

**SISTEM KOMUNIKASI NIRKABEL PADA ANTARMUKA  
PERANGKAT PORTABEL MONITORING KETINGGIAN AIR  
BERBASIS MODUL TRANSCEIVER NRF24L01+ PA LNA**

*Wireless Communication System On Portable Water Level  
Monitoring Device Interface Based on NRF24L01+ PA LNA  
Transceiver Module*

**TUGAS AKHIR**

Disusun sebagai syarat kelulusan Pendidikan  
pada program studi strata 1 Teknik Elektro Universitas Sangga Buana  
YPKP

**ADHITYA NAUFAL FIRDAUS**

**2114207004**




**PROGRAM STUDI SARJANA TEKNIK ELEKTRO**

**FAKULTAS TEKNIK**

**UNIVERSITAS SANGGA BUANA YPKP**

**BANDUNG**

**2022**

	<b>UNIVERSITAS SANGGA BUANA</b>	<b>FORMULIR LEMBAR PENGESAHAN</b>	
		No. Revisi	
		Berlaku efektif	

## LEMBAR PENGESAHAN TUGAS AKHIR

### SISTEM KOMUNIKASI NIRKABEL PADA ANTARMUKA PERANGKAT PORTABEL MONITORING KETINGGIAN AIR BERBASIS MODUL TRANSCEIVER NRF24L01+ PA LNA

*Wireless Communication System On Portable Water Level Monitoring  
Device Interface Based on NRF24L01+ PA LNA Transceiver Module*

Disusun oleh :

**ADHITYA NAUFAL FIRDAUS**

**2114207004**

Telah disetujui dan disahkan sebagai Tugas Akhir Program S1 Teknik Elektro  
Fakultas Teknik Universitas Sangga Buana YPKP

Bandung, Agustus 2022

Disahkan oleh :

Pembimbing 1

**Kusmadi, ST., MT.**

**NIK. 432.200.202**

Penguji 1

Penguji 2


**Nina Lestari, ST., MT.**  
**NIK. 432.200.123**

**Ir. Rudy Gunawan, MT.**  
**NIK. 432.200.181**

Mengetahui,

Ketua Prodi Teknik Elektro

**Ketut Abimanyu, ST.,MT**  
**NIK. 432.200.166**

	<b>UNIVERSITAS SANGGA BUANA</b>	<b>FORMULIR LEMBAR PENGESAHAN</b>	
	<b>Jl. PH.H. Mustofa No 68, Cikutra, Cibeunying kidul, Bandung 40124</b>	No. Revisi	
		Berlaku efektif	

## LEMBAR PERNYATAAN ORISINALITAS

Nama : Adhitya Naufal Firdaus  
 NPM : 2114207004  
 Alamat : Jl Jati No. 7 RT 004 / RW 005, Desa Cimanganten, Kecamatan  
 Tarogong Kaler, Kabupaten Garut  
 No. Tlp/HP : +6282240480918  
 E-mail : adhityanaufal18@gmail.com  
 Menyatakan bahwa Tugas Akhir ini merupakan karya orisinal saya sendiri, dengan  
 judul:

**SISTEM KOMUNIKASI NIRKABEL PADA ANTARMUKA PERANGKAT  
 PORTABEL MONITORING KETINGGIAN AIR BERBASIS MODUL  
 TRANSCEIVER NRF24L01+ PA LNA**

*Wireless Communication System On Portable Water Level Monitoring Device  
 Interface Based on NRF24L01+ PA LNA Transceiver Module*

Atas pernyataan ini, saya siap menanggung resiko/sanksi yang dijatuhkan kepada saya apabila di kemudian ditemukan adanya pelanggaran terhadap kejujuran akademik atau etika keilmuan dalam karya ini, atau ditemukan bukti yang menunjukkan ketidakaslian karya ini.

Bandung, Agustus 2022



Adhitya Naufal Firdaus  
2114207004

## ABSTRAK

Pengukuran ketinggian air sangat penting bagi kegiatan operasional sumur panas bumi, khususnya ketika dilakukan kegiatan uji dan manuver sumur produksi. Air yang berasal dari sumur panas bumi ditampung terlebih dahulu di kolam penampungan sebelum dialirkan ke sumur reinjeksi. Ketinggian air pada kolam tersebut harus selalu dijaga agar tidak terjadi luapan air ke lingkungan. Biasanya metode pengukuran ketinggian air di kolam penampungan dilakukan secara manual karena kolam penampungan tidak dilengkapi level transmitter. Metode tersebut sangat tidak efektif dan efisien karena ketinggian air tidak dapat dimonitor secara *real time*. Maka dirancanglah sebuah sistem komunikasi antarmuka nirkabel untuk memonitor ketinggian air menggunakan sensor ultrasonik JSN-SR04T dan modul komunikasi *transceiver* NRF24L01+ PA LNA. Sensor JSN-SR04T merupakan sensor jarak tahan air yang memanfaatkan gelombang ultrasonik. Sementara itu NRF24L01+ PA LNA merupakan modul komunikasi nirkabel jarak jauh yang memanfaatkan frekuensi gelombang radio 2.4-2.5 GHz yang dilengkapi oleh *Low Noise Amplifier* dan *Power Amplifier*. Data ketinggian dari sensor yang ditransmisikan oleh modul komunikasi NRF24L01+ PA LNA akan ditampilkan pada LCD dan platform IOT *thingspeak* serta DCS (*Distributed Control System*) secara nirkabel. Dengan menggunakan metode ini, ketinggian air dapat dimonitor dengan cepat secara *real time* dan efisien. Hasil pengujian sensor dan modul komunikasi menunjukkan bahwa sensor JSN-SR04T mampu mengukur jarak pada rentang 20 cm sampai dengan 500 cm. Sedangkan pada jarak kurang dari 20 cm dan lebih dari 500 cm, sensor tidak akurat dan tidak direkomendasikan untuk mengukur jarak. Sementara itu kemampuan modul NRF24L01+ PA LNA cukup baik dan efektif dalam mentransmisikan data pada jarak maksimum 200 meter.

Kata kunci: sistem komunikasi nirkabel, monitoring ketinggian air, sensor ultrasonik, JSN-SR04T, NRF24L01+ PA LNA

## ABSTRACT

Measurement of water level is very important for geothermal well operations, especially when testing and maneuvering geothermal production wells. Water from geothermal wells is first collected in a cooling water pond before being channeled to reinjection wells. The method is very ineffective and efficient because the water level cannot be monitored in real time and has the potential to cause work accidents. So a wireless interface communication system was designed to monitor water level using the JSN-SR04T ultrasonic sensor and the NRF24L01+ PA LNA transceiver communication module. The JSN-SR04T sensor is a waterproof distance sensor that utilizes ultrasonic waves. Meanwhile, the NRF24L01+ PA LNA is a long-distance wireless communication module that utilizes a 2.4-2.5 GHz radio wave frequency equipped with a Low Noise Amplifier and Power Amplifier. The altitude data from the sensor transmitted by the NRF24L01+ PA LNA communication module will be displayed on the LCD and the thingspeak IoT platform as well as the DCS (Distributed Control System) wirelessly. By using this method, the water level can be monitored quickly in real time and efficiently. The test results of sensors and communication modules show that the JSN-SR04T sensor is able to measure distances in the range of 20 cm to 500 cm. Meanwhile, at distances of less than 20 cm and more than 500 cm, the sensor is not accurate and is not recommended for measuring distances. Meanwhile, the capability of the NRF24L01+ PA LNA module is quite good and effective in transmitting data at a maximum distance of 200 meters.

*Keywords: wireless communication system, water level monitoring, ultrasonic sensor, JSN-SR04T, NRF24L01+ PA LNA*

## **KATA PENGANTAR**

Segala puji hanya bagi Allah Azza Wa Jalla, Tuhan semesta alam yang telah memberikan rahmat dan hidayah sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul “SISTEM KOMUNIKASI NIRKABEL PADA ANTARMUKA PERANGKAT PORTABEL MONITORING KETINGGIAN AIR BERBASIS MODUL TRANSCEIVER NRF24L01+ PA LNA”.

Tugas Akhir ini disusun untuk memenuhi tugas akhir perkuliahan dan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana Strata 1 di Program Studi Teknik Elektro Universitas Sangga Buana. Selain itu, tugas akhir ini juga disusun sebagai salah satu wujud implementasi dari ilmu yang didapatkan selama masa perkuliahan di Program Studi Teknik Elektro Universitas Sangga Buana. Dengan segala kerendahan hati, perkenankan penulis mengucapkan terimakasih kepada seluruh pihak yang terlibat pada penyusunan tugas akhir ini.

Penulis menyadari bahwa penelitian ini masih jauh dari kesempurnaan. Oleh karena itu, penulis berharap kritik dan saran yang membangun sehingga penulis dapat meningkatkan kompetensi dalam melakukan penelitian. Penulis berharap penelitian ini dapat memberikan manfaat bagi penulis pada khususnya dan pembaca pada umumnya. Semoga Allah Azza Wa Jalla selalu memberikan hidayah kepada kita semua.

Bandung, Agustus 2022

Penulis

## DAFTAR ISI

LEMBAR PENGESAHAN.....	i
LEMBAR PERNYATAAN ORISINALITAS .....	iii
ABSTRAK .....	iv
ABSTRACT .....	v
KATA PENGANTAR .....	vi
DAFTAR ISI.....	vii
DAFTAR GAMBAR .....	ix
DAFTAR TABEL.....	xi
DAFTAR ISTILAH .....	xii
DAFTAR LAMPIRAN.....	xiv
BAB I PENDAHULUAN .....	1
1.1. Latar Belakang Masalah .....	1
1.2. Pengembangan Penelitian Terdahulu .....	4
1.3. Rumusan Masalah.....	5
1.4. Maksud dan Tujuan .....	5
1.5. Ruang Lingkup .....	5
1.6. Metode Penelitian .....	6
1.7. Sistematika Penulisan .....	7
BAB II LANDASAN TEORI .....	8
2.1 Sistem Komunikasi Nirkabel.....	8
2.2 Modul <i>Transceiver</i> NRF24L01+ PA LNA.....	9
2.3 Sensor Ultrasonik JSN-SR04T .....	11
2.4 Arduino Nano .....	12
2.5 NodeMCU ESP8266.....	13
2.6 <i>Thingspeak</i> .....	14
2.7 Cooling Water <i>Pond</i> Lapangan Panas Bumi Ulubelu .....	15
BAB III DESAIN DAN METODE .....	16
3.1 Diagram Blok Sistem Komunikasi .....	16
3.2 Desain Alat .....	16
3.2.1 Desain Mekanik .....	16
3.2.2 Desain Program .....	17



3.2.3	Desain Rangkaian Modul <i>Transmitter</i> .....	18
3.2.4	Desain Rangkaian Modul <i>Receiver</i> .....	19
3.3	Metode Pengambilan Data.....	20
3.3.1.	Pengambilan Data Sensor Ultrasonik JSN-SR04T.....	20
3.3.2.	Pengambilan Data Jarak Komunikasi Modul NRF24L01+ PA LNA .....	20
3.3.3.	Waktu dan Tempat Penelitian.....	20
3.3.4.	Alat dan Bahan .....	21
3.4.	Diagram Alir atau <i>Flowchart</i> .....	22
BAB IV PENGUJIAN DAN HASIL ANALISA .....		23
4.1	Implementasi Sistem Komunikasi .....	23
4.2	Pengujian Sensor Ultrasonik JSN-SR04T .....	26
4.2.1.	Hasil Pengujian Karakteristik Sensor Ultrasonik JSN-SR04T.....	28
4.2.2.	Analisa Hasil Pengujian Karakteristik Sensor Ultrasonik JSN-SR04T.....	28
4.2.3.	Hasil Pengujian Akurasi Sensor Ultrasonik JSN-SR04T .....	29
4.2.4.	Analisa Hasil Pengujian Akurasi Sensor Ultrasonik JSN-SR04T.....	33
4.3	Pengujian Jangkauan Komunikasi Modul NRF24L01+ PA LNA .....	34
4.3.1.	Pengambilan Data Modul Komunikasi NRF24L01+ PA LNA.....	35
4.3.2.	Analisa Hasil Pengambilan Data Modul Komunikasi .....	36
4.4	Integrasi Sistem sebagai Solusi Masalah Penelitian.....	38
BAB V PENUTUP.....		40
5.1	Simpulan.....	40
5.2	Saran .....	40
DAFTAR PUSTAKA .....		42
LAMPIRAN A Datasheet Modul NRF24L01+ PA LNA.....		44
LAMPIRAN B Datasheet Sensor Ultrasonik JSN-SR04T .....		49
LAMPIRAN C Datasheet Arduino Nano .....		53
LAMPIRAN D Datasheet NodeMCU ESP8266 .....		57
LAMPIRAN E Desain Mekanik .....		61
LAMPIRAN F Desain Program.....		63
LAMPIRAN G Desain Diagram Skematik.....		70
LAMPIRAN H Perbandingan Lisensi <i>Thingspeak</i> .....		74

## DAFTAR GAMBAR

Gambar 2. 1	Arsitektur Point to Point dan Point to Multipoint [6].....	8
Gambar 2. 2	Diagram Blok Dasar Sistem Komunikasi Nirkabel[6].....	8
Gambar 2. 3	Modul NRF24L01+ PA LNA .....	10
Gambar 2. 4	Cara kerja modul NRF24L01+ PAL01 LNA[8] .....	10
Gambar 2. 5	Sensor Ultrasonik JSN-SR04T [4] .....	11
Gambar 2. 6	Mikrokontroler Arduino Nano[11].....	13
Gambar 2. 7	NodeMCU ESP8266 [5] .....	14
Gambar 2. 8	Cooling pond lapangan panas bumi area Ulubelu.....	15
Gambar 3. 1	Diagram blok sistem komunikasi.....	16
Gambar 3. 2	Desain mekanik cover modul <i>transmitter</i> .....	17
Gambar 3. 3	Desain mekanik cover modul <i>receiver</i> .....	17
Gambar 3. 4	Desain program modul transmitter.....	17
Gambar 3. 5	Desain program modul transmitter.....	18
Gambar 3. 6	Diagram satu garis modul <i>transmitter</i> .....	19
Gambar 3. 7	Diagram rangkaian modul <i>transmitter</i> .....	19
Gambar 3. 8	Diagram satu garis modul <i>receiver</i> .....	19
Gambar 3. 9	Diagram rangkaian modul <i>receiver</i> .....	19
Gambar 3. 10	Diagram alir transmitter .....	22
Gambar 3. 11	Diagram alir receiver.....	22
Gambar 4. 1	Hasil implementasi modul transmitter .....	24
Gambar 4. 2	Hasil implementasi modul receiver.....	25
Gambar 4. 3	Implementasi tampilan antarmuka LCD .....	25
Gambar 4. 4	Implementasi tampilan antarmuka <i>thingspeak</i> .....	25
Gambar 4. 5	Pengambilan data sensor ultrasonik JSN-SR04T.....	26
Gambar 4. 6	Tampilan data jarak pada LCD modul receiver .....	26
Gambar 4. 7	Pengujian akurasi dan <i>error</i> sensor JSN-SR04T.....	27
Gambar 4. 8	Grafik karakteristik sensor ultrasonik JSN-SR04T.....	29
Gambar 4. 9	Grafik akurasi sensor ultrasonik JSN-SR04T .....	33
Gambar 4. 10	Penempatan modul transmitter.....	34

Gambar 4. 11 Penempatan modul receiver .....	34
Gambar 4. 12 Titik pengambilan data NRF24L01+ PA LNA .....	36
Gambar 4. 13 Grafik pengujian tingkat kesuksesan konektivitas modul transceiver berdasarkan jarak jangkauan .....	37

## DAFTAR TABEL

Tabel 2. 1 Spesifikasi modul NRF24L01+ PA LNA[6] .....	11
Tabel 2. 2 Spesifikasi sensor ultrasonik JSN-SR04T[7] .....	12
Tabel 2. 3 Spesifikasi Arduino Nano[10] .....	13
Tabel 2. 4 Spesifikasi NodeMCU ESP8266 [12].....	14
Tabel 3. 1 Alat dan bahan .....	21
Tabel 4. 1 Hasil Pengujian Sensor Ultrasonik JSN-SR04T .....	30
Tabel 4. 2 Hasil Pengujian Modul Komunikasi NRF24L01+ PA LNA+ pada kondisi halangan atau ( <i>Line of Sight</i> ).....	35

## DAFTAR ISTILAH

Antarmuka	: Gabungan dari elemen – elemen dari suatu sistem, pengguna dan komunikasi dan interaksi keduanya
Arduino Nano	: <i>Board development</i> mikrokontroler berukuran kecil berbasis chip ATmega328P
<i>Brine</i>	: Air hasil separasi energi panas bumi yang memiliki berbagai kandungan mineral.
<i>Cooling Water Pond</i>	: Kolam penampungan di lapangan panas bumi yang berfungsi menampung <i>brine</i> atau air yang berasal dari sumur produksi panas bumi.
<i>Geothermal</i>	: Energi panas yang terdapat dan terbentuk di dalam kerak bumi
<i>Internet of things</i>	: Sebuah konsep dimana objek mampu mengirimkan data menggunakan jaringan internet
JSN-SR04T	: Jenis sensor ultrasonik tahan air yang dilengkapi kabel sepanjang 2,5 meter
<i>Line of Sight</i>	: Suatu kondisi dimana antara pengirim (Tx) dengan penerima (Rx) dapat saling melihat tanpa ada penghalang
Nirkabel	: Transfer informasi antara dua atau lebih titik yang tidak terhubung oleh penghantar listrik
NodeMCU ESP8266	: <i>Microcontroller</i> yang sudah dilengkapi dengan module WIFI
NRF24L01+ PA LNA	: Modul komunikasi nirkabel jarak jauh yang memanfaatkan frekuensi gelombang radio 2.4-2.5 GHz <i>Industrial Scientific and Medical (ISM)</i> dan dilengkapi <i>Power Amplifier</i> dan <i>Low Noise Amplifier</i>

- Thingspeak* : Sebuah website open source berbasis internet of things (IOT) yang memberikan layanan untuk kebutuhan IOT dan dapat menerima informasi menggunakan protokol HTTP melalui jaringan internet
- Ultrasonik : Suara atau getaran dengan frekuensi yang mempunyai frekuensi sangat tinggi yaitu 20 kHz

## **DAFTAR LAMPIRAN**

### **LAMPIRAN A**

Datasheet Modul NRF24L01+ PA LNA

### **LAMPIRAN B**

Datasheet Sensor Ultrasonik JSN-SR04T

### **LAMPIRAN C**

Datasheet Arduino Nano

### **LAMPIRAN D**

Datasheet NodeMCU ESP8266

### **LAMPIRAN E**

Desain Mekanik

### **LAMPIRAN F**

Desain Program

### **LAMPIRAN G**

Desain Diagram Skematik

### **LAMPIRAN H**

Perbandingan Lisensi *Thingspeak*

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang Masalah

Seiring perkembangan zaman, semakin banyak inovasi teknologi yang dikembangkan untuk mempermudah pekerjaan manusia. Sehingga pekerjaan yang dilakukan manusia diharapkan menjadi lebih efektif dan efisien. Salah satunya yang berkembang sangat pesat adalah sistem komunikasi nirkabel. Sistem komunikasi nirkabel telah merambah ke berbagai aspek kehidupan manusia dari bidang industri hingga rumah tangga. Komunikasi nirkabel adalah komunikasi yang menggunakan frekuensi/spektrum radio yang memungkinkan transmisi (penerimaan atau pengiriman) informasi (suara, gambar, video, data) tanpa koneksi fisik yang bersifat *mobile* (GSM, CDM, Flexi, 3G) dan bersifat tetap (*wireless local loop, bluetooth, WiFi, WiMax*) [1].

Interaksi antara pengguna dan komputer dapat terjadi jika tersedia media interaksi. Media interaksi ini diwujudkan dalam bentuk antarmuka. Antarmuka dapat diartikan gabungan dari elemen – elemen dari suatu sistem, pengguna dan komunikasi dan interaksi keduanya [2].

Panas bumi (*Geothermal*) adalah sebuah sumber energi panas yang terdapat dan terbentuk di dalam kerak bumi. Panas bumi adalah sumber energi panas yang terkandung di dalam air panas, uap air dan batuan bersama mineral dan gas yang secara genetis semuanya tidak dapat dipisahkan dalam suatu sistem panas bumi dan untuk pemanfaatannya diperlukan proses penambangan. Energi panas bumi dapat digunakan sebagai pengganti tenaga listrik yang menggunakan bahan bakar minyak



sehingga dapat dijadikan sumber energi alternatif untuk menghemat cadangan minyak nasional[3].

