-row">
    <div class="card">
      <h2>Kapasitas Baterai</h2>
      <i class="fas fa-battery-half"></i>
      <div class="sensor">
        <div                           class="battery"><p
id="kapasitasBaterai">...</p></div>
      </div>
    </div>

    <div class="card">
      <h2>Kondisi Charger</h2>
      <i class="fas fa-plug"></i>
      <div class="sensor">
```

```
        <div                              class="status"><p
id="statusCharging">...</p></div>
      </div>
    </div>
  </div>

  <footer>
    <p>© 2025 Monitoring System</p>
  </footer>
</body>
</html>
```

Lampiran -3 Data Sheet Sensor Tegangan

Arduino 25V Voltage Sensor Module

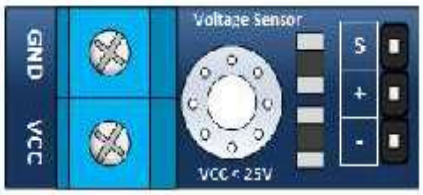
The Basics

The Arduino analog input is limited to a 5 VDC input. If you wish to measure higher voltages, you will need to resort to another means. One way is to use a voltage divider. The one discussed here is found all over Amazon and eBay.

It is fundamentally a 5:1 voltage divider using a 30K and a 7.5K Ohm resistor.

Keep in mind, you are restricted to voltages that are less than 25 volts. More than that and you will exceed the voltage limit of your Arduino input.

Basic Connection



Inputs

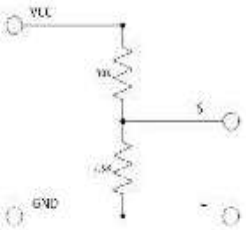
- **GND –** This is where you connect the low side of the voltage you are measuring. Caution! : This is the same electrical point as your Arduino ground.
- **VCC:** The is where you connect the high side of the voltage you are measuring

Outputs

- **S:** This connects to your Arduino analog input.
- **– (or minus):** This connects to your Arduino ground.
- **+:** This is not connected. It does absolutely nothing... zilch... nada... jack diddly doo doo.

Schematic

The schematic for this is pretty straight forward. As previously mentioned, its just a couple of resistors. In fact, you could build your own in a pinch.



Lampiran -4 Data Sheet Module Relay 2 Chanel

Name: **Relay Module 2-Channel**
Code: **MR009-004.1**



This *Relay Module 2-Channel* is a module designed to allow you to control two relays in a very simple and intuitive manner. Being compatible with Arduino, the most immediate way to use it is to connect it to an Arduino board using flexible jumpers.

Exploiting the characteristics of the relays mounted on the module and through the use of two Arduino digital I/O pins, it is possible to control motors, inductive loads and other devices; this product is therefore fundamental in domotics projects or, more in general, in robotics projects.

The module is equipped with optocouplers on *IN1* and *IN2* lines in such a way that it ensures the galvanic insulation between the relay load and the control board which drives this module.

CONNECTIONS

Pin	*Function*
IN1	TTL digital input
IN2	TTL digital input
GND	Ground
+5V	Power (+5V)
NO1	Normally open contact
COM1	Common contact
NC1	Normally closed contact

MICROBOT di Prosseda Mirko – Strada Chiesuola 117, 04010 Borgo Carso (LT), Italy
web: http://www.microbot.it email: info@microbot.it tel: +39-347-2159276

NO2	Normally open contact
COM2	Common contact
NC2	Normally closed contact

Tab.1 – Connections

CHARACTERISTICS

Pin	Function
Supply voltage	+5V
Supply current	144mA typ. (160mA max.)
Current on pin IN	14mA typ.
Rated load	7A 250VAC
Operating temperature	-30°C / +70°C
Operate time max.	10ms Max.
Release time max.	5ms Max.
Insulation resistance	100Mohm Min.
Mechanical Life Expectancy	10,000,000 operations
Electrical Life Expectancy	10,000 operations
Dimensions	1.7" x 1.3" (43.2 x 33.0 mm)
Weight	0,92oz (26,2g)

Tab.2 - Characteristics

