



**RANCANG BANGUN ALAT PENGUKUR JARAK  
TERINTEGRASI ANDROID**

**IMPLEMENTASI *ROTARY ENCODER* PADA ALAT  
PENGUKUR JARAK**

**TUGAS AKHIR**

**Islah Rulia Hasanah**

**1803321007**

**PROGRAM STUDI ELEKTRONIKA INDUSTRI**

**JURUSAN TEKNIK ELEKTRO**

**POLITEKNIK NEGERI JAKARTA**

**2021**



**RANCANG BANGUN ALAT PENGUKUR JARAK  
TERINTEGRASI ANDROID**

**IMPLEMENTASI *ROTARY ENCODER* PADA ALAT  
PENGUKUR JARAK**

**TUGAS AKHIR**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar  
Diploma Tiga**

**Islah Rulia Hasanah  
1803321007**

**PROGRAM STUDI ELEKTRONIKA INDUSTRI  
JURUSAN TEKNIK ELEKTRO  
POLITEKNIK NEGERI JAKARTA**

**2021**

## HALAMAN PERNYATAAN ORISINALITAS

**Tugas Akhir ini adalah hasil karya saya sendiri dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar.**

**Nama** Islah Rulia Hasanah

**NIM** 1803321007

**Tanda Tangan**

A handwritten signature in black ink, consisting of stylized letters and a flourish.

**Tanggal** 29 Juli 2021

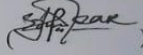
# LEMBAR PENGESAHAN TUGAS AKHIR

## LEMBAR PENGESAHAN TUGAS AKHIR

Tugas Akhir diajukan oleh :

Nama : Islah Rulia Hasanah  
NIM : 1803321007  
Program Studi : Elektronika Industri  
Judul Tugas Akhir : Rancang Bangun Alat Pengukur Jarak Terintegrasi  
Android  
Sub Judul Tugas Akhir : Implementasi *Rotary Encoder* pada Alat Pengukur Jarak

Telah diuji oleh tim penguji dalam Sidang Tugas Akhir pada (HARI DAN TANGGAL) dan dinyatakan LULUS.

Pembimbing I : Endang Saepudin, Dipl.Eng., M.Kom.   
NIP. 196202271992031002

Depok, 19 Agustus 2021

Disahkan Oleh

Ketua Jurusan Teknik Elektro



Ii. Sri Danaryani, M.T.

NIP. 196305031991032001

## KATA PENGANTAR

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat- Nya, penulis mampu menyelesaikan Laporan Tugas Akhir berjudul **“Rancang Bangun Alat Pengukur Jarak Terintegrasi Android”** yang bersubjudul **“Implementasi *Rotary Encoder* pada Alat Pengukur Jarak”** dalam rangka memenuhi salah satu syarat untuk mencapai gelar Diploma Tiga Politeknik.

Penulis menyadari bahwa tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai dengan penyusunan tugas akhir ini, sangatlah sulit bagi penulis untuk menyelesaikan tugas akhir ini. Oleh karena itu, penulis mengucapkan terimakasih kepada:

1. Ir. Sri Danaryani, M.T. selaku Ketua Jurusan Teknik Elektro Politeknik Negeri Jakarta
2. Endang Saepudin, Dipl.Eng., M.Kom. selaku dosen pembimbing yang telah mengarahkan dan membimbing penulis dalam penyusunan tugas akhir
3. Orang tua dan keluarga yang telah memberikan bantuan dukungan, material dan moral
4. Rekan Tugas Akhir yang telah berbagi ilmu, pengalaman, serta memberi semangat kepada penulis.
5. Kepada teman-teman EC B 2018 yang sudah memberi semangat kepada penulis.

Akhir kata, penulis berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga Tugas Akhir ini membawa manfaat bagi pengembangan ilmu di masa yang akan datang.