Lapangan kerja pembangkit listrik tenaga panas bumi (*Geothermal*) terbagi menjadi dari dua jenis area yaitu, dominan fasa uap dan dominan fasa air. Area panas bumi Ulubelu, di Provinsi Lampung merupakan area yang memiliki tipe dominan fasa air dibandingkan dengan fasa uap. Sehingga setiap kluster sumur panas bumi wajib memiliki *cooling water pond* atau kolam penampung air ketika dilakukan uji sumur produksi. *Cooling water pond* di area Ulubelu memiliki kedalaman 500 cm.

Sumur panas bumi di Area Ulubelu memiliki kandungan air yang lebih besar dibandingkan dengan uap dengan presentase 80:20. Oleh karena itu, ketinggian *cooling water pond* kluster sangat vital dalam kegiatan operasional sumur panas bumi. Khususnya ketika proses uji sumur produksi panas bumi dan manuver sumur produksi.

Namun beberapa *cooling water pond* atau kolam penampung air pada kluster sumur produksi tidak dilengkapi dengan *Level Transmitter* sebagai alat monitoring ketinggian air. Pengadaan *level transmitter* memerlukan biaya yang cukup tinggi karena harga alat yang mahal dan membutuhkan koneksi kabel yang cukup panjang. Selain itu diperlukan perawatan yang massif karena *probe level transmitter* sering terkontaminasi oleh silika yang terkandung pada air *brine* panas bumi. Kontaminasi silika pada probe dapat mengakibatkan pengukuran tidak akurat dan kerusakan pada transmitter. Sehingga pengukuran ketinggian air dilakukan secara manual oleh operator. Metode tersebut tidak efisien karena ketinggian air

tidak dapat dimonitor secara *real time* dan diperlukan mobilisasi operator dengan waktu tempuh lebih dari 15 menit untuk sampai ke lokasi.

Maka dari itu, diperlukan sebuah metode untuk memonitor ketinggian air secara *real time* yang akurat dan handal. Salah satu metode pengukuran ketinggian air yang dapat digunakan adalah pemanfaatan sensor ultrasonik. Karena lokasi pengukuran berada di lingkungan terbuka, maka diperlukan sensor yang handal pada saat kondisi hujan. Salah satu jenis sensor ultrasonik yang dapat tahan terhadap air (*waterproof*) adalah sensor JSN-SR04T. Sensor JSN-SR04T merupakan sensor ultrasonik tahan air yang dilengkapi kabel sepanjang 2,5 meter yang memanfaatkan gelombang ultrasonik[4].

Kemudian agar ketinggian air dapat dimonitor secara *real time* dari jarak jauh, diperlukan sebuah sistem komunikasi yang mampu mentransmisikan data secara nirkabel. Modul NRF24L01+ PA LNA merupakan modul komunikasi nirkabel jarak jauh yang memanfaatkan frekuensi gelombang radio 2.4-2.5 GHz *Industrial Scientific and Medical* (ISM) dengan kecepatan pengiriman data mencapai 2 Mbps[5].

Pada penelitian ini dirancang sebuah sistem komunikasi antarmuka nirkabel untuk dapat memonitor ketinggian air menggunakan sensor ultrasonik JSN-SR04T dan modul komunikasi *transceiver* NRF24L01+ PA LNA. Data ketinggian tersebut kemudian ditampilkan pada LCD dan platform web *thingspeak* secara nirkabel. Dengan menggunakan metode ini, ketinggian air dapat dimonitor dengan cepat secara *real time* dan efisien.

## 1.2. Pengembangan Penelitian Terdahulu

Berikut beberapa penelitian yang dijadikan referensi oleh peneliti:

Tabel 1. 1 Jurnal Penelitian Terdahulu

No	Judul, Author, Tahun	Tahun	Hasil penelitian	Pembaharuan
1.	Monitoring Tinggi Permukaan Air Pada Tandon Menggunakan Sensor Ultrasonik SR04 Berbasis Arduino Uno Dengan Komunikasi Wireless NRF24L01, Achmad Sulthoni, 2017	2017	Hasil dari percobaan yang telah dilakukan ini adalah cukup baik dan tidak jauh berbeda dengan jarak yang terbaca oleh mistar. Sensor bekerja baik pada range suhu dalam tandon 26°C sampai 28°C.	<ul style="list-style-type: none"> <li>- Mengganti sensor ultrasonik dengan JSN-SR04T.</li> <li>- Penambahan <i>Power Amplifier</i> dan <i>Low Noise Amplifier</i>.</li> <li>- Mengganti Arduino Uno dengan modul wifi NodeMCU ESP8266.</li> </ul>
2	Implementasi Sistem Pendeteksi Ketinggian Air Dengan Menggunakan Wireless Sensor Network Node Point To Point	2018	Tingkat akurasi sensor ultrasonik lebih dari 89%, ketepatan pembacaan data 80% dan akurasi pengiriman data 100%	<ul style="list-style-type: none"> <li>- Mengganti sensor ultrasonik dengan JSN-SR04T.</li> <li>- Penambahan <i>Power Amplifier</i> dan <i>Low Noise Amplifier</i>.</li> <li>- Mengganti Arduino Uno dengan modul wifi NodeMCU ESP8266.</li> </ul>
3.	Rancang Bangun Sistem Telemetry Pengukuran Ketinggian Gelombang Pasang Surut Air Laut secara Realtime Menggunakan Arduino Uno	2018	Sistem berfungsi dapat bekerja dengan baik dalam melakukan pengukuran, pengiriman dan penerimaan data hasil pengukuran ketinggian gelombang	<ul style="list-style-type: none"> <li>- Mengganti sensor ultrasonik dengan JSN-SR04T.</li> <li>- Penambahan <i>Power Amplifier</i> dan <i>Low Noise Amplifier</i>.</li> <li>- Mengganti Arduino Uno dengan modul wifi NodeMCU ESP8266.</li> </ul>

4.	Sistem Pemonitor Tinggi Air Bendungan Menggunakan Modul Wireless	2019	Komunikasi antara alat pemonitor ketinggian air dengan modem mampu mengirimkan data sensor hingga jarak 95 meter pada area terbuka.	- Penambahan <i>Power Amplifier</i> dan <i>Low Noise Amplifier</i> .
----	--	------	---	--

### 1.3. Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, maka dirumuskan masalah sebagai berikut :

1. Bagaimana cara monitoring ketinggian air secara *realtime*?
2. Bagaimana metode pengukuran ketinggian yang akurat dan handal?
3. Bagaimana cara monitoring ketinggian air dari jarak jauh tanpa menggunakan kabel?

### 1.4. Maksud dan Tujuan

Adapun maksud dan tujuan dari penelitian ini adalah untuk merancang suatu sistem komunikasi antarmuka pada perangkat monitoring ketinggian air *portabel* secara nirkabel menggunakan modul *transceiver* NRF24L01+ PA LNA yang mampu mengirimkan data ketinggian ke dalam tampilan *realtime*.

### 1.5. Ruang Lingkup

Adapun ruang lingkup penelitian ini adalah sebagai berikut:

1. Modul dan perangkat yang digunakan pada penelitian ini adalah menggunakan jenis modul *transceiver* NRF24L01+ PA LNA sebagai media komunikasi.
2. Sensor Ultrasonik JSN-SRT04 sebagai sensor pengukur ketinggian.
3. Pengujian fungsi alat secara parsial dilakukan di *workshop*.

## 1.6. Metode Penelitian

Metode yang dilakukan pada penelitian ini dijabarkan ke dalam beberapa tahapan penelitian, yaitu:

### 1. Perencanaan

Pada tahapan ini ditentukan tujuan penulisan, tema dan sasaran penelitian yang direncanakan dengan baik. Pemilihan instrumen sensor dan modul komunikasi juga dipertimbangkan agar sasaran penelitian tercapai.

### 2. Uji Coba

Tahapan uji coba dipilih setelah tahapan perencanaan dilaksanakan dengan baik. Uji coba dilakukan pada modul komunikasi *transceiver* NRF24L01+ PA LNA. Sehingga dari uji coba ini, penulis mampu mengidentifikasi jarak tempuh yang mampu dijangkau oleh modul komunikasi ini.

### 3. Desain

Dalam proses rancang bangun alat maka tahapan desain merupakan hal yang sangat penting. Desain sistem komunikasi yang akan penulis buat dimulai dari pemilihan jenis sensor dan modul komunikasi, *wiring diagram* rangkaian sistem komunikasi pada perangkat monitoring ketinggian air menggunakan modul *transceiver* NRF24L01+ PA LNA, pembuatan perangkat lunak (*coding*) yang dilakukan menggunakan Arduino IDE dan bentuk pembacaan hasil pengukuran pada tampilan antarmuka.

### 4. Implementasi

Implementasi dilakukan untuk menguji efektifitas dan fungsi dari alat yang dibuat. Pengujian dilakukan pada sebuah aquarium di workshop yang merepresentasikan *cooling pond* PT Pertamina Geothermal Energy Area Ulubelu, Lampung.

## 1.7. Sistematika Penulisan

Sistematika penulisan yang digunakan dalam penulisan hasil penelitian ini ialah sebagai berikut:

### BAB I Pendahuluan

Berisi latar belakang masalah, pembaharuan penelitian, rumusan masalah, maksud dan tujuan penelitian, metode penelitian dan sistematika penulisan.

### BAB II Landasan Teori

Berisi landasan teori dasar dan pendukung serta pengenalan terhadap penghubung seluruh kegiatan penelitian baik perangkat keras maupun perangkat lunaknya.

### BAB III Desain dan Metode

Berisi rancangan desain dan tahapan implementasi penelitian sistem komunikasi antarmuka nirkabel menggunakan sensor ultrasonik JSN-SR04T dan modul transceiver NRF24L01+ PA LNA.

### BAB IV Pengujian dan Analisa Hasil Implementasi

Pada bab ini dibahas mengenai hasil perancangan dari sistem komunikasi antarmuka nirkabel pada perangkat monitoring ketinggian air menggunakan modul *transceiver* NRF24L01+ PA LNA+PA+LNA serta hasil pengujian dan analisisnya.

### BAB V Penutup

Bab ini berisi simpulan dari penelitian yang telah dilakukan serta saran untuk penelitian yang bisa dikembangkan dari penelitian ini.

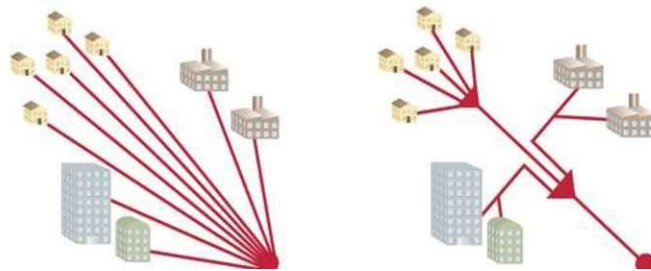
## BAB II

### LANDASAN TEORI

#### 2.1 Sistem Komunikasi Nirkabel

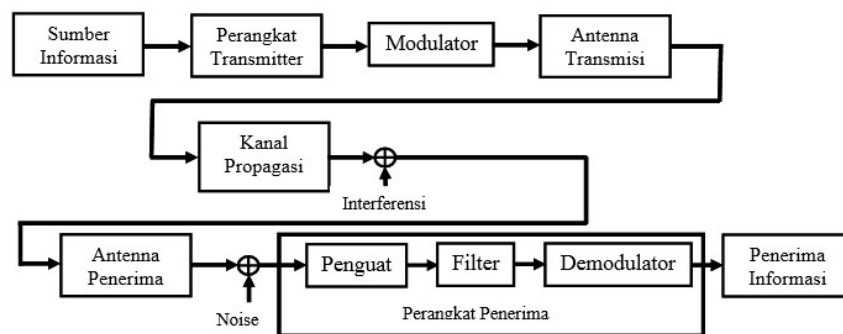
Sistem komunikasi nirkabel adalah sistem komunikasi yang disalurkan tanpa memakai kabel. Sistem komunikasi nirkabel berkembang pesat seiring dengan kebutuhan pelanggan. Sistem komunikasi ini memiliki dua arsitektur yaitu *Point to Point (PTP)* dan *Point to Multipoint (PTMP)* [6].

*Point to Point (PTP)* sering disebut juga dengan *leased lines*, yaitu merupakan komunikasi antara dua lokasi yang berbeda. Misalnya dianalogikan seperti komunikasi antara sebuah kantor cabang dengan kantor perwakilan dalam satu wilayah seperti terlihat pada Gambar 2. 1 di bawah:



Gambar 2. 1 Arsitektur Point to Point dan Point to Multipoint [6]

Sistem komunikasi nirkabel digambarkan melalui diagram blok dasar yang ditunjukkan oleh Gambar 2. 2 berikut:



Gambar 2. 2 Diagram Blok Dasar Sistem Komunikasi Nirkabel[6]

Komunikasi nirkabel merupakan suatu bentuk pengiriman informasi antara dua titik atau lebih yang tidak terhubung oleh konduktor listrik atau kabel. Penyambungan antar titik membutuhkan kanal radio sebagai media penyalur sinyal informasi yang dikirimkan. Selama melewati kanal radio, sinyal melalui berbagai macam halangan pada jalur propagasi. Hal ini mengakibatkan terjadinya pelemahan sinyal pada penerima[7].

Kanal nirkabel merupakan faktor pokok yang membatasi kinerja sistem komunikasi nirkabel. Jarak transmisi antara pemancar (*transmitter*) dan penerima (*receiver*) dapat bervariasi dari *line of sight* (LOS) sampai tertutup oleh penghalang seperti bangunan, bukit dan pepohonan (NLOS). Berbeda dengan kanal yang menggunakan kabel yang dapat diprediksi, kanal radio bersifat random dan tidak mampu dianalisis dengan mudah[7].

## **2.2 Modul Transceiver NRF24L01+ PA LNA**

Modul NRF24L01 adalah modul komunikasi nirkabel jarak jauh yang memanfaatkan frekuensi gelombang radio 2.4-2.5 GHz *Industrial Scientific and Medical* (ISM). Kecepatan pengiriman data modul ini cukup tinggi hingga mencapai 2 Mbps. Daya keluaran, saluran frekuensi, dan pengaturan protokol diprogram melalui antarmuka *Serial Parallel Interface* (SPI) untuk berkomunikasi dengan mikrokontroler. Konsumsi arus yang digunakan modul ini sangat rendah, yaitu hanya 9.0 mA pada daya output -6 dBm dan 12.3 mA pada mode penerimaan (*receiver*) [5].

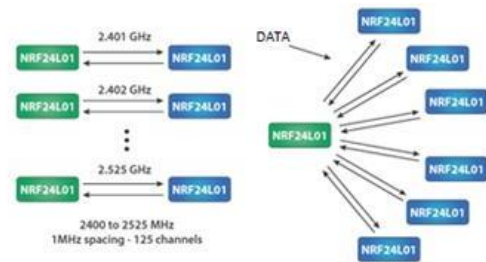
Terdapat beberapa modul yang menggunakan chip Selain itu, modul NRF24L01 juga memiliki fitur *Ultra Low Power* (ULP) yang memungkinkan daya



tahan baterai berbulan-bulan hingga bertahun-tahun[8]. Bentuk dan cara kerja dari modul NRF24L01+ PA LNA ditunjukkan pada gambar 2. 3 dan 2. 4 berikut.



Gambar 2. 3 Modul NRF24L01+ PA LNA



Gambar 2. 4 Cara kerja modul NRF24L01+ PAL01 LNA[8]

Modul transceiver NRF24L01 ini kompatibel terhadap 125 saluran yang berbeda serta mampu menghubungkan 125 jaringan pada satu area. Modul komunikasi ini mampu melakukan komunikasi dengan enam modul komunikasi lain dalam waktu bersamaan[8].

Adapun pin yang terdapat pada modul NRF24L01 yaitu VCC, GND, CSN, CE, MOSI, MISO dan IRQ. Pin VCC berfungsi sebagai masukan tegangan sebesar 3,3 V. Pin GND untuk menghubungkan modul ke. Pin CE (*Chip Enable*) untuk mengaktifkan komunikasi SPI (*Serial Peripheral Interface*). Pin CSN (*Chip Select Not*) untuk menyalakan masukan high atau mematikan SPI saat masukan selain high. Pin SCK (*Serial Clock*) untuk memasukkan *clock* pada komunikasi SPI[8]. Adapun spesifikasi modul NRF24L01+ PA LNA dapat dilihat pada tabel 2. 1 berikut.

Tabel 2. 1 Spesifikasi modul NRF24L01+ PA LNA[9]

No	Parameter	Spesifikasi
1	Maximum output power	+20 dBm
2	Emission mode current (peak)	115 mA
3	Receive Mode Current (peak)	45 mA
4	Power-down mode current	4.2 $\mu$ A
5	Sensitivity 2Mbps mode in received	-92 dBm
6	Sensitivity 1Mbps mode in received	-95 dBm
7	Sensitivity 6000kbps mode in received	-104 dBm
8	PA gain	20 dB
9	LNA gain	10 dB
10	LNA Noise figure	2.6 dB
11	Antenna Gain (peak)	2 dBI

### 2.3 Sensor Ultrasonik JSN-SR04T

Sensor JSN-SR04T merupakan sensor ultrasonik tahan air yang dilengkapi kabel sepanjang 2,5 meter serta terhubung ke papan breakout sebagai pengontrol sensor dan pemroses sinyal. Sensor ultrasonik bekerja dengan memancarkan gelombang ultrasonik. Gelombang ultrasonik dipantulkan dari objek yang diarahkan kemudian dideteksi oleh sensor ultrasonik[4]. Bentuk fisik dari sensor JSNSR04T ditunjukkan pada gambar 2. 5 berikut.



Gambar 2. 5 Sensor Ultrasonik JSN-SR04T [4]

Untuk menghitung jarak antara sensor dan objek yang diarahkan dapat dilakukan dengan menghitung waktu yang telah berlalu antara transmisi dan penerimaan gelombang suara seperti pada persamaan berikut[4].

$$\text{Jarak (s)} = \text{Kecepatan suara (cm / s)} \times \text{waktu (\mu s)} \quad (1)$$

Dimana diketahui bahwa kecepatan suara (cm/s) adalah 0.343 cm/s sedangkan waktu ( $\mu$ s) adalah waktu antara transmisi dan penerimaan gelombang suara dalam mikrodetik. Berbeda dengan sensor ultrasonik HC-SR04, sensor ultrasonik JSN-SR04T ini hanya menggunakan satu transduser ultrasonik. Transduser ini bertindak sebagai pemancar dan penerima gelombang ultrasonik. [4].

Adapun spesifikasi sensor ultrasonik JSN-SR04T dapat dilihat pada tabel 2. 2 berikut.

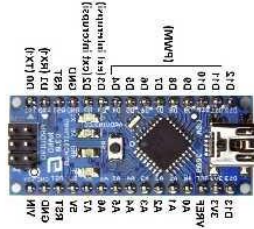
Tabel 2. 2 Spesifikasi sensor ultrasonik JSN-SR04T[4]

No	Parameter	Spesifikasi
1	Operating Voltage	DC 3.0 – 5.5 V
2	Working Current	Less than 8 mA
3	Probe Frequency	40 kHz
4	Farthest Range	600 cm
5	Recent Range	20 cm
6	Distance Accuracy	+ - 1 cm
7	Resolution	1 mm
8	Measuring Angle	75 degree
9	Enter the trigger signal	1, 10 $\mu$ s above the TTL pulse 2, the serial port to send instructions 0X55
10	Output the echo signal	Output pulse width level signal, or TTL
11	Wiring	3– 5.5 V (power positive) Trig (RX) RX Echo (Output) TX GND (power supply) negative
12	Product Size	L42*W29*H12 mm
13	Operating temperature	-20°C to +70°C
15	Product color	PCB board is blue

## 2.4 Arduino Nano

Arduino Nano merupakan *board development* mikrokontroler berukuran kecil berbasis chip ATmega328P. Secara fungsi, Arduino nano hampir sama dengan Arduino Uno. Yang membedakannya adalah Arduino nano menggunakan konektor Mini-B USB dan tidak memiliki *jack power* serta memiliki dua pin analog

yang lebih banyak dibandingkan dengan Arduino Uno[10]. Bentuk fisik dari Arduino nano ditunjukkan seperti pada gambar 2. 6 berikut.



Gambar 2. 6 Mikrokontroler Arduino Nano[11]

Adapun Spesifikasi Arduino Nano dapat dilihat pada tabel 2. 3 berikut:

Tabel 2. 3 Spesifikasi Arduino Nano[12]

No	Parameter	Spesifikasi
1	Mikrokontroler	ATmega328
2	Tegangan Operasi	5 V
3	Tegangan Input	7-12 V
4	Batas Tegangan Input	6-20 V
5	Pin I/O Digital	14 (6 dipakai untuk output PWM)
6	Pin Input Analog	8
7	Arus DC per pin I/O	40 mA
8	Flash Memory	16 kB (ATmega 168) 32 kB (ATmega328)
9	SRAM	1 kB (ATmega 168) 2 kB (ATmega328)
10	EEPROM	512 Bytes (ATmega 168) 1 kB (ATmega328)
11	Kecepatan Clock	16 MHz
12	Dimensi	0.73 cm × 1.70 cm
13	Panjang	45 mm
14	Lebar	18 mm
15	Berat	5 g

## 2.5 NodeMCU ESP8266

Modul ESP8266 adalah modul WiFi berbiaya rendah yang didukung penuh. Melalui TCP/IP atau UDP. ESP8266 dikembangkan oleh pengembang Cina, "Espreffif". Modul WiFi ini merupakan *System on Chip* (SOC), sehingga dapat langsung diprogram dengan ESP8266 tanpa memerlukan mikrokontroler

tambahan[5]. Bentuk fisik NodeMCU ESP8266 ditunjukkan pada gambar 2. 7 berikut.



Gambar 2. 7 NodeMCU ESP8266 [5]

Adapun spesifikasi NodeMCU ESP8266 dapat dilihat pada tabel 2. 4 berikut.

Tabel 2. 4 Spesifikasi NodeMCU ESP8266 [13]

No	Parameter	Spesifikasi
1	Operating Voltage	2.5 – 3.6 VDC
2	Input Voltage	5 VDC
3	GPIO	13
4	ADC	1 Pin (10 Bit)
5	Wi-fi Protocol	802.11 b/g/n/e/i

## 2.6 Thingspeak

*Thingspeak* merupakan sebuah *website open source* berbasis *internet of things* (IOT) yang memberikan layanan untuk kebutuhan IOT dan dapat menerima informasi menggunakan protokol HTTP melalui jaringan internet[14]. Platform ThingSpeak mampu mengumpulkan, menyimpan, menganalisa, membuat visualisasi, serta melakukan aksi atas informasi yang dikirim dari sensor seperti Arduino, Raspberry Pi, BeagleBone Hitam, dan perangkat keras lainnya[15]. *Thingspeak* memiliki beberapa lisensi yaitu *free*, *student*, *home*, *academic* dan *standard*. Adapun perbandingan fitur masing-masing lisensinya dapat dilihat pada lampiran H.

## 2.7 Cooling Water Pond Lapangan Panas Bumi Ulubelu

*Cooling water pond* adalah kolam penampungan di lapangan panas bumi yang berfungsi menampung *brine* atau air yang berasal dari sumur produksi ketika proses uji sumur ataupun manuver sumur produksi panas bumi. Selain itu *cooling pond* juga berfungsi untuk memaksimalkan proses pendinginan *brine* sebelum dialirkan ke sumur reinjeksi menggunakan pompa [16].

Area Geothermal Ulubelu merupakan salah satu wilayah kerja Pertamina Geothermal Energy yang berada di daerah Gunung Tanggamus, Provinsi Lampung. Area Geothermal Ulubelu memiliki sumur-sumur produksi yang memiliki sifat dua fasa dengan dominasi *brine* (80%) dan *steam* (20%) sehingga setiap cluster rata-rata memiliki dua sampai empat *cooling pond*. *Cooling water pond* di Area Ulubelu memiliki kedalaman 500 cm. Tampilan dari *cooling pond* di lapangan panas bumi ditunjukkan pada gambar 2. 8 berikut[16].



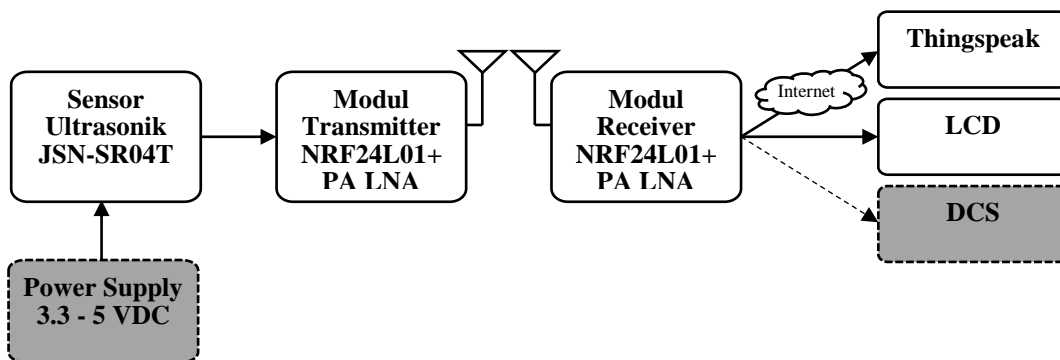
Gambar 2. 8 Cooling pond lapangan panas bumi area Ulubelu

## BAB III

### DESAIN DAN METODE

#### 3.1 Diagram Blok Sistem Komunikasi

Pada bagian desain dan implementasi ini ditampilkan sebuah diagram blok yang menggambarkan alur konsep alat keseluruhan dari awal sampai akhir untuk memudahkan penelitian ini. Ruang lingkup penelitian ini dimulai dari sensor ultrasonik sampai dengan tampilan LCD dan *Thingspeak*. Sedangkan *power supply* dan sistem DCS merupakan bagian yang terpisah dari penelitian ini. Prosesnya dijabarkan pada gambar 3. 1 berikut.



Gambar 3. 1 Diagram blok sistem komunikasi

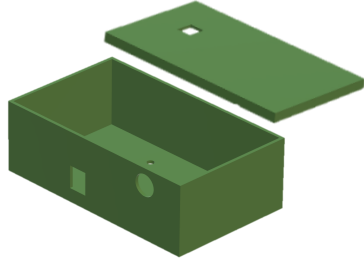
#### 3.2 Desain Alat

Tahapan desain dan perancangan sangat penting dalam penelitian ini. Pada penelitian ini perancangan dibagi menjadi perancangan mekanik, perancangan program, perancangan modul *transmitter* dan perancangan modul *receiver*.

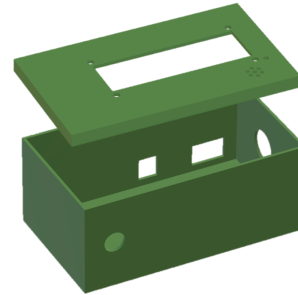
##### 3.2.1 Desain Mekanik

Setelah sistem dapat berfungsi dengan baik, diperlukan desain mekanik untuk menjaga kehandalan sensor dan perangkat lainnya. Desain mekanik yang dilakukan yaitu pembuatan *cover* modul *transmitter* dan *cover* modul *receiver*.

Desain mekanik *cover* modul *transmitter* dan modul *receiver* dapat dilihat pada gambar 3. 2 dan 3. 3 berikut.



Gambar 3. 2 Desain mekanik cover modul *transmitter*



Gambar 3. 3 Desain mekanik cover modul *receiver*

### 3.2.2 Desain Program

Desain program atau *coding* dilakukan agar perangkat dapat beroperasi sesuai konsepnya. Program dirancang menggunakan Arduino IDE meliputi sisi *transmitter* atau sensor dan sisi *receiver* atau *user interface*. Adapun coding dapat dilihat pada gambar 3. 4 berikut.

```
// Define Trig and Echo pin:
#define trigPin 2
#define echoPin 3

#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>

// Define variables:
long duration;
int distance;
long sumDistance=0;
long avgDistance;

RF24 radio(7, 8); // CE, CSN

// Let these addresses be used for the pair
uint8_t address[][6] = {"1Node", "2Node"};

void setup() {
  // Define inputs and outputs
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

  if (!radio.begin()) {
    Serial.println(F("radio hardware is not responding!!"));
    while (1) {} // hold in infinite loop
  }
  radio.setPALevel(RF24_PA_MIN);
  radio.setPayloadSize(sizeof(avgDistance));
  radio.stopListening();
  // set the TX address of the RX node into the TX pipe
  radio.openWritingPipe(address[1]); // always uses pipe 0

  // Begin Serial communication at a baudrate of 9600:
  Serial.begin(115200);
}

void loop() {
  sumDistance=0;
  int j=0;
  for(int i=0; i<20;i++) {
    // Clear the trigPin by setting it LOW:
    digitalWrite(trigPin, LOW);

    delayMicroseconds(5);

    // Trigger the sensor by setting the trigPin high for 10 microseconds
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    // Read the echoPin. pulseIn() returns the duration (length of the p
    duration = pulseIn(echoPin, HIGH);

    // Calculate the distance:
    distance = duration*0.034/2;
    if (distance>800) {
      i--;
      j++;
    } else {
      sumDistance+=distance;
    }

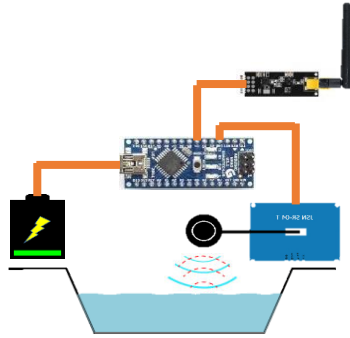
    if (j>=20) {
      sumDistance-=20;
      i=20;
    }
    delay(100);
  }
  avgDistance=sumDistance/20;

  radio.write(&avgDistance, sizeof(long));
  2
  // Print the distance on the Serial Monitor (Ctrl+Shift+M):
  Serial.print("Distance = ");
  Serial.print(avgDistance);
  Serial.println(" cm");
}
```

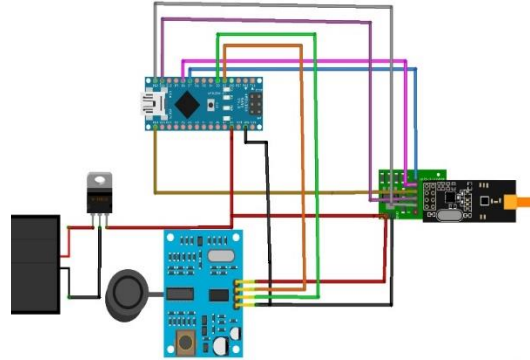
Gambar 3. 4 Desain program modul transmitter







Gambar 3. 6 Diagram satu garis modul *transmitter*

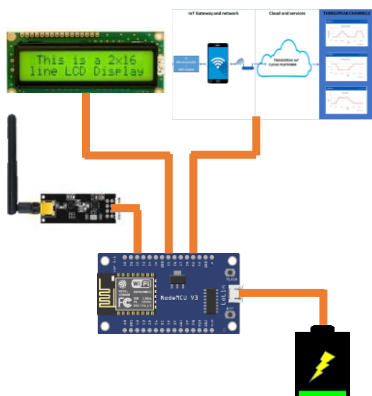


Gambar 3. 7 Diagram rangkaian modul *transmitter*

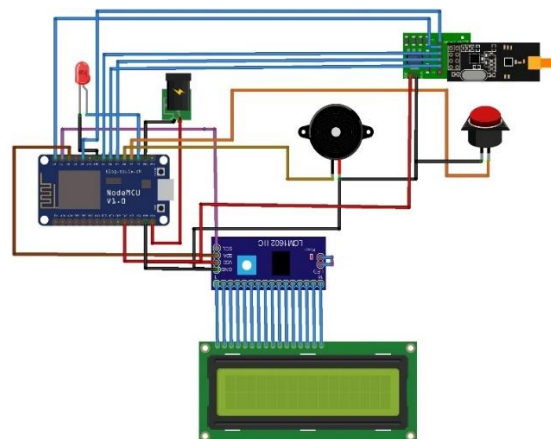
### 3.2.4 Desain Rangkaian Modul Receiver

Desain modul *receiver* meliputi NRF24L01+ PA LNA sebagai media komunikasi yang akan menerima data secara nirkabel dari modul *transmitter*, Nodemcu ESP8266 sebagai pengontrol tambahan yang mengkonversi data jarak atau ketinggian ke tampilan LCD serta website *thingspeak* sebagai *cloud* dan *user interface* untuk menampilkan data melalui koneksi internet.

Desain modul *receiver* dilakukan dengan menggunakan *software eagle* dan *fritzing*. Desain diagram satu garis dan rangkaian modul receiver ditunjukkan pada gambar 3. 8 dan gambar 3. 9 berikut atau dapat dilihat pada Lampiran G.



Gambar 3. 8 Diagram satu garis modul *receiver*



Gambar 3. 9 Diagram rangkaian modul *receiver*

### **3.3 Metode Pengambilan Data**

Pada tahap ini dilakukan metode pengambilan data untuk menguji fungsi dan kehandalan sistem. Pengambilan data yang dilakukan mencakup pengambilan data sensor ultrasonik JSN-SR04T dan Modul nirkabel NRF24L01+ PA LNA.

#### **3.3.1. Pengambilan Data Sensor Ultrasonik JSN-SR04T**

Pengambilan data sensor ultrasonik JSN-SR04T dilakukan dengan cara menempatkan sensor ultrasonik JSN-SR04T pada sebuah *aquarium* berisi air dengan variasi ketinggian yang berbeda. Data ketinggian air yang dideteksi oleh sensor tersebut kemudian dikirimkan oleh modul nirkabel NRF24L01+ PA LNA untuk ditampilkan pada display LCD dan web *thingspeak*. Adapun lisensi *thingspeak* yang digunakan pada penelitian ini adalah *free license*. Perbandingan lisensi dan fitur *thingspeak* dapat dilihat pada lampiran H.

#### **3.3.2. Pengambilan Data Jarak Komunikasi Modul NRF24L01+ PA LNA**

Pengambilan data jarak komunikasi modul NRF24L01+ PA LNA dilakukan dengan cara memindahkan modul *receiver* pada variasi jarak yang berbeda.

#### **3.3.3. Waktu dan Tempat Penelitian**

Penelitian ini dilakukan pada bulan Mei 2022 sampai dengan Juni 2022. Adapun tempat penelitian dilakukan di PT. Pertamina Geothermal Energy Area Ulubelu, Kabupaten Tanggamus, Provinsi Lampung.

### 3.3.4. Alat dan Bahan

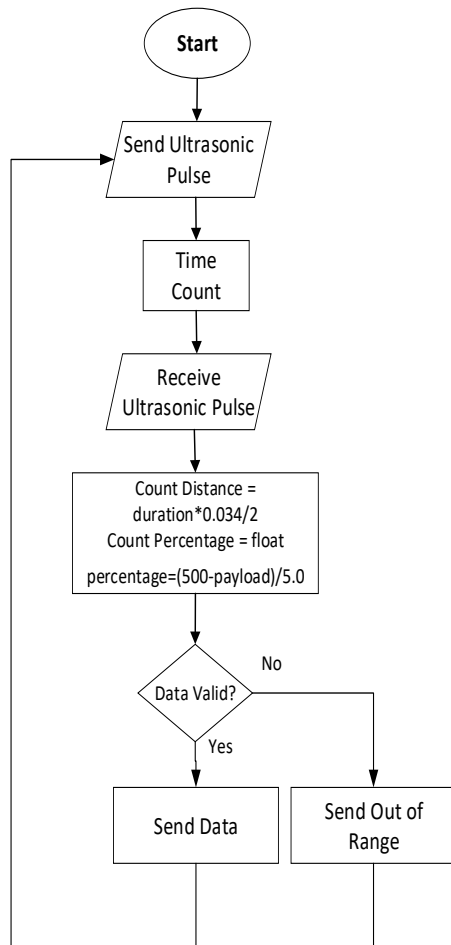
Adapun peralatan yang digunakan dalam penelitian ini ditampilkan pada tabel 3. 1 berikut.

Tabel 3. 1 Alat dan bahan

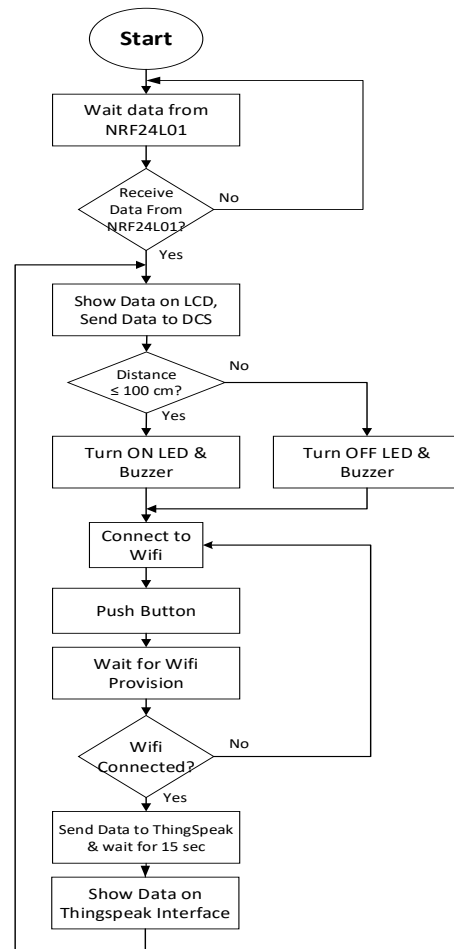
No	Nama	Jumlah
1	NRF24L01+ PA LNA	2
2	Arduino Nano	1
3	ESP8266 NodeMcu	1
4	DC Jack Barrel	2
5	Voltage to current converter 0-5v to 4-20mA	1
6	Power Supply 12V 1A	1
7	Push Button	1
8	Female Header	2
9	Male Header	1
10	LCD with I2C 1602	1
11	Capacitor 1000uF 6.8V	1
12	WaterProof Ultra sonic sensor JSN-SR04T	1
13	Buzzer	1
14	LED	1
15	Resistor 220 ½ W	1
16	Buck Converter MP1584	1

### 3.4. Diagram Alir atau *Flowchart*

Diagram alir sistem komunikasi antarmuka nirkabel berbasis modul transceiver NRF24L01+ PA LNA ditunjukkan pada gambar 3. 10 dan 3. 11 berikut.



Gambar 3. 10 Diagram alir *transmitter*



Gambar 3. 11 Diagram alir *receiver*

## **BAB IV**

### **PENGUJIAN DAN ANALISA HASIL IMPLEMENTASI**

#### **4.1 Implementasi Sistem Komunikasi**

Implementasi penelitian yang dilakukan adalah merancang sistem komunikasi antarmuka nirkabel pada perangkat portabel ketinggian air menggunakan modul *transceiver* NRF24L01+ PA LNA. Implementasi sistem komunikasi antarmuka nirkabel ini terdiri dari dua perangkat utama yaitu modul *transmitter* yang berfungsi sebagai pengirim sinyal berupa data jarak dari sensor ultrasonik JSN-SR04T dan modul *receiver* yang berfungsi sebagai penerima sinyal berupa data jarak yang selanjutnya akan ditampilkan pada tampilan antarmuka. Perangkat keras yang digunakan pada penelitian ini terdiri dari dua bagian utama yaitu sensor ultrasonik JSN-SR04T dan modul *transceiver* NRF24L01+ PA LNA+ PA/LNA.

##### **4.1.1. Implementasi Modul *Transmitter***

Modul *Transmitter* yang telah dirancang terdiri dari rangkaian modul sensor ultrasonik tahan air JSN-SR04T, Arduino Nano dan modul *transceiver* NRF24L01+ PA LNA. Sensor ultrasonik tahan air JSN-SR04T berfungsi untuk mengukur jarak atau ketinggian air. Sensor ultrasonik tersebut dihubungkan dengan mikrokontroler Arduino Nano menggunakan program Arduino IDE. Keluaran sensor ultrasonik tersebut kemudian diolah dan dikirimkan secara nirkabel oleh modul *transceiver* NRF24L01+ PA LNA. Hasil implementasi modul *transmitter* ditampilkan pada gambar 4. 1 berikut.



Gambar 4. 1 Hasil implementasi modul transmitter

#### 4.1.2. Implementasi Modul *Receiver*

Modul *Receiver* yang telah dirancang terdiri dari rangkaian modul *receiver* NRF24L01+ PA LNA, ESP8266 Nodemcu, LCD, buzzer, LED dan *provisioning Wi-fi*. Modul *receiver* NRF24L01+ PA LNA pada rangkaian ini berfungsi sebagai penerima sinyal data jarak dari modul *transmitter*. Data jarak atau ketinggian yang telah diterima modul kemudian ditampilkan pada display LCD dan dikirimkan ke *platform thingspeak* melalui kontroler ESP8266 Nodemcu. Jaringan internet dapat terkoneksi secara otomatis dengan pengaturan *provisioning Wi-fi*. Pengaturan *provisioning Wi-fi* dilakukan melalui aplikasi android *EspTouch: SmartConfig for ESP8266, ESP32*. Setelah terkoneksi dengan internet, data jarak dan ketinggian dapat dimonitor secara langsung pada *web thingspeak*. Adapun hasil implementasi modul *receiver* ditampilkan pada gambar 4. 2 berikut.



Gambar 4. 2 Hasil implementasi modul receiver

#### 4.1.3. Implementasi Tampilan Antarmuka

Tampilan antarmuka sistem komunikasi nirkabel yang dilakukan adalah dengan implementasi LCD dan platform *Thingspeak* pada gambar 4. 3 dan 4. 4 berikut.