Depok, 29 Juli 2021

Penulis

## Implementasi *Rotary Encoder* pada Alat Pengukur Jarak

### **Abstrak**

*Kebutuhan alat ukur yang mudah digunakan dan mempunyai nilai ketelitian yang tinggi mendorong semakin banyaknya penciptaan alat ukur. Tugas akhir ini merancang sebuah alat pengukur jarak terintegrasi android yang menggunakan sensor rotary encoder dengan jenis incremental rotary encoder sebagai pendeteksi jaraknya. Jenis ini dapat digunakan untuk mengukur posisi sudut dari sebuah shaft yang berotasi. Pada alat ini sensor rotary encoder dipasang pada roda agar shaft dari rotary encoder dapat berotasi seiring dengan perputaran roda. Sensor ini bekerja dengan mendeteksi setiap perputaran yang terjadi. Ketika shaft pada rotary encoder berotasi, maka sederetan pulsa yang dihasilkan akan muncul pada masing-masing pin keluaran. Selanjutnya hasil pendeteksian sensor diteruskan ke ESP32 untuk diproses agar menjadi data pengukuran jarak. Data hasil pengukuran jarak kemudian akan ditampilkan di LCD berukuran 20x4 dan juga aplikasi android yang sudah terhubung dengan alat. Alat ini dirancang agar dapat dengan nyaman digunakan dan mudah dalam pembacaan hasil pengukuran jarak. Saat alat didorong ke arah depan, angka hasil pengukuran yang ditampilkan di LCD akan bertambah. Saat alat ditarik ke arah belakang, angka hasil pengukuran akan berkurang lalu kemudian berhenti di angka 0. Tujuan pembuatan alat ini adalah meningkatkan ketelitian hasil pengukuran dan menghilangkan keterbatasan jarak pengukuran.*

*Kata kunci: alat ukur, ESP32, pengukuran, rotary encoder*

## *Implementation of Rotary Encoder on Distance Measuring Device*

### ***Abstract***

*The need of measuring instruments that are easy to use and have a high value of accuracy encourage the creation of more measuring instruments. This final project designs an android integrated distance measuring device that uses a rotary encoder sensor with an incremental rotary encoder type as a distance detector. This type can be used to measure the angular position of a rotating shaft. In this device rotary encoder sensor is mounted on the wheel so that the shaft of the rotary encoder can rotate along with the rotation of the wheel. This sensor works by detecting every rotation that occurs. When the shaft on the rotary encoder rotates, a series of pulses are generated which appear on each output pin. Furthermore, the sensor detection results are forwarded to the ESP32 to be processed to become distance measurement data. The distance measurement data will then be displayed on a 20x4 LCD and also an android application that is already connected to the device. This tool is designed to be easy and convenient to use and easy to read distance measurement results. When the tool is pushed forward, the number of measurement results displayed on the LCD will increase. When the tool is pulled backwards, the measurement result will decrease and then stop at 0. The purpose of making this tool is to increase the accuracy of the measurement data and eliminate the limitations of the distance measurement.*

*Keywords: ESP32, measurement, measuring instrument, rotary encoder*

## DAFTAR ISI

HALAMAN PERNYATAAN ORISINALITAS.....	iii
LEMBAR PENGESAHAN .....	iv
KATA PENGANTAR .....	v
Abstrak .....	vi
Abstract .....	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR .....	x
DAFTAR TABEL.....	xi
DAFTAR LAMPIRAN.....	xii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	2
1.3 Tujuan.....	3
1.4 Luaran.....	3
BAB II TINJAUAN PUSTAKA.....	4
2.1 Rotary Encoder .....	4
2.1.1 Prinsip Kerja Rotary Encoder .....	7
2.2 Logic Level Converter.....	7
2.3 ESP32 .....	8
2.4 Arduino IDE .....	10
BAB III PERENCANAAN DAN REALISASI.....	12
3.1 Rancangan Alat.....	12
3.1.1 Deskripsi Alat .....	12
3.1.2 Cara Kerja Alat .....	12



3.1.3	Spesifikasi Alat .....	13
3.1.4	Diagram Blok .....	14
3.2	Realisasi Alat .....	15
3.2.1	Wiring Diagram .....	15
3.2.2	Penginstalasian Sensor Rotary Encoder .....	16
3.2.3	Pemrograman Pengukuran Jarak Menggunakan Arduino IDE .....	17
BAB IV PEMBAHASAN.....		21
4.1	Pengujian .....	21
4.1.1	Deskripsi Pengujian .....	21
4.1.2	Prosedur Pengujian .....	21
4.1.3	Data Hasil Pengujian.....	22
4.1.4	Evaluasi Hasil Pengujian .....	22
BAB V PENUTUP.....		24
5.1	Kesimpulan.....	24
DAFTAR PUSTAKA .....		25