Gambar 4. 3 Implementasi tampilan antarmuka LCD



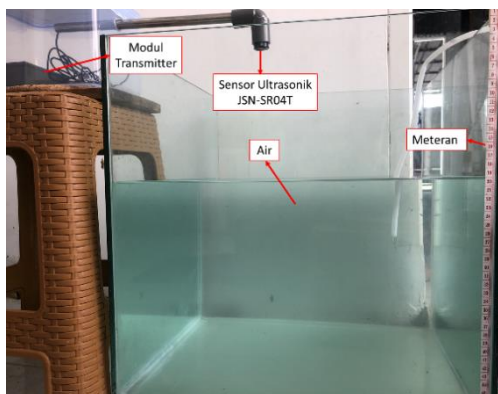
Gambar 4. 4 Implementasi tampilan antarmuka *thingspeak*



## 4.2 Pengujian Sensor Ultrasonik JSN-SR04T

Pengujian yang dilakukan adalah pengujian karakteristik dan pengujian akurasi sensor ultrasonik JSN-SR04T. Pengujian karakteristik sensor dilakukan dengan cara menempatkan sensor di atas aquarium berisi air dengan variasi ketinggian yang berbeda. Aquarium yang digunakan untuk pengambilan data memiliki ukuran panjang 50 cm, lebar 50 cm dan ketinggian 50 cm. Pada bagian sudut kanan, aquarium dilengkapi dengan meteran yang berfungsi untuk mengukur jarak atau ketinggian air secara manual. Kemudian untuk menguji kehandalan sensor pada kondisi hujan dilakukan simulasi dengan cara meneteskan air menggunakan selang.

Data jarak atau ketinggian air yang dideteksi oleh sensor kemudian dikirimkan oleh modul nirkabel NRF24L01+ PA LNA untuk ditampilkan pada display LCD. Selain itu, akan ditampilkan pada web *thingspeak* dengan waktu 15 detik. Data yang tertampil pada LCD kemudian dibandingkan dengan pengukuran manual untuk mendapatkan nilai deviasi atau *error* yang terjadi pada alat. Kondisi pengujian karakteristik sensor dapat dilihat pada gambar 4. 5 dan 4. 6 berikut.



Gambar 4. 5 Pengambilan data sensor ultrasonik JSN-SR04T



Gambar 4. 6 Tampilan data jarak pada LCD modul receiver

Adapun untuk pengujian akurasi dan *error* sensor, dilakukan dengan cara mengarahkan sensor ke dinding dari rentang jarak 100 cm - 600 cm. Mengingat sudut pengukuran sensor adalah sebesar 75°, maka sensor ditempatkan pada ketinggian 100 cm agar hasil pengukuran lebih valid. Dokumentasi pengujian akurasi dan *error* sensor ditunjukkan pada gambar 4.7 berikut.



Gambar 4. 7 Pengujian akurasi dan *error* sensor JSN-SR04T

Kemudian untuk mendapatkan nilai akurasi dan *error* dilakukan lima kali pengambilan sampel pada titik pengukuran yang sama. Tingkat prosentase nilai *error* dapat dihitung berdasarkan margin atau deviasi yang didapat pada jarak nyata atau pengukuran manual dengan hasil rata-rata pengukuran sensor [17], atau dinotasikan dengan menggunakan persamaan 2 berikut :

$$Error (\%) = \frac{\left(\frac{\sum f x}{n}\right) - s}{s} \times 100 \quad (2)$$

Dimana :

$\sum f x$  = Hasil penjumlahan pengukuran sensor

$n$  = Jumlah percobaan yang dilakukan

$s$  = Jarak pengukuran manual

#### 4.2.1. Hasil Pengujian Karakteristik Sensor Ultrasonik JSN-SR04T

Pengujian karakteristik sensor dilakukan dengan melakukan pengambilan 10 sampel data pada rentang pengukuran 5 cm sampai dengan 50 cm. Pengambilan data dilakukan bertahap pada setiap jarak 5 cm. Hasil pengujian karakteristik sensor ultrasonik JSN-SR04T ditunjukkan pada tabel 4. 1 berikut.

Tabel 4. 1 Hasil pengujian karakteristik sensor ultrasonik JSN-SR04T

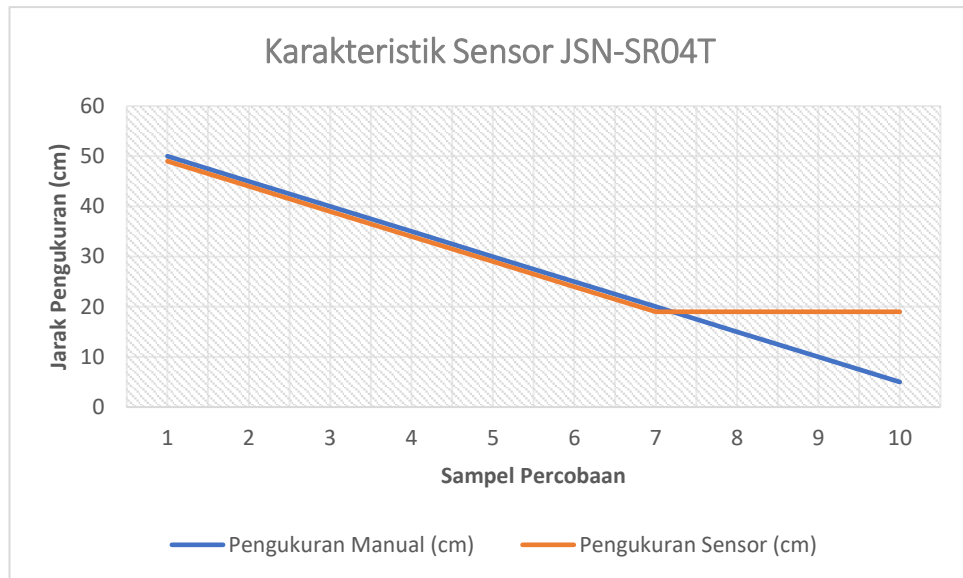
Sampel Percobaan	Pengukuran Manual (cm)	Pengukuran Sensor (cm)	Deviasi (cm)	Error (%)
1	50	49	1	2.00%
2	45	44	1	2.22%
3	40	39	1	2.50%
4	35	34	1	2.86%
5	30	29	1	3.33%
6	25	24	1	4.00%
7	20	19	1	5.00%
8	15	19	-4	<i>out of range</i>
9	10	19	-9	<i>out of range</i>
10	5	19	-14	<i>out of range</i>

#### 4.2.2. Analisa Hasil Pengujian Karakteristik Sensor Ultrasonik JSN-SR04T

Dari hasil pengambilan data di atas, didapatkan kecenderungan bahwa semakin dekat jarak pengukuran yang dilakukan, maka semakin tinggi nilai error yang terjadi. Hal ini terjadi dikarenakan nilai deviasi yang terjadi antara pengukuran manual dan pengukuran sensor cenderung stabil pada nilai deviasi 1 cm. Hal ini sesuai dengan spesifikasi yang telah dijabarkan oleh Purwanto (2019) bahwa *distance accuracy* sensor ultrasonik JSN-SR04T ini adalah  $\pm 1$  cm.[4]

Kemudian dari data di atas, dapat diketahui juga bahwa pada pengukuran jarak yang lebih rendah dari 20 cm, maka hasil pengukuran menunjukkan nilai yang sama yaitu 19 cm. Hal ini dikarenakan kemampuan sensor ultrasonik JSN-SR-04T

dalam mendeteksi jarak yaitu antara 20 cm – 600 cm sesuai dengan spesifikasi yang telah dijabarkan oleh Purwanto (2019) [4]. Grafik karakteristik sensor ultrasonik JSN-SR04T ditunjukkan pada gambar 4. 8 berikut.



Gambar 4. 8 Grafik karakteristik sensor ultrasonik JSN-SR04T

Selain itu, dapat diketahui pula bahwa ketahanan sensor terhadap tetesan air secara kontinyu menunjukkan hasil yang cukup baik. Hal tersebut ditunjukkan pada grafik 4.8 di atas dimana hasil pengukuran ketinggian air menggunakan sensor masih dapat terbaca akurat pada variasi ketinggian yang berbeda.

#### 4.2.3. Hasil Pengujian Akurasi Sensor Ultrasonik JSN-SR04T

Pengujian akurasi sensor dilakukan dengan melakukan pengambilan data pada rentang pengukuran 100 cm sampai 500 cm pada objek dinding. Pengujian dilakukan sebanyak lima kali sampel percobaan pada setiap jarak pengukuran untuk dapat mengetahui akurasi dan *error* penggunaan sensor.

**a. Pengujian Akurasi Sensor Ultrasonik JSN-SR04T pada Jarak 100 cm**

Dari pengujian yang telah dilakukan pada jarak 100 cm, didapatkan data pengukuran sensor yang ditampilkan pada tabel 4. 2 tersebut.

Tabel 4. 2 Hasil Pengujian Sensor Ultrasonik JSN-SR04T pada jarak 100 cm

Sampel Percobaan	Pengukuran Manual (cm)	Pengukuran Sensor (cm)	Deviasi (cm)	Konsistensi (%)	Error (%)	Akurasi (%)
1	100	95	5	100.00%	5.00%	95.00%
2	100	95	5			
3	100	95	5			
4	100	95	5			
5	100	95	5			
Rata-rata		95	5			

Data pada tabel 4. 2 di atas menampilkan perbandingan hasil pengukuran yang dilakukan antara pengukuran manual dan pengukuran dengan menggunakan sensor pada jarak 100 cm. Dari lima kali sampel percobaan didapatkan rata-rata nilai akurasi sebesar 95%.

**b. Pengujian Akurasi Sensor Ultrasonik JSN-SR04T pada Jarak 200 cm**

Dari pengujian yang telah dilakukan pada jarak 200 cm, didapatkan data pengukuran sensor yang ditampilkan pada tabel 4. 3 tersebut.

Tabel 4. 3 Hasil Pengujian Sensor Ultrasonik JSN-SR04T pada jarak 200 cm

Sampel Percobaan	Pengukuran Manual (cm)	Pengukuran Sensor (cm)	Deviasi (cm)	Konsistensi (%)	Error (%)	Akurasi (%)
1	200	194	6	60.00%	2.70%	97.30%
2	200	195	5			
3	200	195	5			
4	200	195	5			
5	200	194	6			
Rata-rata		194.6	5.4			

Data pada tabel 4. 3 di atas menampilkan perbandingan hasil pengukuran yang dilakukan antara pengukuran manual dan pengukuran dengan menggunakan sensor pada jarak 200 cm. Dari lima kali sampel percobaan didapatkan rata-rata nilai akurasi sebesar 97.3%.

**c. Pengujian Akurasi Sensor Ultrasonik JSN-SR04T pada Jarak 300 cm**

Dari pengujian yang telah dilakukan pada jarak 300 cm, didapatkan data pengukuran sensor yang ditampilkan pada tabel 4. 4 tersebut.

Tabel 4. 4 Hasil Pengujian Sensor Ultrasonik JSN-SR04T pada jarak 300 cm

Sampel Percobaan	Pengukuran Manual (cm)	Pengukuran Sensor (cm)	Deviasi (cm)	Konsistensi (%)	Error (%)	Akurasi (%)
1	300	295	5	60.00%	1.87%	98.13%
2	300	295	5			
3	300	294	6			
4	300	293	7			
5	300	295	5			
Rata-rata		294.4	5.6			

Data pada tabel 4. 4 di atas menampilkan perbandingan hasil pengukuran yang dilakukan antara pengukuran manual dan pengukuran dengan menggunakan sensor pada jarak 300 cm. Dari lima kali sampel percobaan didapatkan rata-rata nilai akurasi sebesar 98.13%.

**d. Pengujian Akurasi Sensor Ultrasonik JSN-SR04T pada Jarak 400 cm**

Dari pengujian yang telah dilakukan pada jarak 400 cm, didapatkan data pengukuran sensor yang ditampilkan pada tabel 4. 5 tersebut.

Tabel 4. 5 Hasil Pengujian Sensor Ultrasonik JSN-SR04T pada jarak 400 cm

Sampel Percobaan	Pengukuran Manual (cm)	Pengukuran Sensor (cm)	Deviasi (cm)	Konsistensi (%)	Error (%)	Akurasi (%)
1	400	394	6	40.00%	1.70%	98.30%
2	400	393	7			
3	400	394	6			
4	400	392	8			
5	400	393	7			
Rata-rata		393.2	6.8			

Data pada tabel 4. 5 di atas menampilkan perbandingan hasil pengukuran yang dilakukan antara pengukuran manual dan pengukuran dengan menggunakan sensor pada jarak 400 cm. Dari lima kali sampel percobaan didapatkan rata-rata nilai akurasi sebesar 98.30%.

**e. Pengujian Akurasi Sensor Ultrasonik JSN-SR04T pada Jarak 500 cm**

Dari pengujian yang telah dilakukan pada jarak 500 cm, didapatkan data pengukuran sensor yang ditampilkan pada tabel 4. 6 tersebut.

Tabel 4. 6 Hasil Pengujian Sensor Ultrasonik JSN-SR04T pada jarak 500 cm

Sampel Percobaan	Pengukuran Manual (cm)	Pengukuran Sensor (cm)	Deviasi (cm)	Konsistensi (%)	Error (%)	Akurasi (%)
1	500	491	9	60.00%	1.92%	98.08%
2	500	491	9			
3	500	490	10			
4	500	490	10			
5	500	490	10			
Rata-rata		490.4	9.6			

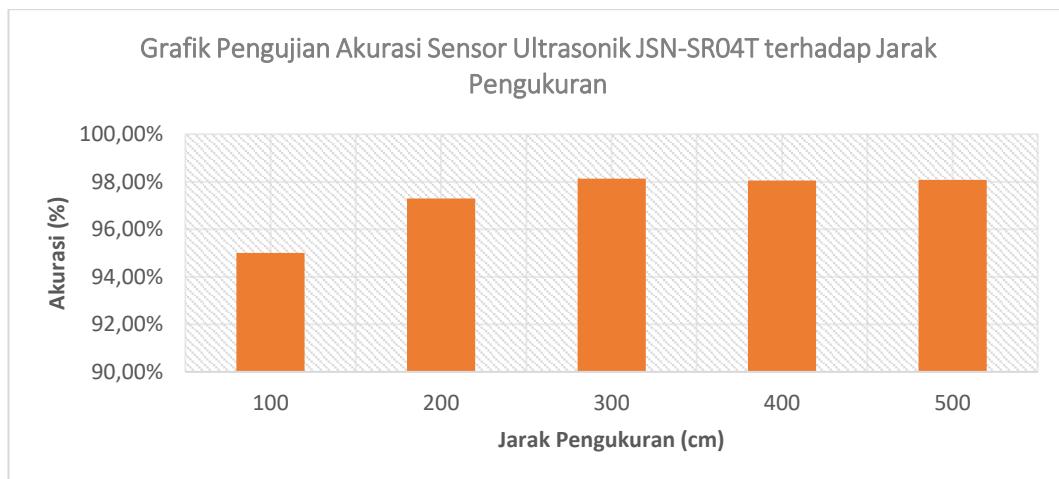
Data pada tabel 4. 6 di atas menampilkan perbandingan hasil pengukuran yang dilakukan antara pengukuran manual dan pengukuran dengan menggunakan sensor pada jarak 500 cm. Dari lima kali sampel percobaan didapatkan rata-rata nilai akurasi sebesar 98.08%.

#### 4.2.4. Analisa Hasil Pengujian Akurasi Sensor Ultrasonik JSN-SR04T

Dari pengambilan sampel data yang dilakukan, kemudian dilakukan kompilasi data untuk dapat menganalisa tingkat error pada setiap jarak pengukuran. Kompilasi pengujian sensor ditunjukkan pada tabel 4. 8 berikut.

Tabel 4. 7 Kompilasi Pengujian Sensor Ultrasonik JSN-SR04T pada jarak 600 cm

Sampel Percobaan	Pengukuran Manual (cm)	Pengukuran Sensor rata-rata (cm)	Deviasi (cm)	Error (%)	Akurasi (%)
1	100	95	5	5.00%	95.00%
2	200	194.6	5.4	2.70%	97.30%
3	300	294.4	5.6	1.87%	98.13%
4	400	392.2	7.8	1.95%	98.05%
5	500	490.4	9.6	1.92%	98.08%
Rata-rata	300	293.32	6.68	2.69%	97.31%



Gambar 4. 9 Grafik akurasi sensor ultrasonik JSN-SR04T

Dari grafik hasil pengujian akurasi sensor JSN-SR04T pada gambar 4. 9 di atas, dapat diketahui bahwa pengukuran sensor pada jarak 100 cm sampai dengan 500 cm menunjukkan rata-rata nilai akurasi 97.31%. Kemudian dapat diketahui juga bahwa jarak pengukuran sensor yang paling optimal adalah pada jarak 300 cm karena menunjukkan nilai akurasi yang paling besar yaitu 98.13%.



### 4.3 Pengujian Jangkauan Komunikasi Modul NRF24L01+ PA LNA

Pengujian jangkauan komunikasi modul NRF24L01+ PA LNA dilakukan dengan cara memindahkan modul *transmitter*. Pada pengujian ini, modul *transmitter* digunakan untuk mengukur jarak atau ketinggian air pada gorong-gorong setiap jarak 10 meter. Sementara modul *receiver* ditempatkan pada sebuah ruangan yang terhalang oleh bangunan. Untuk menguji komunikasi modul, dilakukan pengukuran pada jarak kurang dari 100 cm. *Buzzer* akan berbunyi ketika jarak yang terukur terbaca kurang dari 100 cm. Ketika *buzzer* tidak berbunyi dan pembacaan pada tampilan LCD tidak bergerak pada pengukuran kurang dari 100 cm, maka mengindikasikan bahwa modul sudah tidak mampu berkomunikasi. Untuk menguji konektivitas, dilakukan lima kali sampel percobaan pada setiap jangkauan jarak. Sehingga dari pengujian ini didapatkan jangkauan jarak maksimum yang dapat dioperasikan. Gambaran pengambilan data dapat dilihat pada gambar 4. 10 dan 4. 11 di bawah ini.



Gambar 4. 10 Penempatan modul transmitter



Gambar 4. 11 Penempatan modul receiver

#### 4.3.1. Pengambilan Data Modul Komunikasi NRF24L01+ PA LNA

Pengambilan data jarak komunikasi modul dilakukan dengan cara memindahkan modul *transmitter* pada variasi jarak yang berbeda. Sehingga dari pengujian ini didapatkan data seberapa jauh jangkauan dari modul NRF24L01+ PA LNA dalam mentransmisikan sinyal atau data. Adapun hasil pengambilan data yang telah diperoleh dijabarkan pada tabel 4. 9 berikut.

Tabel 4. 8 Hasil Pengujian Modul Komunikasi NRF24L01+ PA LNA

No	Jarak (meter)	Pengukuran Manual (cm)	Pengukuran Sensor (cm)	Error (%)	Akurasi (%)	Konektivitas (%)	Buzzer (ON/OFF)
1	10	57.2	57	0.35%	99.65%	100.00%	ON
2	20	59.15	59	0.25%	99.75%	100.00%	ON
3	30	59.6	59	1.01%	98.99%	100.00%	ON
4	40	60.2	60	0.33%	99.67%	100.00%	ON
5	50	60.4	60	0.66%	99.34%	100.00%	ON
6	60	65.4	65	0.61%	99.39%	100.00%	ON
7	70	65.7	65	1.07%	98.93%	100.00%	ON
8	80	61.8	61	1.29%	98.71%	100.00%	ON
9	90	69.5	69	0.72%	99.28%	100.00%	ON
10	100	69.8	69	1.15%	98.85%	100.00%	ON
11	110	74.1	73	1.48%	98.52%	100.00%	ON
12	120	73.9	73	1.22%	98.78%	100.00%	ON
13	130	71	70	1.41%	98.59%	100.00%	ON
14	140	69.2	68	1.73%	98.27%	100.00%	ON
15	150	69.1	68	1.59%	98.41%	100.00%	ON
16	160	71	70	1.41%	98.59%	100.00%	ON
17	170	72.2	71	1.66%	98.34%	100.00%	ON
18	180	72.4	71	1.93%	98.07%	100.00%	ON
19	190	73.2	72	1.64%	98.36%	100.00%	ON
20	200	74.5	73	2.01%	97.99%	100.00%	ON
21	210	75	0	100.00%	0.00%	0.00%	OFF
22	220	75.3	0	100.00%	0.00%	0.00%	OFF
23	230	75.8	0	100.00%	0.00%	0.00%	OFF
RATA-RATA		67.47	66.65	1.18%	98.82%		

Data pada tabel 4. 9 di atas menampilkan hasil pengukuran ketinggian air mengalir pada gorong-gorong dengan variasi jarak yang berbeda. Dilakukan

pengambilan data bertahap setiap jarak 10 meter untuk mengetahui kemampuan jangkauan komunikasi modul. Titik pengambilan sampel ditunjukkan pada gambar 4. 12 berikut.