## DAFTAR GAMBAR

Gambar 1. 1 Rol Meter .....	1
Gambar 1. 2 Walking Measure .....	2
Gambar 2. 1 <i>Rotary Encoder</i> .....	4
Gambar 2. 2 <i>Struktur Rotary Encoder</i> .....	5
Gambar 2. 3 Pulsa-pulsa yang dihasilkan <i>rotary encoder</i> .....	5
Gambar 2. 4 <i>Logic Level Converter</i> .....	8
Gambar 2. 5 Menghubungkan perangkat 5V dan 3,3V .....	8
Gambar 2. 6 Pin-pin pada ESP32S .....	9
Gambar 2. 7 Arduino IDE .....	11
Gambar 3. 1 Diagram Blok Sub Alat.....	15
Gambar 3. 2 <i>Wiring Diagram</i> .....	16
Gambar 3. 3 Pemasangan <i>rotary encoder</i> pada alat .....	16
Gambar 3. 4 <i>Rotary encoder</i> terhubung ke ESP32.....	17
Gambar 3. 5 Pemasangan <i>Logic Level Converter</i> pada PCB .....	17
Gambar 3. 6 Program Pengukuran Jarak .....	18
Gambar 3. 7 Program Pengukuran Jarak .....	18
Gambar 3. 8 Program Menampilkan Pengukuran Jarak.....	19
Gambar 3. 9 Program Pengukuran Jarak .....	19
Gambar 3. 10 Program Pengukuran Jarak .....	20

## DAFTAR TABEL

Tabel 2. 1 Spesifikasi <i>Rotary Encoder</i> LPD3806.....	6
Tabel 2. 2 Perbedaan ESP32 dengan mikrokontroler lain.....	9
Tabel 3. 1 Komponen yang digunakan.....	13
Tabel 3. 2 Software yang digunakan.....	14
Tabel 4. 1 Alat yang dibutuhkan.....	21

## DAFTAR LAMPIRAN

LAMPIRAN 1 .....	L-1
LAMPIRAN 2 .....	L-2
LAMPIRAN 3 .....	L-3

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

*Rotary encoder* merupakan sensor optik yang digunakan untuk mengetahui posisi putaran suatu benda. Sensor ini bekerja dengan membaca pulsa yang dihasilkan fototransistor. Ketika poros berputar maka sederetan pulsa akan muncul pada masing-masing pin keluaran (Abilovani, Yahya, & Bakhtiar, 2018). *Rotary encoder* umumnya menggunakan sensor optik untuk menghasilkan serial pulsa yang dapat diartikan menjadi gerakan, posisi, dan arah. Sehingga posisi sudut suatu poros benda berputar dapat diolah menjadi informasi berupa kode digital oleh *rotary encoder* untuk diteruskan oleh rangkaian kendali (Fikri & Endryansyah, 2019). Karena mendeteksi perputaran, *rotary encoder* sering digunakan untuk melakukan suatu pengukuran.

Pengukuran adalah satu bentuk aktivitas membandingkan suatu besaran yang diukur dengan alat ukur. Pengukuran tidak hanya terbatas pada kuantitas fisik. Sesuatu yang dapat diukur dan dapat dinyatakan dengan angka disebut besaran, sedangkan pembanding dalam suatu pengukuran disebut satuan. (Sugih Miftahul Huda, Awaliyah Zuraiyah, & Lukmanul Hakim, 2019). Salah satu besaran yang dapat diukur adalah jarak. Untuk mengetahui harga suatu besaran diperlukan alat ukur.



Gambar 1. 1 Rol Meter

Sumber: <https://ilmugeografi.com/wp-content/uploads/2019/03/5ca8043f-meteran.jpg>

Salah satu alat ukur jarak yang sering digunakan adalah rol meter seperti yang ditunjukkan pada Gambar 1.1 terdapat berbagai jenis rol meter dengan panjang yang berbeda-beda dengan maksimal panjang 100 meter. Selain rol meter, terdapat *walking measure* seperti yang ditunjukkan pada Gambar 1.2 yaitu alat ukur jarak yang memiliki roda dan digunakan dengan cara didorong. Alat ini

dapat mengukur jarak hingga 9.999 meter. Alat ukur ini dapat digunakan pada berbagai macam permukaan. *Walking measure* yang ada saat ini memiliki keterbatasan yaitu komponen yang digunakan pada alat ini masih menggunakan komponen analog dan masih mengandalkan mekanik dalam perhitungan jarak, alat ini memiliki ketelitian yang hanya sebesar 1 m dan pengguna alat ini masih harus menulis secara manual untuk menyimpan data hasil pengukuran. Oleh karena itu dibutuhkan alat pengukur jarak yang praktis digunakan dan menggunakan komponen digital.



Gambar 1. 2 *Walking Measure*

Sumber: <https://www.amazon.co.uk/Lightweight-Compact-Stretchable-Walking-Distance/dp/B01LX5N41O>

Dari permasalahan tersebut, pada tugas akhir ini penulis mengusung sebuah ide untuk mengembangkan alat pengukur jarak yang memiliki roda seperti *walking measure* yang terintegrasi dengan android. Alat ini menggunakan komponen digital seperti mikrokontroler untuk memproses data pengukuran, *Rotary Encoder* digunakan sebagai sensor untuk mengidentifikasi putaran roda, *Liquid Crystal Display (LCD)* sebagai *Human Machine Interface (HMI)* untuk menampilkan data hasil pengukuran, *Buzzer* digunakan sebagai fitur pengingat untuk penanda jarak dan aplikasi android yang digunakan untuk menampilkan dan menyimpan data hasil pengukuran jarak. Alat yang kami rancang ini dapat digunakan untuk mengukur berbagai macam permukaan, tidak memiliki batasan pengukuran, memiliki ketelitian yang lebih baik yaitu sebesar 0.01 m, dan alat ini sudah terintegrasi android.

## 1.2 Perumusan Masalah

- a. Bagaimana cara kerja *rotary encoder* agar dapat mengukur jarak?
- b. Bagaimana instalasi *rotary encoder* pada alat pengukur jarak?

### **1.3 Tujuan**

- a. Mengimplemetasikan *rotary encoder* sebagai sensor pada alat pengukur jarak.
- b. Meningkatkan ketelitian hasil pengukuran dan menghilangkan keterbatasan jarak pengukuran.

### **1.4 Luaran**

- a. Alat Pengukur Jarak Otomatis
- b. Aplikasi Android
- c. Laporan Tugas Akhir

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Kesimpulan yang didapatkan penulis dari Tugas Akhir yang berjudul “Rancang Bangun Alat Pengukur Jarak Terintegrasi Android” dengan sub judul “Implementasi *Rotary Encoder* pada Alat Pengukur Jarak” adalah sebagai berikut:

1. Setiap perputaran *rotary encoder* menghasilkan pulsa gelombang kotak yang selanjutnya diteruskan ke ESP32 untuk diolah menjadi data pengukuran jarak. Penggunaan sensor *rotary encoder* pada alat ini berhasil mengeluarkan output sesuai dengan harapan.
2. Sensor *rotary encoder* dipasang menyatu dengan roda agar dapat berotasi seiring dengan perputaran roda. Untuk menghubungkan *rotary encoder* dengan ESP32 diperlukan *logic level converter* untuk mengubah tegangannya karena *rotary encoder* memiliki tegangan kerja 5V sedangkan ESP32 memiliki tegangan kerja 3.3V.