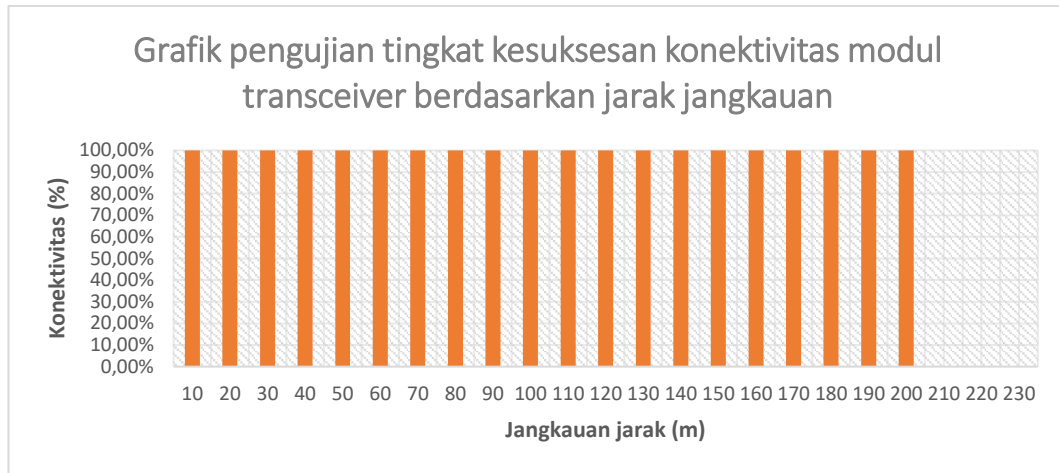


Gambar 4. 12 Titik pengambilan data NRF24L01+ PA LNA

#### 4.3.2. Analisa Hasil Pengambilan Data Modul Komunikasi

Dari pengujian yang telah dilakukan, kemudian dilakukan analisa data untuk dapat mengetahui jangkauan jarak yang dapat ditempuh oleh modul. Dari hasil pengujian modul komunikasi di atas, didapatkan hasil bahwa rata-rata error sebesar 1.18% dan diketahui bahwa terdapat fluktuasi nilai deviasi. Fluktuasi deviasi pengukuran yang terjadi disebabkan karena pengaruh riak air yang mengalir pada gorong-gorong. Sehingga mengakibatkan nilai pengukuran menjadi berubah-ubah.

Kemudian dilakukan analisa jangkauan jarak komunikasi maksimum dari modul NRF24L01+ PA LNA. Analisa tersebut ditunjukkan pada grafik pada gambar 4. 13 berikut.



Gambar 4. 13 Grafik pengujian tingkat kesuksesan konektivitas modul transceiver berdasarkan jarak jangkauan

Grafik di atas menunjukkan jangkauan komunikasi terhadap tingkat konektivitas pengukuran yang terjadi. Dari grafik tersebut diketahui bahwa pada jarak lebih dari 200 meter, konektivitas pengukuran menunjukkan nilai 0% yang ditandai dengan tampilan LCD yang menunjukkan nilai nol dan *buzzer* yang tidak berbunyi. Hal tersebut mengindikasikan modul sudah tidak mampu menerima sinyal komunikasi.

Salah satu faktor yang menyebabkan terbatasnya jangkauan jarak komunikasi modul tersebut adalah terjadinya *fading* pada saat penerimaan sinyal. *Fading* merupakan gejala pembenturan dalam jalur gelombang radio yang mengakibatkan bertambahnya redaman terhadap sinyal yang diterima. Redaman tersebut berupa bangunan dan perbedaan elevasi pada saat pengujian sehingga menghalangi proses transmisi data. Pada kondisi geografis yang tetap, *fading* dapat dipengaruhi oleh perubahan cuaca pada atmosfer. Karena perambatan gelombang radio harus satu garis pandang dan kondisi atmosfer yang berubah-ubah maka memungkinkan terjadinya refleksi, difraksi dan refraksi[18].

Kemudian dari pengujian yang telah dilakukan juga dapat diketahui bahwa jangkauan jarak komunikasi yang mampu ditempuh oleh modul NRF24L01+ PA LNA ini terbukti lebih jauh apabila dibandingkan dengan modul NRF24L01 versi sebelumnya yang memiliki spesifikasi jarak tempuh maksimum 100 meter pada kondisi tanpa halangan atau *Line of Sight*[19].

Sehingga dari pengujian yang telah dilakukan dapat diketahui bahwa kemampuan modul NRF24L01+ PA LNA dalam mentransmisikan data yaitu sejauh maksimum 200 meter pada kondisi halangan (*NLOS*).

#### **4.4 Integrasi Sistem sebagai Solusi Masalah Penelitian**

Berdasarkan implementasi dan pengujian yang telah dilakukan, dapat diketahui bahwa implementasi sistem komunikasi nirkabel pada antarmuka perangkat portabel monitoring ketinggian air berbasis modul transceiver NRF24L01+ PA LNA mampu mengatasi permasalahan monitoring ketinggian air di *cooling water pond* dengan cara mengganti metode pengukuran manual menjadi pengukuran secara *real time*.

Untuk menghasilkan metode pengukuran ketinggian air yang akurat dan handal dapat diketahui dari hasil pengujian sensor yang telah dilakukan. Dari pengujian sensor dapat diketahui bahwa penggunaan sensor ultrasonik JSN-SR04T pada jarak atau kedalaman 500 cm menunjukkan hasil yang cukup akurat dengan rata-rata akurasi 97.89%. Sehingga sensor JSN-SR04T dapat menjadi solusi terkait metode pengukuran ketinggian air yang akurat dan handal.

Kemudian setelah dilakukan integrasi antarmuka antara sensor ultrasonik JSN-SR04T dengan modul transceiver NRF24L01+ PA LNA menunjukkan hasil

pengukuran ketinggian air *realtime* yang dapat dimonitor dari jarak jauh menggunakan website *thingspeak*. Data ketinggian air yang telah diukur oleh sensor mampu ditransmisikan secara nirkabel menggunakan modul NRF24L01+PA LNA dengan jarak maksimum 200 meter.

## **BAB V**

### **PENUTUP**

#### **5.1 Simpulan**

Setelah dilakukan penelitian ini, maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Implementasi sistem komunikasi nirkabel pada antarmuka perangkat portabel monitoring ketinggian air berbasis modul transceiver NRF24L01+ PA LNA mampu mengatasi permasalahan monitoring ketinggian air di *cooling water pond* dengan cara mengganti metode pengukuran manual menjadi pengukuran secara *real time*.
2. Penggunaan sensor ultrasonik JSN-SR04T pada jarak atau kedalaman 500 cm menunjukkan nilai rata-rata akurasi sebesar 97.31%.
3. Dari pengujian sistem komunikasi menunjukkan hasil monitoring ketinggian air *realtime* yang dapat dimonitor dari jarak jauh menggunakan website *thingspeak* dengan jangkauan komunikasi maksimum sepanjang 200 meter.

#### **5.2 Saran**

Berdasarkan penelitian yang telah dilakukan, terdapat beberapa saran untuk peningkatan dan pengembangan penelitian selanjutnya yaitu :

1. Modul komunikasi dapat diintegrasikan dengan system DCS (*Distributed Control System*) agar proses monitoring menjadi lebih cepat dan handal.

2. Perlu dilakukan pengujian jangkauan NRF24L01+ PA LNA menggunakan *spectrum analyzer* untuk mengetahui kualitas pengiriman sinyal secara terukur.
3. Perlu dilakukan pengujian sistem komunikasi pada kondisi tanpa halangan untuk mengetahui kemampuan modul NRF24L01+ PA LNA yang lebih komprehensif.
4. Perlu dilakukan pengujian sistem komunikasi *multichannel* pada modul NRF24L01+PA LNA.
5. Perlu dikembangkan menggunakan modul NodeMCU ESP32 yang memiliki fitur lebih lengkap daripada NodeMCU ESP8266.
6. Perlu dilakukan uji komparasi antara website *thingspeak* versi gratis dan versi berbayar.



## DAFTAR PUSTAKA

- [1] Wiranto, B. I. Setiawan, and S. K. Saptomo, "Sistem Kontrol Irigasi Otomatis Nirkabel Wireless Automatic Control Irrigation System," *J. Irig.*, vol. 9, no. 2, pp. 108–114, 2014.
- [2] N. Hidayati, "Modul interaksi manusia komputer," no. 200309005. Program Studi Manajemen Informatika, Bina Sarana Informatika Jakarta, 2017.
- [3] I. E. Nadeak, "Penentuan Pola Penyebaran Fluida Geotermal Bawah Permukaan Dengan Metode Geomagnet Daerah Panas Bumi Siogung-Ogung Kabupaten Samosir," Universitas Negeri Medan, 2016.
- [4] H. Purwanto, M. Riyadi, D. W. Widiastuti, and I. W. A. Kusuma, "Komparasi Sensor Ultrasonik HC-SR04 Dan JSN-SR04T Untuk Aplikasi Sistem Deteksi Ketinggian Air," *J. SIMETRIS*, vol. 10, no. 2, pp. 717–724, 2019.
- [5] U. J. Shobrina, R. Primananda, and R. Maulana, "Analisis Kinerja Pengiriman Data Modul Transceiver NRF24101 , Xbee dan Wifi ESP8266 Pada Wireless Sensor Network," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 2, no. 4, pp. 1510–1517, 2018.
- [6] D. W. Irawan, F. Rofii, and A. Qustoniah, "Rancang Bangun Close Circuit Television Secara Nirkabel Menggunakan Zigbee 2,4Ghz," *Widya Tek.*, vol. 26, no. 2, 2018, doi: 10.31328/jwt.v26i2.791.
- [7] P. Mubarokah, L., Handayani, "Karakteristik Redaman dan Shadowing dalam Ruang pada Kanal Radio 2,4 GHz.," *J. Tek. ITS*, vol. 4, no. 1, pp. 25–30, 2015.
- [8] A. Septiano and T. Ghozali, "Nrf 24L01 Sebagai Pemancar/Penerima Untuk Wireless Sensor Netwok," *J. TEKNO (Civil Engineering, Elektr. Engineering Ind. Engineering)*, vol. 17, no. April, pp. 26–34, 2020.
- [9] A. Budiman, "Sistem Monitoring Keamanan Pelayaran Nelayan Berbasis Internet Of Things," pp. 9–44, 2019.
- [10] Akhiruddin, "Rancang Bangun Alat Pendeteksi Ketinggian Air Sungai Sebagai Peringatan Dini Banjir Berbasis Arduino Nano," *J. Electr. Technol.*, vol. Vol.3 No., no. 3, pp. 174–179, 2018, [Online]. Available: <https://jurnal.uisu.ac.id/index.php/jet/article/view/963>.
- [11] S. Iksal, Suherman, "Perancangan Sistem Kendali Otomatisasi On-Off Lampu Berbasis Arduino dan Borland Delphi," *Semin. Nas. Rekayasa Teknol.*, no. November, pp. 117–123, 2018.

- [12] A. D. B. Sadewo, E. R. Widasari, and A. Muttaqin, "Perancangan Pengendali Rumah menggunakan Smartphone Android dengan Konektivitas Bluetooth," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 1, no. 5, pp. 415–425, 2017.
- [13] H. R. Iskandar, D. I. Saputra, and H. Yuliana, "Eksperimental Uji Kekerohan Air Berbasis Internet of Things Menggunakan Sensor DFRobot SEN0189 dan MQTT Cloud Server," *J. Umj*, no. Sigdel 2017, pp. 1–9, 2019, [Online]. Available: <https://jurnal.umj.ac.id/index.php/semnastek/article/download/5164/3444>.
- [14] U. A. Saputro and A. Tuslam, "Sistem Deteksi Kebakaran Berbasis Internet Of Things Dengan Pesan Peringatan Menggunakan NodeMCU," vol. 7, no. 1, pp. 24–30, 2022.
- [15] M. Y. F. I Wayan Utama, Wahyudi Oktavia Gama, I Gede Artha Negara, "Sistem Otomatis Penyiraman Bunga Tanaman Menggunakan Gemitir," vol. 4, no. 2, pp. 1–13, 2021.
- [16] Pertamina (Persero), "Energia Weekly 11 November 2019," no. 45, p. 20, 2019.
- [17] A. Amrullah, "Perbandingan Tingkat Akurasi Pengukuran Ketinggian Air pada Sensor HC-SR04, HY-SRF05, dan JSN-SR04T," vol. 7, no. 1, pp. 1–5, 2022.
- [18] S. Purwanto, "Pengaruh Fading Pada Sistem Komunikasi Gelombang Mikro Tetap dan Bergerak," vol. 3, no. 1, pp. 33–39, 2011.
- [19] Taryana Suryana, "Implementasi Modul Nirkabel nRF24L01+ Sebagai Media Pengiriman dan Penerimaan Data Dengan Antarmuka NodeMCU," pp. 1–26, 2021.

## LAMPIRAN A

### DATASHEET NRF24L01+ PA LNA



**HT Handson Technology**

**User Guide**

**nRF24L01+PA+LNA 2.4GHz RF Transceiver Module**

This module based on Nordic nRF24L01+ with integrated Power Amplifier (PA) and Low-Noise-Amplifier (LNA) for extended range of up to 1,000 meter. This transceiver IC operates in the 2.4GHz band and has many new features! Take all the coolness of the nRF2401A and add some extra pipelines, buffers, and an auto-retransmit feature. You can use it in you project without any wireless design, just need to leave a 8 pins interface to this module, use the SPI interface to control the module on transferring data.



**SKU: MDU1087**

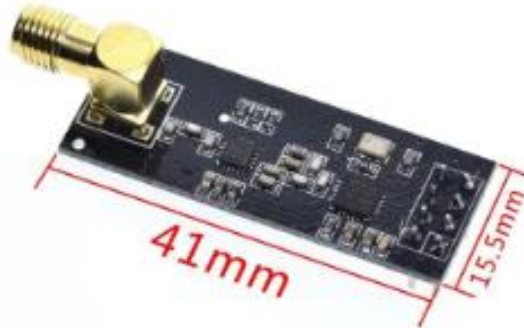
**Brief Data:**

- Worldwide license-free 2.4GHz ISM band.
- Integrated RF Power Amplifier and Low Noise Amplifier (LNA).
- Max Output Power: ~20dBm.
- Receiver Sensitivity: ~-95dBm.
- External SMA Antenna.
- 250kbps, 1Mbps and 2Mbps on-air data-rate.
- Communication Range: >1000+ meter (line of sight).
- 1.9 to 3.6V power supply range.
- ±60ppm 16MHz crystal.
- 5V tolerant inputs.

1 | [www.handsontec.com](http://www.handsontec.com)

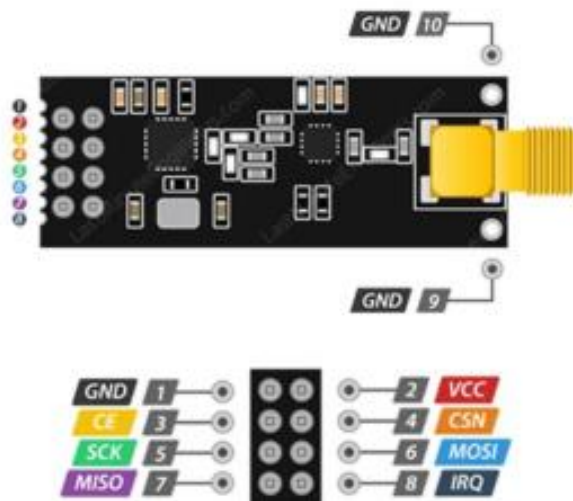
**Mechanical Dimension:**

Unit: mm



**Module Pins Assignment:**

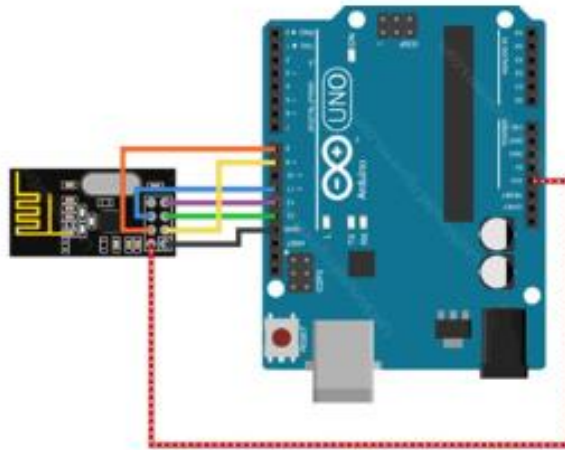
**nRF24L01+ PA/LNA Pinout**



Top View Pin Assignment

### **Connecting NRF24L01+ to Arduino:**

We need two modules (with Arduino) to complete this test, one board will act as transmitter and the other as receiver. The schematics for this project is quite simple, connect one NRF24L01 to each of the two Arduino boards that will be used for this project as shown in the schematics below.



NRF24L01+ Pin	Arduino Pin
GND	GND
VCC	+3.3V
CE	D9
CSN	D8
SCK	D13
MOSI	D11
MISO	D12

To interface the Arduino with the module, we'll be using TMRh20's RF24 library, which conveniently packages the low-level communications between the RF module and the MCU into an easy-to-use C++ class.

Download the library from the below link and copy to Arduino IDE libraries folder:

<https://github.com/nRF24/RF24>

The full documentation on the library can be found in the link.

### **Arduino Code – For Transmitter:**

In our experiment we will just send 'Welcome to Handson Tech' message from the transmitter to the receiver.

Here is the sketch we will be using for our transmitter:

```
//Include Libraries
#include <SPI.h>
#include <nRF24L01.h>
```

```

#include <RF24.h>
//create an RF24 object
RF24 radio(9, 8); // CE, CSN

//address through which two modules communicate.
const byte address[6] = "00001";

void setup()
{
  radio.begin();
  radio.setPALevel(RF24_PA_LOW); //Transmitter RF Power Setting
  //MIN=-18dBm, LOW=-12dBm, HIGH=-6dBm, MAX=0dBm.
  radio.setChannel(108); //above most WiFi frequencies. RF Channel setting 0-125
  radio.setDataRate( RF24_250Kbps );

  //set the address
  radio.openWritingPipe(address);

  //Set module as transmitter
  radio.stopListening();
}
void loop()
{
  //Send message to receiver
  const char text[] = "Welcome to HandsOn Tech";
  radio.write(&text, sizeof(text));

  delay(500);
}

```

### **Arduino Code – For Receiver:**

Here is the sketch we will be using for our receiver

```

//Include Libraries
#include <SPI.h>
#include <RF24L01.h>
#include <RF24.h>

//create an RF24 object
RF24 radio(9, 8); // CE, CSN

//address through which two modules communicate.
const byte address[6] = "00001";

void setup()
{
  while (!Serial);
  Serial.begin(9600);

  radio.begin();
  radio.setPALevel(RF24_PA_LOW); //Transmitter RF Power Setting
  //MIN=-18dBm, LOW=-12dBm, HIGH=-6dBm, MAX=0dBm.
  radio.setChannel(108); //above most WiFi frequencies. RF Channel setting 0-125
  radio.setDataRate( RF24_250Kbps );
  //set the address
  radio.openReadingPipe(0, address);

  //Set module as receiver
  radio.startListening();
}

void loop()
{
  //Read the data if available in buffer
  if (radio.available())
  {
    char text[32] = {0};
    radio.read(&text, sizeof(text));
    Serial.println(text);
  }
}

```

At the end we just print the received message on serial monitor. If you did everything ok and there are no mistakes in connections, you should see something like this in your Serial Monitor.

```
COM8
11:24:46.131 -> Welcome to Handson Tech
11:24:46.630 -> Welcome to Handson Tech
11:24:47.164 -> Welcome to Handson Tech
11:24:47.663 -> Welcome to Handson Tech
11:24:48.162 -> Welcome to Handson Tech
11:24:48.695 -> Welcome to Handson Tech
11:24:49.194 -> Welcome to Handson Tech
11:24:49.728 -> Welcome to Handson Tech
11:24:50.258 -> Welcome to Handson Tech
11:24:50.791 -> Welcome to Handson Tech
11:24:51.324 -> Welcome to Handson Tech
11:24:51.823 -> Welcome to Handson Tech
11:24:52.355 -> Welcome to Handson Tech
11:24:52.888 -> Welcome to Handson Tech
11:24:53.387 -> Welcome to Handson Tech
11:24:53.886 -> Welcome to Handson Tech
11:24:54.418 -> Welcome to Handson Tech

 Autoscroll  Show hexdump
Newline 9600 baud Clear output
```

**Web Resources:**

- <https://www.deviceplus.com/how-tos/arduino-guide/nrf24l01-rf-module-tutorial/>
- <https://github.com/nRF24/RF24>
- <https://lastminuteengineers.com/nrf24l01-arduino-wireless-communication/>
- <https://howtomechatronics.com/tutorials/arduino/arduino-wireless-communication-nrf24l01-tutorial/>
- <https://howtomechatronics.com/tutorials/arduino/how-to-build-an-arduino-wireless-network-with-multiple-nrf24l01-modules/>

## LAMPIRAN B

### DATASHEET JSN-SR04T

#### JSN-SR04T-2.0

##### 20-600 cm Ultrasonic Waterproof Range Finder

JSN-SR04T-2.0 ultrasonic distance measurement module can provide 20cm-600cm non-contact distance sensing function, ranging accuracy up to 2mm; module includes the transceiver of an integrated ultrasonic sensor and control circuit. Mode one usage and the Division's JSN-SR04T-2.0 module.