## DAFTAR PUSTAKA

- Abilovani, Z. B., Yahya, W., & Bakhtiar, F. A. (2018). Implementasi Protokol MQTT Untuk Sistem Monitoring Perangkat IoT. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 7521-7527.
- Amin, M., Triyanto, J., & Sukandar. (2019). Antar Muka Quadrature Rotary Encoder Pada STM32F407VGT6 Brakiterapi HDR IR-192 Menggunakan Modul LS7184N. *Prima*, 24-32.
- Fikri, A. A., & Endryansyah. (2019). Sistem Pengaturan PID Motor DC Sebagai Penggerak Mini Conveyor Berbasis Matlab. *Jurnal Teknik Elektro*, 293-301.
- Muliadi, Al Imran, & Rasul, M. (2020). Pengembangan Tempat Sampah Pintar Menggunakan ESP32. *Jurnal Media Elektrik*, 73-79.
- Nugroho, S. A., Suryawan, I. K., & Wardana, I. N. (2015). Penerapan Mikrokontroler sebagai Sistem Kendali Perangkat Listrik Berbasis Android. *Eksplora Informatika*, 135-144.
- Setiawan, A., & Purnamasari, A. I. (2019). Pengembangan Smart Home Dengan Microcontrollers ESP32 Dan MC-38 Door Magnetic Switch Sensor Berbasis Internet of Things (IoT) Untuk Meningkatkan Deteksi Dini Keamanan Perumahan. *Jurnal Resti (Rekayasa Sistem dan Teknologi Informasi)*, 451-457.
- Sokop, S. J., Mamahit, D. J., & Sompie, S. R. (2016). Trainer Periferal Antarmuka Berbasis Mikrokontroler Arduino Uno. *E-Journal Teknik Elektro dan Komputer vol.5 no.3 (2016), ISSN : 2301-8402* , 13-23.
- Sugih Miftahul Huda, A., Awaliyah Zuraiyah, T., & Lukmanul Hakim, F. (2019). Prototype Alat Pengukur Jarak Dan Sudut Kemiringan Digital Menggunakan Sensor Ultrasonik Dan Accelerometer Berbasis Arduino Nano. *Bina Insani ICT Journal*, 185-194.

## LAMPIRAN 1

### DAFTAR RIWAYAT HIDUP PENULIS



Islah Rulia Hasanah

Anak pertama dari empat bersaudara. Lahir di Jakarta, 8 November 2000. Lulus dari SD Islam Amaryllis tahun 2012, MTs Negeri 7 Jakarta tahun 2015, MAN 15 Jakarta tahun 2018. Gelar Diploma Tiga (D3) diperoleh pada tahun 2021 dari Jurusan Teknik Elektro, Program Studi Elektronika Industri, Politeknik Negeri Jakarta.

**LAMPIRAN 2**

**FOTO ALAT**



Tampilan data hasil pengukuran jarak pada LCD saat pengujian



Wiring alat keseluruhan

## LAMPIRAN 3

### LISTING PROGRAM PENGUKURAN JARAK

```

#include <WiFi.h>
#include <ESPmDNS.h>
#include <WiFiUdp.h>
#include <ArduinoOTA.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#if defined(ESP32)
#include <WiFi.h>
#elif defined(ESP8266)
#include <ESP8266WiFi.h>
#endif
#include <Firebase_ESP_Client.h>
#include "addons/TokenHelper.h"
#include "addons/RTDBHelper.h"
#include "utility.h"

// Set the LCD address to 0x27 for a 16 chars and 2 line display
LiquidCrystal_I2C lcd(0x3F, 20, 4);
FirebaseData fbdo;
FirebaseAuth auth;
FirebaseConfig config;
TaskHandle_t Task1;
TaskHandle_t Task2;

void IRAM_ATTR ai0() {
  if (digitalRead(17) == LOW) {
    counter++;
  } else {
    if (counter > 0) {
      counter--;
    } else {
      counter = 0;
    }
  }
}

void IRAM_ATTR ai1() {
  if (digitalRead(16) == LOW) {
    if (counter > 0) {
      counter--;
    } else {
      counter = 0;
    }
  } else {
    counter++;
  }
}

void setup() {
  Serial.begin(115200);
  Serial.println("Booting");
}