This product adopts industrial-grade integrated ultrasonic probe design, waterproof type, stable performance, all the MCU on the market. 1, the module performance is stable, the measurement distance is accurate. And foreign SRF05, SRF02 and other ultrasonic rangefinder module comparable. Module high precision, blind (20cm), stable range is the product successfully to the market a strong basis.

##### Features:

1. small size, easy to use;
2. low voltage, low power consumption;
3. high precision measurement;
4. strong anti-Interference;
5. Integrated closed waterproof cable probe, suitable for wet, bad measurement occasions

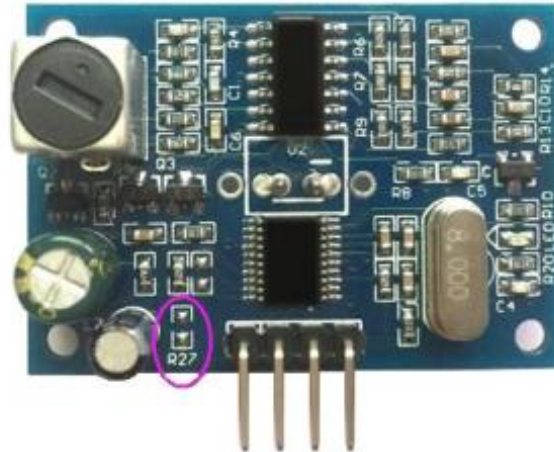
##### Specifications:

	Pulse width output / Serial Output
Operating Voltage	DC 3.0-5.5V
Working current	Less than 8mA
Probe frequency	40KHz
Farthest range	600cm
Recent range	20cm
Distance accuracy	+/- 1cm
Resolution	1mm
Measuring angle	75 degree
Enter the trigger signal	1, 10uS above the TTL pulse 2, the serial port to send instructions 0X55
Output the echo signal	Output pulse width level signal, or TTL
Wiring	3-5.5V (power positive) Trig (RX) RX Echo (output) TX GND (power supply negative)
Product Size	L42 * W29 * H12 mm
Operating temperature	-20 ° C to + 70 ° C
Product color	PCB board is blue

[www.jahankit.ir](http://www.jahankit.ir)



## Function Description:



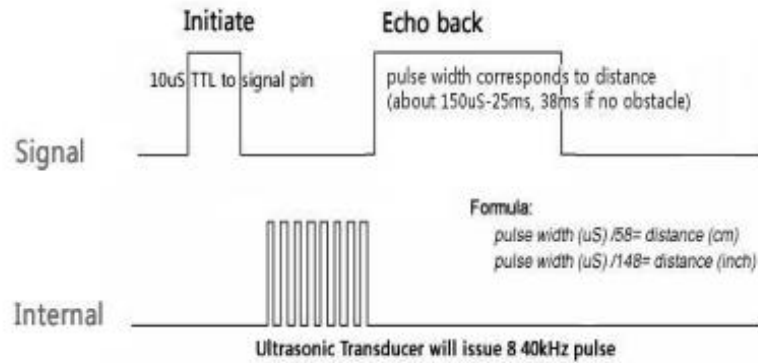
**Mode 1:** R27 = open that is not welding. The pattern is described below

1. the basic working principle:

- (1) using IO port TRIG trigger range, to a minimum of 10us high letter.
- (2) module automatically send 8 40khz square wave, automatically detect whether there is a signal to return;
- (3) a signal to return, through the IO port ECHO output a high level, high time is the duration of ultrasound from the launch to the return time. Test distance = (high time \* speed of sound (340M / s)) / 2;
- (4) module is triggered after the distance measurement, if you cannot receive the echo (the reason exceeds the measured range or the probe is not on the measured object), ECHO port will automatically become low after 60MS, marking the End of measurement, whether successful or not.
- (5) LED indicator, LED non-power indicator, it will receive the signal after the module will be lit, then the module is working.

[www.jahankit.ir](http://www.jahankit.ir)

2. ultrasonic timing diagram:



A short ultrasonic pulse is transmitted at the time 0, reflected by an object. The sensor receives this signal and converts it to an electric signal. The next pulse can be transmitted when the echo fades. This time period is called cycle period. The recommend cycle period should be no less than 50ms. If a 10µs width trigger pulse is sent to the signal pin, the ultrasonic module will output eight 40kHz ultrasonic signals and detect the echo back. The measured distance is proportional to the echo pulse width and can be calculated by the formula above. If no obstacle is detected, the output pin will give a 38ms high level signal.

**Mode 2:** R27 = 47K is the welding 47K resistance. The pattern is described below

Serial output format for the TTL level, that: 100MS module for the cycle of automatic output distance value, the unit is mm. Serial baud rate: 9600, n, 8, 1.

After the module is powered on, it will enter the working mode directly, and the range will be carried out every 100ms within the module and one frame will be output from the pin TX, including four 8-bit data. The frame format is: 0XFF + H\_DATA + L\_DATA + SUM

1. 0XFF: for a frame to start the data, used to judge;

2. H\_DATA: the upper 8 bits of the distance data;

3. L\_DATA: the lower 8 bits of the distance data;

4. SUM: data and, for the effect of its 0XFF + H\_DATA + L\_DATA = SUM (only low 8)

[www.jahankit.ir](http://www.jahankit.ir)

Note: H\_DATA and L\_DATA synthesize 16-bit data, that is, the distance in millimeters

E.G:

Product response FF 07 A1 A7

Where the check code  $SUM = A7 = (0x07 + 0xA1 + 0xFF) \& 0x00FF$

0x07 is the high data of the distance;

0xA1 is the lower data of the distance;

Distance value is 0x07A1; converted to decimal for 1953; unit: mm

Description: The module outputs the nearest distance value in the dead zone, and outputs 0 if the module does not measure the data or is out of range.

LED indicator, LED non-power indicator, the module connected to work after the light, then the module is working.

**Mode 3:** R27 = 120K that is welding 120K resistance. In the serial port mode

After the module is powered on, the module enters standby mode. The serial output format is TTL level, serial port baud rate: 9600, n, 8, 1. When the RX port receives the 0X55 instruction, the module starts a ranging and outputs one frame from the pin TX with four 8-bit data. The frame format is: 0XFF + H\_DATA + L\_DATA + SUM

1.0XFF: for a frame to start the data, used to judge;

2.H\_DATA: the upper 8 bits of the distance data;

3.L\_DATA: the lower 8 bits of the distance data;

4.SUM: data and, for the effect of its  $0XFF + H\_DATA + L\_DATA = SUM$  (only low 8)

Note: H\_DATA and L\_DATA synthesize 16-bit data, that is, the distance in millimeters

Description: The module outputs the nearest distance value in the dead zone, and outputs 0 if the module does not measure the data or is out of range.

LED indicator, LED non-power indicator, it will receive the 0X55 trigger signal after the module will be lit, then the module is working.

[www.jahankit.ir](http://www.jahankit.ir)

## LAMPIRAN C

### DATASHEET ARDUINO NANO

# *Arduino Nano (V2.3)*

## *User Manual*



Released under the Creative Commons Attribution Share-Alike 2.5 License

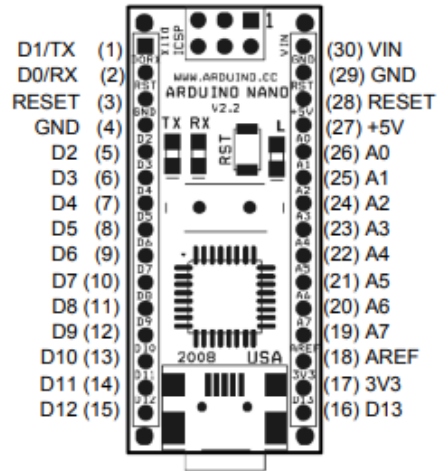
<http://creativecommons.org/licenses/by-sa/2.5/>

More information:

[www.arduino.cc](http://www.arduino.cc)

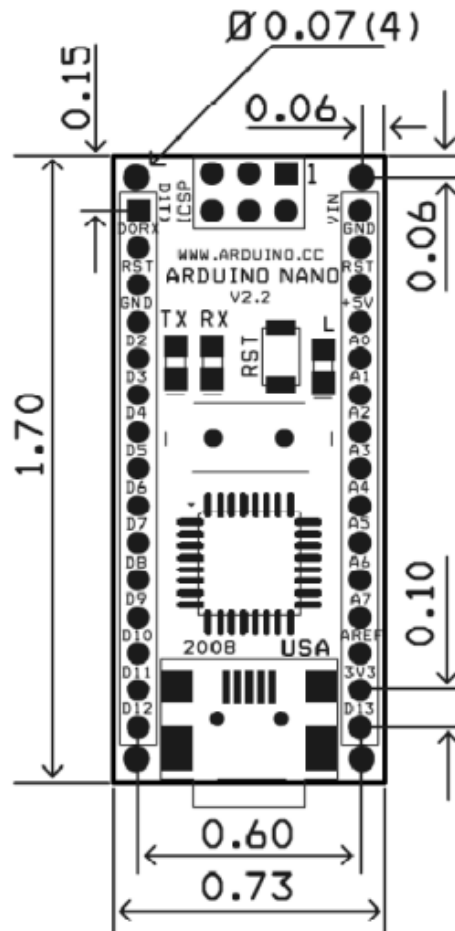
Rev. 2.3

## Arduino Nano Pin Layout



Pin No.	Name	Type	Description
1-2, 5-16	D0-D13	I/O	Digital input/output port 0 to 13
3, 28	RESET	Input	Reset (active low)
4, 29	GND	PWR	Supply ground
17	3V3	Output	+3.3V output (from FTDI)
18	AREF	Input	ADC reference
19-26	A7-A0	Input	Analog input channel 0 to 7
27	+5V	Output or Input	+5V output (from on-board regulator) or +5V (input from external power supply)
30	VIN	PWR	Supply voltage

**Arduino Nano Mechanical Drawing**



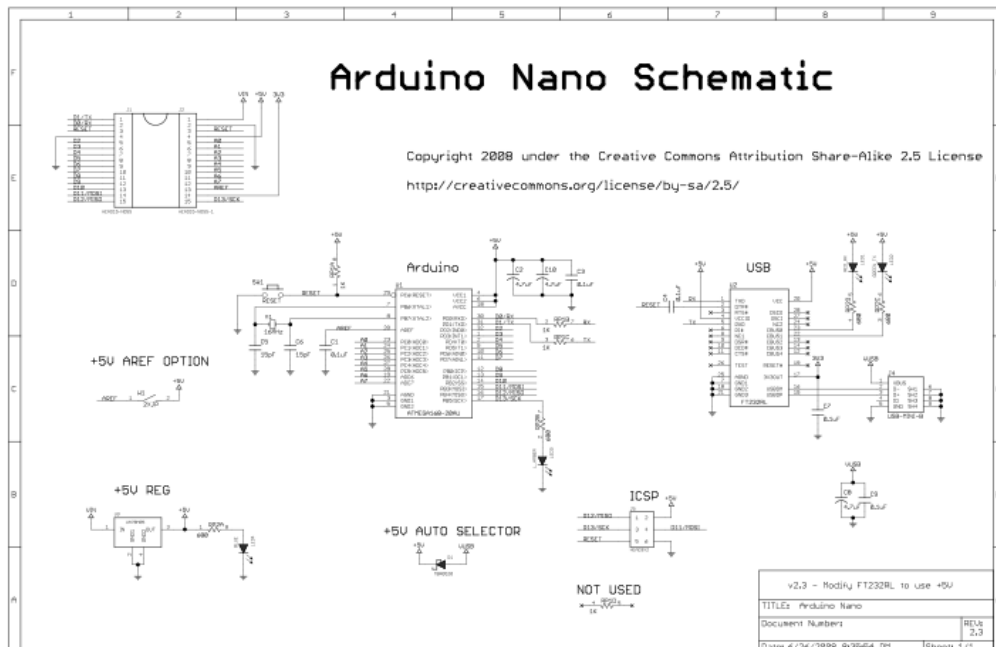
ALL DIMENSIONS ARE IN INCHES

### Arduino Nano Bill of Material

Item Number	Qty.	Ref. Dest.	Description	Mfg. P/N	MFG	Vendor P/N	Vendor
1	5	C1,C3,C4,C7,C9	Capacitor, 0.1uF 50V 10% Ceramic X7R 0805	C0805C104K5RACTU	Kemet	80-C0805C104K5R	Mouser
2	3	C2,C8,C10	Capacitor, 4.7uF 10V 10% Tantalum Case A	T491A475K010AT	Kemet	80-T491A475K010	Mouser
3	2	C5,C6	Capacitor, 18pF 50V 5% Ceramic NOP/COG 0805	C0805C180J5GACTU	Kemet	80-C0805C180J5G	Mouser
4	1	D1	Diode, Schottky 0.5A 20V	MBR0520LT1G	ONsemi	863-MBR0520LT1G	Mouser
5	1	J1,J2	Headers, 36PS 1 Row	68000-136HLF	FCI	649-68000-136HLF	Mouser
6	1	J4	Connector, Mini-B Recept Rt. Angle	67503-1020	Molex	538-67503-1020	Mouser
7	1	J5	Headers, 72PS 2 Rows	67996-272HLF	FCI	649-67996-272HLF	Mouser
8	1	LD1	LED, Super Bright RED 100mcd 640nm 120degree 0805	APT20125RCPRV	Kingbright	604-APT20125RCPRV	Mouser
9	1	LD2	LED, Super Bright GREEN 50mcd 570nm 110degree 0805	APHCM2012CGCK-F01	Kingbright	604-APHCM2012CGCK	Mouser
10	1	LD3	LED, Super Bright ORANGE 160mcd 601nm 110degree 0805	APHCM2012SECK-F01	Kingbright	04-APHCM2012SECK	Mouser
11	1	LD4	LED, Super Bright BLUE 80mcd 470nm 110degree 0805	LTST-C170TBKT	Lite-On Inc	160-1579-1-ND	Digikey
12	1	R1	Resistor Pack, 1K +/-5% 62.5mW 4RES SMD	YC164-JR-071KL	Yageo	YC164J-1.0KCT-ND	Digikey
13	1	R2	Resistor Pack, 680 +/-5% 62.5mW 4RES SMD	YC164-JR-07680RL	Yageo	YC164J-680CT-ND	Digikey
14	1	SW1	Switch, Momentary Tact SPST 150gf 3.0x2.5mm	B3U-1000P	Omron	SW1020CT-ND	Digikey
15	1	U1	IC, Microcontroller RISC 16kB Flash, 0.5kB EEPROM, 23 I/O Pins	ATmega168-20AU	Atmel	556-ATMEGA168-20AU	Mouser
16	1	U2	IC, USB to SERIAL UART 28 Pins SSOP	FT232RL	FTDI	895-FT232RL	Mouser
17	1	U3	IC, Voltage regulator 5V, 500mA SOT-223	UA78M05CDCYRG3	TI	595-UA78M05CDCYRG3	Mouser
18	1	Y1	Crystal, 16MHz +/-20ppm HC-49/US Low Profile	ABL-16.000MHZ-B2	Abracon	815-ABL-16-B2	Mouser

## Arduino Nano Schematic

Copyright 2008 under the Creative Commons Attribution Share-Alike 2.5 License  
<http://creativecommons.org/licenses/by-sa/2.5/>



## LAMPIRAN D

### DATASHEET NODEMCU ESP8266





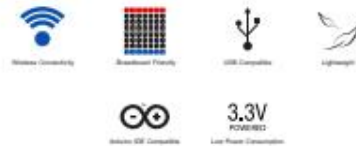
## NodeMCU ESP8266 ESP-12E WiFi Development Board

NodeMCU is an open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. The term "NodeMCU" by default refers to the firmware rather than the DevKit. The firmware uses the Lua scripting language. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson, and spiffs.



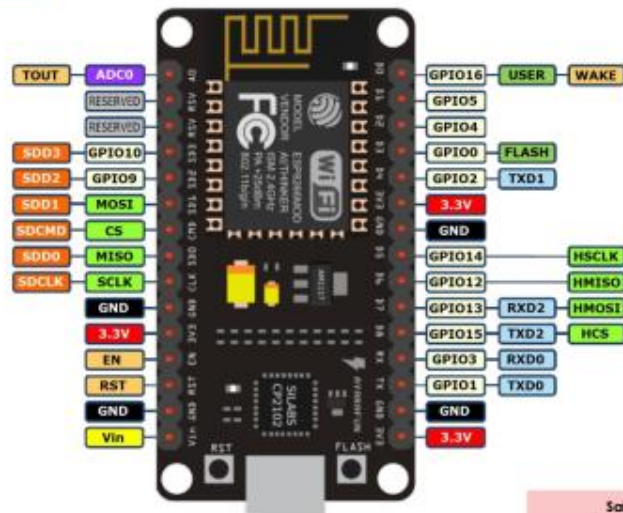
### Features

- Version : DevKit v1.0
- Breadboard Friendly
- Light Weight and small size.
- 3.3V operated, can be USB powered.
- Uses wireless protocol 802.11b/g/n.
- Built-in wireless connectivity capabilities.
- Built-in PCB antenna on the ESP-12E chip.
- Capable of PWM, I2C, SPI, UART, 1-wire, 1 analog pin.
- Uses CP2102 USB Serial Communication interface module.
- Arduino IDE compatible (extension board manager required).
- Supports Lua (alike node.js) and Arduino C programming language.



### PINOUT DIAGRAM

#### NodeMCU ESP8266 v1.0



Source  
<https://lotbytes.wordpress.com/nodemcu-pinout/>

**Safety Precaution**  
 All GPIO pins at 3.3V !!

## NodeMCU ESP8266



Front View



Front View

### Specifications of ESP-12E WiFi Module

<b>Wireless Standard</b>	IEEE 802.11 b/g/n
<b>Frequency Range</b>	2.412 - 2.484 GHz
<b>Power Transmission</b>	802.11b : +16 ± 2 dBm (at 11 Mbps) 802.11g : +14 ± 2 dBm (at 54 Mbps) 802.11n : +13 ± 2 dBm (at HT20, MCS7)
<b>Receiving Sensitivity</b>	802.11b : -93 dBm (at 11 Mbps, CCK) 802.11g : -85 dBm (at 54 Mbps, OFDM) 802.11n : -82 dBm (at HT20, MCS7)
<b>Wireless Form</b>	On-board PCB Antenna
<b>IO Capability</b>	UART, I2C, PWM, GPIO, 1 ADC
<b>Electrical Characteristic</b>	3.3 V Operated 15 mA output current per GPIO pin 12 - 200 mA working current Less than 200 uA standby current
<b>Operating Temperature</b>	-40 to +125 °C
<b>Serial Transmission</b>	110 - 921600 bps, TCP Client 5
<b>Wireless Network Type</b>	STA / AP / STA + AP
<b>Security Type</b>	WEP / WPA-PSK / WPA2-PSK
<b>Encryption Type</b>	WEP64 / WEP128 / TKIP / AES
<b>Firmware Upgrade</b>	Local Serial Port, OTA Remote Upgrade
<b>Network Protocol</b>	IPv4, TCP / UDP / FTP / HTTP
<b>User Configuration</b>	AT + Order Set, Web Android / iOS, Smart Link APP

#### Disclaimer

Information provided in this document are compilation from various online resources. Elnetronic Enterprise does not ensure the completeness, accuracy and reliability of the information and do not own any rights on any registered trademarks involved. Information provided should be intended for reference only.



**Related Sites**

**NodeMCU official site**

[http://nodemcu.com/index\\_en.html](http://nodemcu.com/index_en.html)

**NodeMCU Documentation**

<https://nodemcu.readthedocs.io/en/master/>

**NodeMCU Firmware (GitHub)**

<https://github.com/nodemcu/nodemcu-firmware>

**Project tagged with NodeMCU, HACKADAY.IO**

<https://hackaday.io/projects/?tag=NodeMCU>

**ESP8266 Getting started, by ACROBOTIC industries**

<http://learn.acrobotic.com/tutorials/post/esp8266-getting-started>

**Quick Start to Nodemcu (ESP8266) on Arduino IDE**

by Megesh Jayakumar

<http://www.instructables.com/id/Quick-Start-to-Nodemcu-ESP8266-on-Arduino-IDE/>

**GETTING STARTED WITH PLATFORMIO AND ESP8266 NODEMCU**

by Brandon Cannaday

<https://www.losant.com/blog/getting-started-with-platformio-esp8266-nodemcu>

**Programming ESP8266 ESP-12E NodeMCU V1.0 With Arduino IDE**

Into Wireless Temperature Logger

by Shin Teo


<http://www.instructables.com/id/ESP8266-NodeMCU-v10-ESP12-E-with-Arduino-IDE/>

For more details, we can be reached at the addresses below.