```

```

WiFi.mode(WIFI_STA);
WiFi.begin(ssid, password);
while (WiFi.waitForConnectResult() != WL_CONNECTED) {
  Serial.println("Connection Failed! Rebooting...");
  delay(5000);
  ESP.restart();
}

config.api_key = API_KEY;
auth.user.email = USER_EMAIL;
auth.user.password = USER_PASSWORD;
config.database_url = DATABASE_URL;
config.token_status_callback = tokenStatusCallback;
Firebase.begin(&config, &auth);

Firebase.reconnectWiFi(true);
ArduinoOTA.setPassword("pengukurjarak2021");
ArduinoOTA
.onStart([]() {
  String type;
  if (ArduinoOTA.getCommand() == U_FLASH)
    type = "sketch";
  else // U_SPIFFS
    type = "filesystem";

  // NOTE: if updating SPIFFS this would be the place to unmount
  SPIFFS using SPIFFS.end()
  Serial.println("Start updating " + type);
})
.onEnd([]() {
  Serial.println("\nEnd");
})
.onProgress([](unsigned int progress, unsigned int total) {
  Serial.printf("Progress: %u%%\r", (progress / (total / 100)));
})
.onError([](ota_error_t error) {
  Serial.printf("Error[%u]: ", error);
  if (error == OTA_AUTH_ERROR) Serial.println("Auth Failed");
  else if (error == OTA_BEGIN_ERROR) Serial.println("Begin
Failed");
  else if (error == OTA_CONNECT_ERROR) Serial.println("Connect
Failed");
  else if (error == OTA_RECEIVE_ERROR) Serial.println("Receive
Failed");
  else if (error == OTA_END_ERROR) Serial.println("End Failed");
});

ArduinoOTA.begin();

Serial.println("Ready");
Serial.print("IP address: ");
Serial.println(WiFi.localIP());

pinMode(35, INPUT);

pinMode(16, INPUT_PULLUP);
pinMode(17, INPUT_PULLUP);
//Setting up interrupt
//A rising pulse from encoder activated ai0().

```

```

attachInterrupt(16, ai0, RISING);

//B rising pulse from encodenren activated ail().
attachInterrupt(17, ail, RISING);

xTaskCreatePinnedToCore(
  Task1code, /* Task function. */
  "Task1", /* name of task. */
  10000, /* Stack size of task */
  NULL, /* parameter of the task */
  1, /* priority of the task */
  &Task1, /* Task handle to keep track of created task */
  0); /* pin task to core 0 */
delay(500);

//create a task that will be executed in the Task2code()
function, with priority 1 and executed on core 1
xTaskCreatePinnedToCore(
  Task2code, /* Task function. */
  "Task2", /* name of task. */
  10000, /* Stack size of task */
  NULL, /* parameter of the task */
  1, /* priority of the task */
  &Task2, /* Task handle to keep track of created task */
  1); /* pin task to core 1 */
delay(500);

millis1 = millis();
}

int measureVolts() {
  int sensorValue = analogRead(34); //read the A0 pin value
  float voltage = (sensorValue * 3.3) / (4095); //convert the
value to a true voltage.
  return voltage;
}

void Task1code( void * pvParameters ) {
  // initialize the LCD
  lcd.begin();

  // Turn on the backlight and print a message.
  lcd.backlight();
  lcd.setCursor(4, 0);
  lcd.print("TUGAS AKHIR");
  lcd.setCursor(2, 1);
  lcd.print("PENGUKURAN JARAK");
  lcd.setCursor(6, 2);
  lcd.print("0.00");
  lcd.print(" cm ");

  while (true) {
    int btn_statel = digitalRead(35);
    if (btn_statel == LOW) {
      counter = 0;
      distance = 0;
      lcd.setCursor(6, 2);
      lcd.print(distance);
      lcd.print(" cm ");
    }
  }
}

```

```

}

if ( counter != temp ) {
  // Serial.println (counter);
  temp = counter;
  distance = ((2 * pi * r) / N) * counter ;

  lcd.setCursor(6, 2);

  if (distance < 100) {
    lcd.print(distance);
    lcd.print(" cm ");

    Serial.print(distance);
    Serial.println(" cm ");
  } else {
    lcd.print(distance / 100);
    lcd.print(" m ");

    Serial.print(distance);
    Serial.println(" cm ");
  }
}
}
}
ArduinoOTA.handle();
// vTaskDelay(1000/portTICK_PERIOD_MS);
}
}

//Task2code: blinks an LED every 700 ms
void Task2code( void * pvParameters ) {
  while (true) {
    if (Firebase.ready()) {
      Serial.printf("Send Distance %s\n",
        Firebase.RTDB.setFloat(&fbdo, "/pengukuran/device_1/value",
        distance) ? "ok" : fbdo.errorReason().c_str());
      Serial.printf("Send Volt %s\n",
        Firebase.RTDB.setFloat(&fbdo, "/pengukuran/device_1/voltage",
        measureVolts()) ? "ok" : fbdo.errorReason().c_str());
    }
    vTaskDelay(1000/portTICK_PERIOD_MS);
  }
}

void loop() {

}

```