Terms & Condition apply.

**CONTACT INFORMATION**

 [www.einstronic.com](http://www.einstronic.com)

 010 - 2181014 ( Henry - Owner )

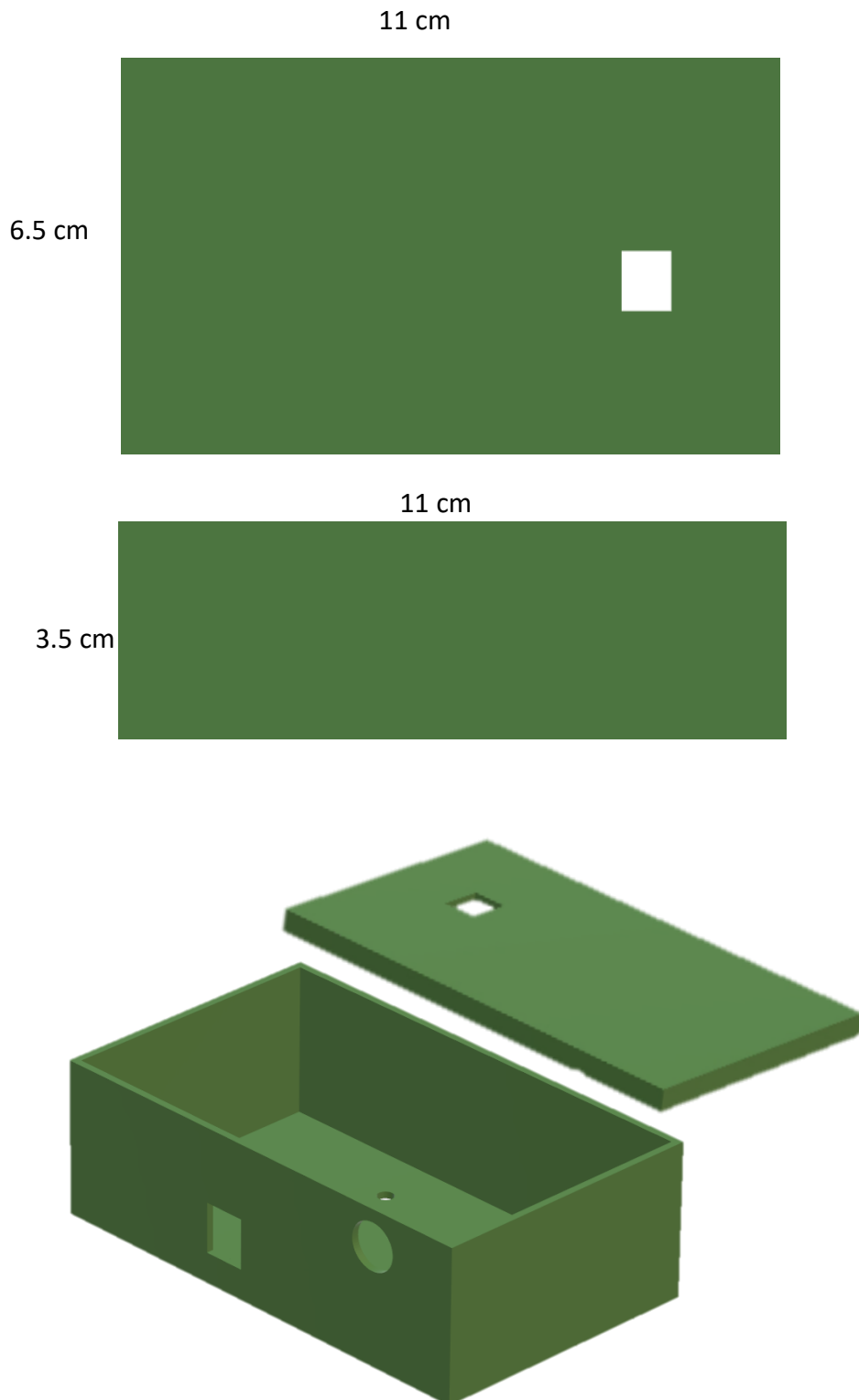
 [einstronics@gmail.com](mailto:einstronics@gmail.com)

 [facebook.com/einstronic](https://facebook.com/einstronic)

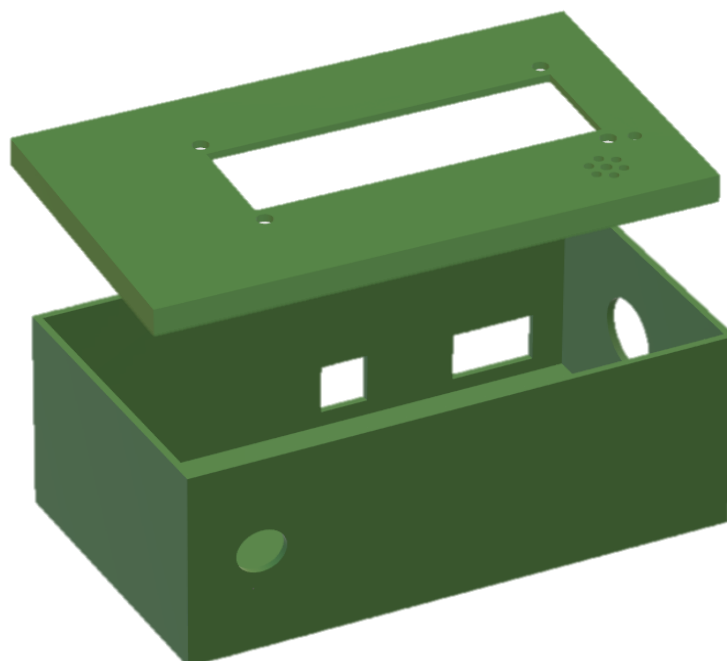
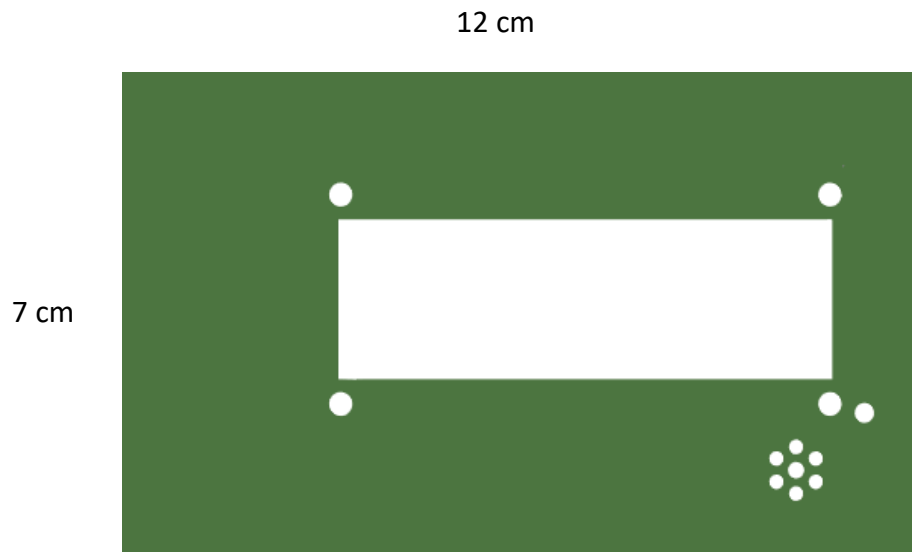
## LAMPIRAN E

### PERANCANGAN MEKANIK

#### a. Perancangan Mekanik Modul Transmitter



**b. Perancangan Mekanik Modul Receiver**



## LAMPIRAN F

### PERANCANGAN PROGRAM

#### a. Perancangan Program Transmitter

```
// Define Trig and Echo pin:
#define trigPin 2
#define echoPin 3

#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>

// Define variables:
long duration;
int distance;
long sumDistance=0;
long avgDistance;

RF24 radio(7, 8); // CE, CSN

// Let these addresses be used for the pair
uint8_t address[][6] = {"1Node", "2Node"};

void setup() {
  // Define inputs and outputs
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

  if (!radio.begin()) {
    Serial.println(F("radio hardware is not responding!!"));
    while (1) {} // hold in infinite loop
  }
  radio.setPALevel(RF24_PA_MIN);
  radio.setPayloadSize(sizeof(avgDistance));
  radio.stopListening();
  // set the TX address of the RX node into the TX pipe
  radio.openWritingPipe(address[1]); // always uses pipe 0

  // Begin Serial communication at a baudrate of 9600:
  Serial.begin(115200);
}

void loop() {
  sumDistance=0;
  int j=0;
  for(int i=0; i<20;i++) {
    // Clear the trigPin by setting it LOW:
    digitalWrite(trigPin, LOW);
```

```

delayMicroseconds(5);

// Trigger the sensor by setting the trigPin high for 10 microseconds:
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);

// Read the echoPin. pulseIn() returns the duration (length of the pulse) in microseconds:
duration = pulseIn(echoPin, HIGH);

// Calculate the distance:
distance = duration*0.034/2;
if (distance>800) {
  i--;
  j++;
} else {
  sumDistance+=distance;
}

if (j>=20) {
  sumDistance=-20;
  i=20;
}
delay(100);
}
avgDistance=sumDistance/20;

radio.write(&avgDistance, sizeof(long));

// Print the distance on the Serial Monitor (Ctrl+Shift+M):
Serial.print("Distance = ");
Serial.print(avgDistance);
Serial.println(" cm");
}

```

## b. Perancangan Program Receiver

```
#define SECRET_CH_ID 1 // replace 0000000 with your channel number
#define SECRET_WRITE_APIKEY "WB6FR3VZJ4X8X4N1" // replace XYZ with your
channel write API Key
#define PROV 3
#define pinLED 1
#define pinBuzzer 15

#include <SPI.h>
#include "printf.h"
#include "RF24.h"

#include <LiquidCrystal_I2C.h>

//LCD 16x2 at address I2C 0x27
LiquidCrystal_I2C lcd(0x27, 16, 2);

#include <ESP8266WiFi.h>
#include "ThingSpeak.h"

WiFiClient client;

unsigned long myChannelNumber = SECRET_CH_ID;
const char * myWriteAPIKey = SECRET_WRITE_APIKEY;

RF24 radio(2, 16); // the (ce_pin, csn_pin) connected to the radio

// Let these addresses be used for the pair
uint8_t address[][6] = {"1Node", "2Node"};

int payload = 0;

byte WiFiChar[] = {
  B00000,
  B01110,
  B10001,
  B00100,
  B01010,
  B00000,
  B00100,
  B00000
};

byte noWiFiChar[] = {
  B00000,
  B10001,
  B01010,
  B00100,
  B01010,
  B10001,
  B00000,
  B00000
};
```



```

byte configChar[] = {
  B00000,
  B01110,
  B10001,
  B10000,
  B10000,
  B10001,
  B01110,
  B00000
};

void setup() {
  Serial.begin(115200);
  pinMode(PROV,INPUT_PULLUP);
  pinMode(pinLED,OUTPUT);
  pinMode(pinBuzzer,OUTPUT);
  v_lcd_init();
  v_display_init();
  delay(2000);
  SPI.pins(14, 12, 13, 16);
  SPI.begin();
  if (!radio.begin(&SPI)) {
    Serial.println(F("radio hardware not responding!!"));
    while (1) {} // hold program in infinite loop to prevent subsequent errors
  }

  radio.setPALevel(RF24_PA_MAX); // RF24_PA_MAX is default.

  // set the RX address of the TX node into a RX pipe
  radio.openReadingPipe(1, address[1]); // using pipe 1
  radio.setPayloadSize(sizeof(payload));

  radio.startListening(); // put radio in RX mode

  wifiConnect();
  lcd.clear();
  if(WiFi.status() == WL_CONNECTED) {
    lcd.setCursor(15, 0);
    lcd.write(0);
  }
  ThingSpeak.begin(client); // Initialize ThingSpeak
} // setup

void loop() {
  if (digitalRead(PROV) == LOW) {
    lcd.setCursor(15, 0);
    lcd.write(2);
    smartConfig();
  }
  // Connect or reconnect to WiFi
  if(WiFi.status() != WL_CONNECTED){
    lcd.setCursor(15, 0);

```

```

    lcd.write(1);
    wifiConnect();
}
// This device is a RX node
uint8_t pipe;
if (radio.available(&pipe)) { // is there a payload? get the pipe number that received it
    uint8_t bytes = radio.getPayloadSize(); // get the size of the payload
    radio.read(&payload, bytes); // fetch payload from FIFO
    Serial.print(F("Received "));
    Serial.print(bytes); // print the size of the payload
    Serial.print(F(" bytes on pipe "));
    Serial.print(pipe); // print the pipe number
    Serial.print(F(": "));
    Serial.println(payload); // print the payload's value
}

if ((payload<=100)&&(payload!=0)) {
    digitalWrite(pinBuzzer,HIGH);
    digitalWrite(pinLED,HIGH);
} else {
    digitalWrite(pinBuzzer,LOW);
    digitalWrite(pinLED,LOW);
}

float percentage=(500-payload)/5.0;
lcd.setCursor(0, 0);
lcd.printf("Dist: %dcm", payload);
lcd.setCursor(0, 1);
lcd.printf("Percent: %.0f%% ", percentage);

if(WiFi.status() == WL_CONNECTED) {
    // set the fields with the values
    lcd.setCursor(15, 0);
    lcd.write(0);
    ThingSpeak.setField(1, payload);
    ThingSpeak.setField(2, percentage);
    // write to the ThingSpeak channel
    int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
    if(x == 200){
        Serial.println("Channel update successful.");
        delay(20000);
    }
    else{
        Serial.println("Problem updating channel. HTTP error code " + String(x));
    }
}
} // loop

bool bsmartConfig; //Smartconfig triggered
bool isWiFiConnected = 0; //is wifi connected
bool isSmartConfigDone = 1; //is smart config done
bool isSmartConfig = 0; //is smart config started
bool isConfigTime = 0; //is time need to be collected

```

```

bool wifiConnect()
{
  WiFi.begin();
  Serial.print("WiFi Connecting.");
  for (int i = 0; i < 40; i++)
  {
    if (WiFi.status() == WL_CONNECTED)
    {
      printWifiStatus();
      isWifiConnected = 1;
      //WiFi.printDiag(Serial);
      return true;
    }
    else
    {
      Serial.print(".");
      //Serial.println(wstatus);
      delay(500);
    }
  }
  Serial.println("\nWifi Connect Failed!");
  isWifiConnected = 0;
  return false;
  //WiFi.printDiag(Serial);
}

void smartConfig()
{
  isSmartConfig = 1;
  isSmartConfigDone = 0;
  isWifiConnected = 0;
  WiFi.mode(WIFI_STA);
  WiFi.beginSmartConfig();
  Serial.println("Start SmartConfig.");
  /* Wait for SmartConfig packet from mobile */
  Serial.println("Waiting for SmartConfig. Launch Mobile App (ex: ESP-TOUCH ) to progress SmartConfig.");
  while (!WiFi.smartConfigDone()) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("SmartConfig done.");
  isSmartConfig = 0;
  isSmartConfigDone = 1;

  /* Wait for WiFi to connect to AP */
  Serial.print("WiFi Connecting.");
  for (int i = 0; i < 60; i++)
  {
    if (WiFi.status() != WL_CONNECTED) {
      delay(500);
      Serial.print(".");
    } else {

```

```

    isWiFiConnected = 1;
    printWifiStatus();
    WiFi.setAutoConnect(true);
    return;
  }
}

Serial.println("\nWifi Connect Failed!");
isWiFiConnected = 0;
}

void printWifiStatus() {
  Serial.println("\nWifi Connected.");
  Serial.printf("SSID: %s\n", WiFi.SSID().c_str());
  Serial.printf("Password: %s\n", WiFi.psk().c_str());
  Serial.print("IP Address: ");
  Serial.println(WiFi.localIP());
}

//LCD initialization
void v_lcd_init() {
  lcd.init();
  lcd.clear();
  lcd.backlight();
  lcd.createChar(0, WiFiChar);
  lcd.createChar(1, noWiFiChar);
  lcd.createChar(2, configChar);
}

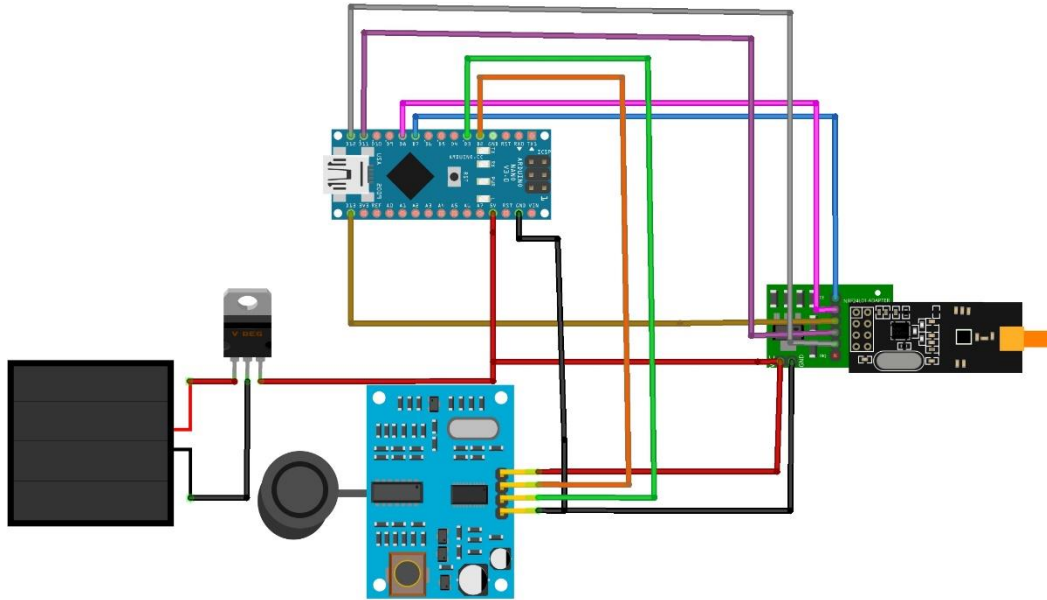
void v_display_init () {
  lcd.setCursor(1,0);
  lcd.print("Water Level");
  lcd.setCursor(4,1);
  lcd.print("Sensor");
}

```

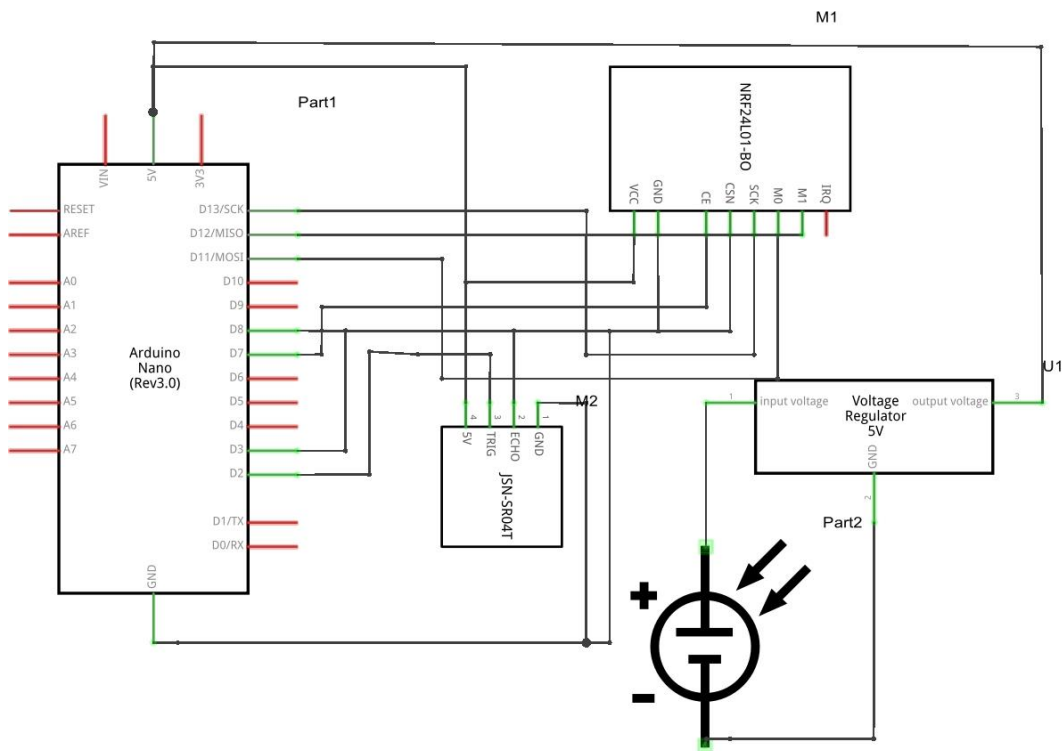
# LAMPIRAN G

## DESAIN SKEMATIK

a. Diagram Skematik Modul Transmitter menggunakan *fritzing*

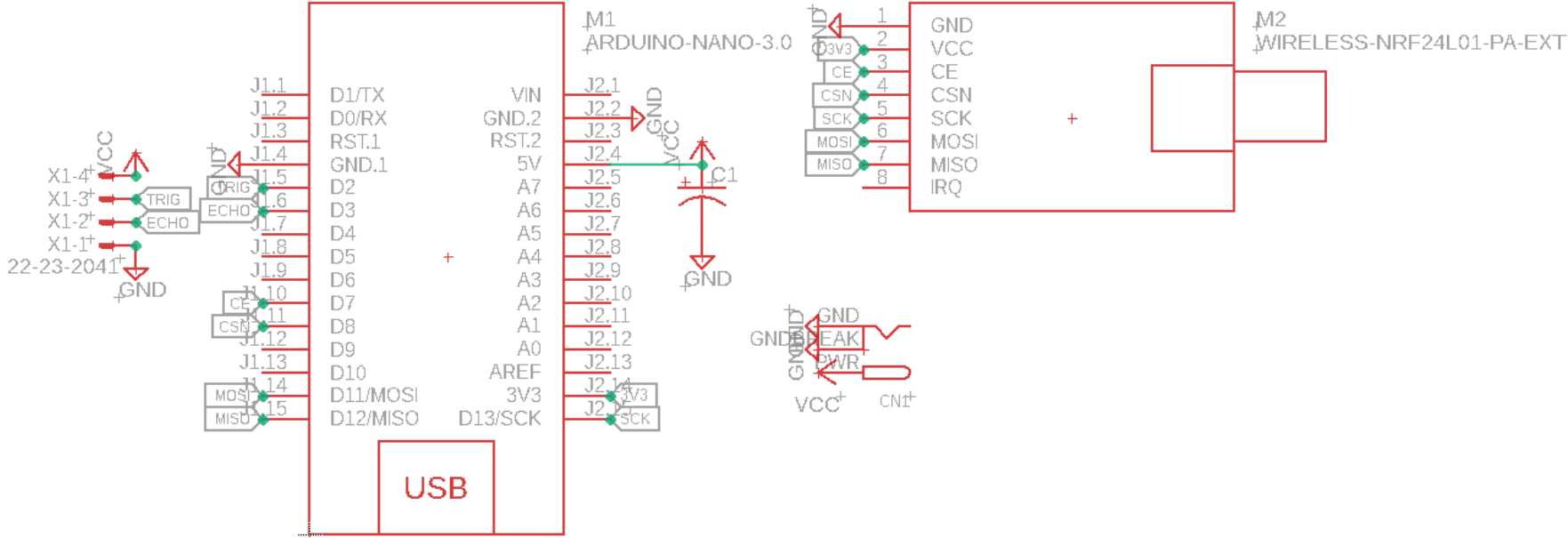


fritzing

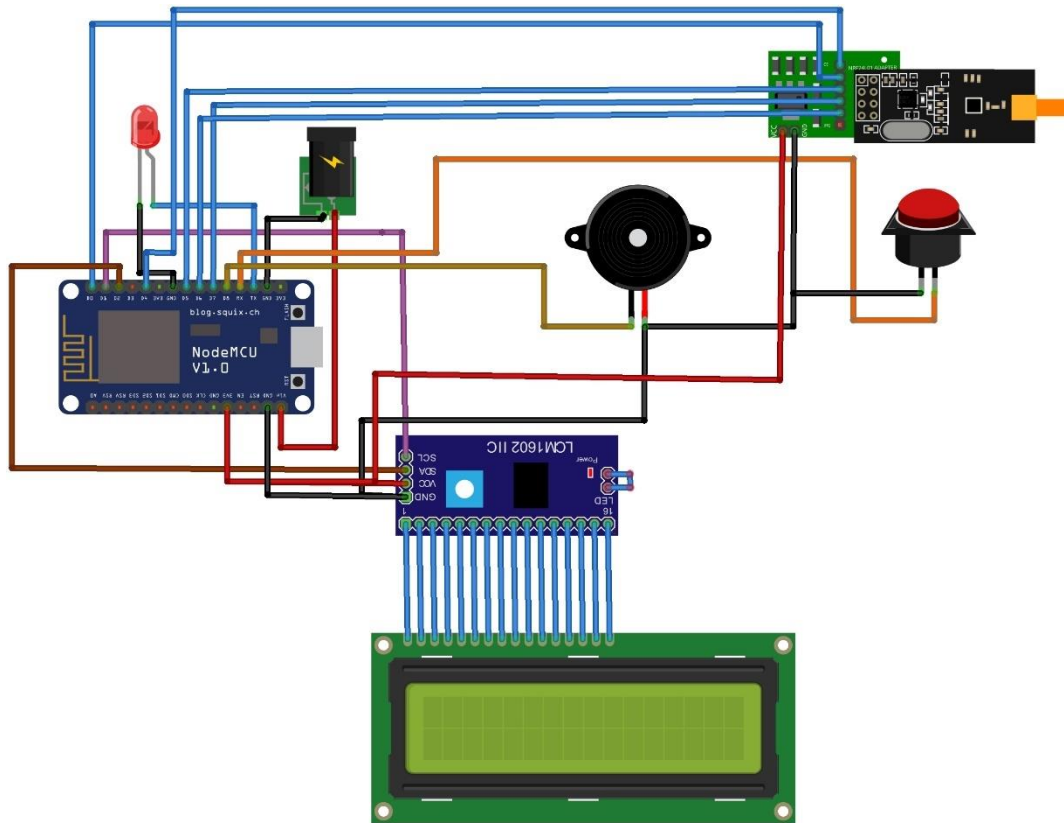


fritzing

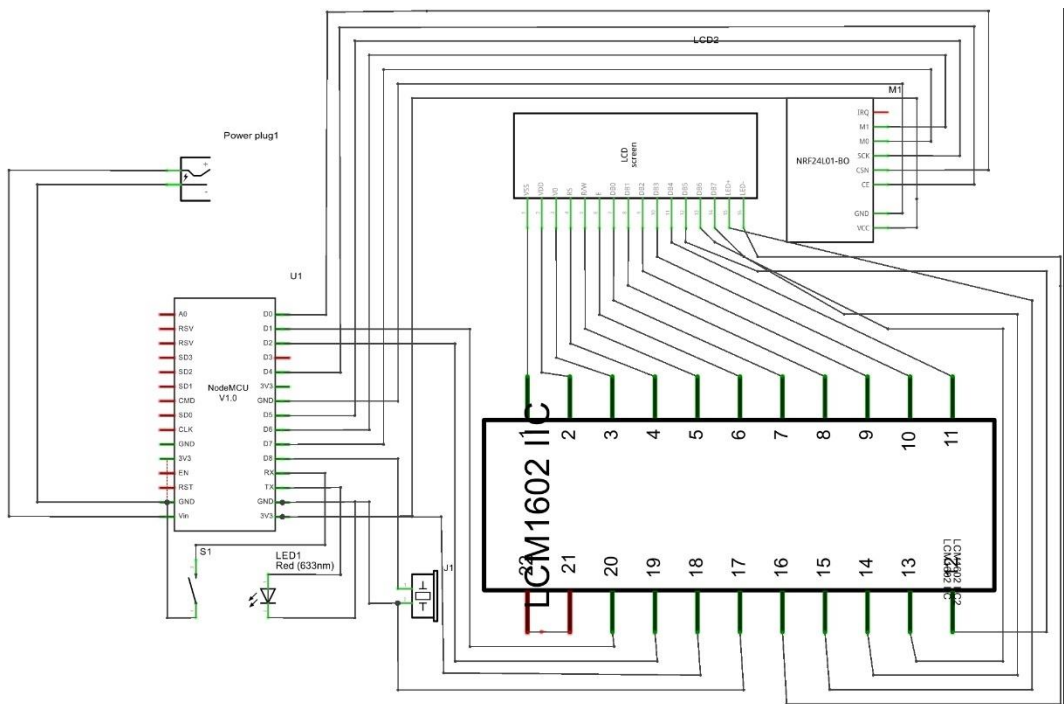
**b. Desain modul transmitter menggunakan software EAGLE**



c. Diagram Skematik Modul Receiver menggunakan *fritzing*

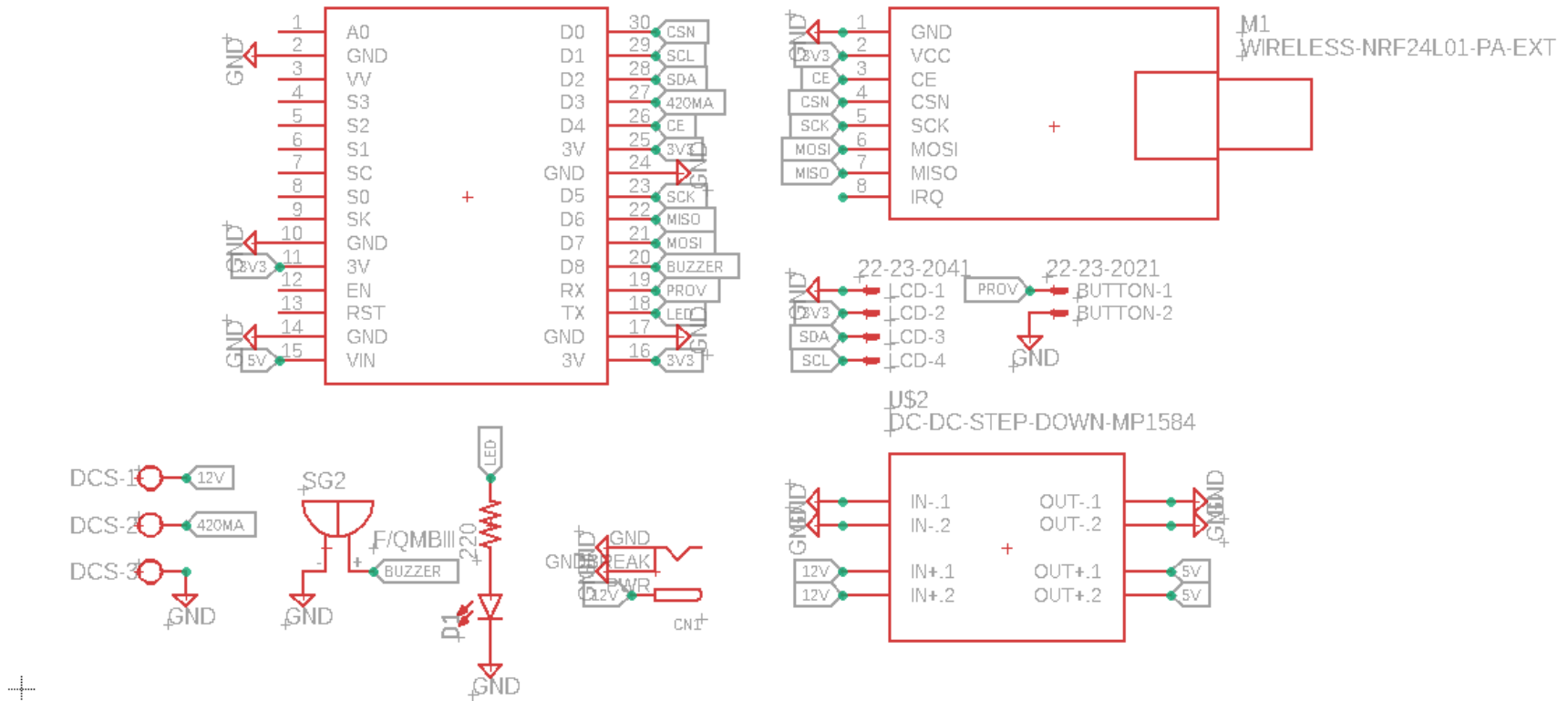


fritzing



fritzing

**d. Desain modul receiver menggunakan software EAGLE**





# LAMPIRAN H

## PERBANDINGAN LISENSI THINGSPEAK

### Home License

ThingSpeak home licenses are for personal use. Pricing is based on the number of channels required and on a count of messages to be processed and stored in a one-year period. ThingSpeak is available as a free service for small non-commercial home projects (<3 million messages/year or ~8,200 messages/day) with limits on capacity and update rates as well as other features. For larger projects and for expanded limits, a home license is available. See the table and Pricing Calculator below for details. ThingSpeak is bought in units, where one unit allows 33 million messages to be processed and stored in a one-year period (~90,000 messages/day). One unit (home license type) also provides the ability to create up to 10 channels on ThingSpeak. See Licensing FAQ (/pages/license\_faq) for more details.

Change country:  
Indonesia

Select a different license type:  
Home

### Why Buy a Home License?

With a home license, you can carry out Maker projects, setup home monitoring, and update ThingSpeak as fast as once per second. With MATLAB apps built into ThingSpeak, you can add custom analyses and visualizations to your data. You also have access to additional features such as private channel sharing and MQTT subscriptions.



	FREE For small non-commercial projects	HOME For personal use only <sup>(1)</sup>
Scalable for larger projects	✗ No. Annual usage is capped.	✓
Number of messages	3 million/year (~8,200/day) <sup>(2)</sup>	33 million/year per unit (~90,000/day per unit) <sup>(2)</sup>
Message update interval limit	Every 15 seconds	Every second
Number of channels	4	10 per unit
MATLAB Compute Timeout	20 seconds	20 seconds
Private channel sharing	Limited to 3 shares	Unlimited
Technical Support	Community Support ( <a href="https://www.mathworks.com/matlabcentral/topics/thingspeak.html">https://www.mathworks.com/matlabcentral/topics/thingspeak.html</a> )	Community Support ( <a href="https://www.mathworks.com/matlabcentral/topics/thingspeak.html">https://www.mathworks.com/matlabcentral/topics/thingspeak.html</a> )
Max image size	✗ Image feature unavailable	5 MB
Messages used per image	✗	100

<sup>(1)</sup>Not for government, commercial, academic-institutional or other organizational use.  
<sup>(2)</sup>Based on constant update rate.

### Pricing calculator

How many channels?

Currently:  To be added:

How often will they collect data?  
Every

Calculated number of channels needed: 10

Calculated Message Use**
2,628,000/year
216,000/month
7,200/day
300/hour
5/minute

Our Recommendation:  
Thingspeak units: 1  
License type: **Home**  
Maximum number of channels allowed: 10

\* Typically one channel per device  
\*\* Assumes constant message rate

### Purchase

License type: Home

ThingSpeak units:

x USD 59.00 price/unit/year

Total: **USD 59.00/year**

You will be taken to the MathWorks store to complete your purchase.

Questions about ThingSpeak purchasing?  
 Licensing FAQ ([/pages/license\\_faq](/pages/license_faq))  
 Contact us

Blog (<https://blogs.mathworks.com/iot/>) | Documentation (<https://www.mathworks.com/help/thingspeak/>) | Tutorials (<https://www.mathworks.com/help/thingspeak/getting-started-with-thingspeak.html>) | Terms ([https://www.mathworks.com/mw\\_account/company/aboutus/policies\\_statements/online-services-agreement.html](https://www.mathworks.com/mw_account/company/aboutus/policies_statements/online-services-agreement.html)) | Privacy Policy ([https://www.mathworks.com/company/aboutus/policies\\_statements](https://www.mathworks.com/company/aboutus/policies_statements))  
[Facebook](http://www.facebook.com/thingspeak) (<http://www.facebook.com/thingspeak>) [Twitter](http://www.twitter.com/thingspeak) (<http://www.twitter.com/thingspeak>) [RSS](http://feeds.feedburner.com/internetofthings) (<http://feeds.feedburner.com/internetofthings>) © 2022 The MathWorks, Inc. (<https://www.mathworks.com/>)

## Standard License

ThingSpeak standard licenses are for commercial use. Pricing is based on the number of channels required and on a count of messages to be processed and stored in a one-year period. With the standard license, you can connect more sensors, and update your data as fast as once per second and get access to paid-only features. ThingSpeak capacity is bought in units, where one unit allows 33 million messages to be processed and stored in a one-year period (~90,000 messages/day). One unit (standard license type) also provides the ability to create up to 250 channels on ThingSpeak. See the table and Pricing Calculator below and Licensing FAQ (/pages/license\_faq) for more details.

Commercial users may sign up for a time-limited free evaluation. The free evaluation of the service has limits on the amount of data you can send to ThingSpeak and the update rate (no faster than once every 15 seconds) as well as other features.

Change country:

Indonesia

Select a different license type:

Standard

Contact a Product Expert

## Why Buy a Standard License?

With a standard license, you can quickly build IoT applications without setting up web servers and without having expertise in web technology. Using the MATLAB apps built into ThingSpeak, you can develop custom visualizations of your data and schedule predictive analytics to run on your sensor data as it is passed to ThingSpeak. You also have access to live technical support, private channel sharing and MQTT subscriptions.



### FREE

For time-limited commercial evaluation of the service

### STANDARD

For all commercial, government and revenue generating activities

Scalable for larger projects	✗ No. Annual usage is capped.	✓
Number of messages	3 million/year (~8,200/day) <sup>(2)</sup>	33 million/year per unit (~90,000/day per unit) <sup>(1)</sup>
Message update interval limit	Every 15 seconds	Every second
Number of channels	4	250 per unit
MATLAB Compute Timeout	20 seconds	60 seconds
Private channel sharing	Limited to 3 shares	Unlimited
Technical Support	Community Support ( <a href="https://www.mathworks.com/matlabcentral/topics/thingspeak.html">https://www.mathworks.com/matlabcentral/topics/thingspeak.html</a> )	Standard MathWorks support
Max image size	✗ Image feature unavailable	5 MB
Messages used per image	✗	100

<sup>(1)</sup>Based on constant update rate.

## Pricing calculator



How many channels?\*

Currently: 0  
To be added: 10



How often will they collect data?

Every 2 Minutes

Calculated number of channels needed: 10

Calculated Message Use**
2,628,000/year
216,000/month
7,200/day
300/hour
5/minute

Our Recommendation:

Thingspeak units: 1  
License type: **Standard**  
Maximum number of channels allowed: 250

\* Typically one channel per device  
\*\* Assumes constant message rate

## Purchase

License type: Standard

ThingSpeak units: 1

x USD 710.00  
price/unit/year

Total: USD 710.00/year

Purchase

You will be taken to the MathWorks store to complete your purchase.

Questions about ThingSpeak purchasing?  
[Licensing FAQ](#) [Contact us](#)  
 (/pages/license\_faq)

Blog (<https://blogs.mathworks.com/iot/>) | Documentation (<https://www.mathworks.com/help/thingspeak/>) | Tutorials (<https://www.mathworks.com/help/thingspeak/getting-started-with-thingspeak.html>) | Terms ([https://www.mathworks.com/mw\\_account/company/aboutus/policies\\_statements/online-services-agreement.html](https://www.mathworks.com/mw_account/company/aboutus/policies_statements/online-services-agreement.html)) | Privacy Policy ([https://www.mathworks.com/company/aboutus/policies\\_statements](https://www.mathworks.com/company/aboutus/policies_statements))

[f](http://www.facebook.com/thingspeak) (<http://www.facebook.com/thingspeak>) [t](http://www.twitter.com/thingspeak) (<http://www.twitter.com/thingspeak>) [f](http://feeds.feedburner.com/internetofthings) (<http://feeds.feedburner.com/internetofthings>) © 2022 The MathWorks, Inc. (<https://www.mathworks.com/>)

## Student License

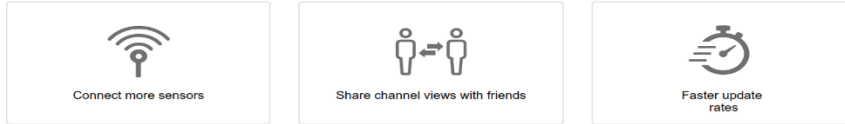
ThingSpeak student licenses are for student use. Pricing is based on the number of channels required and on a count of messages to be processed and stored in a one-year period. ThingSpeak is available as a free service for non-commercial student projects (<3 million messages/year or ~8,200 messages/day) with limits on capacity and update rates as well as other features. For larger student projects and for expanded limits, a student license is available. See the table and Pricing Calculator below for details. ThingSpeak is bought in units, where one unit allows 33 million messages to be processed and stored in a one-year period (~90,000 messages/day). One unit (student license type) also provides the ability to create up to 10 channels on ThingSpeak. See Licensing FAQ (/pages/license\_faq) for more details.

Change country:  
Indonesia

Select a different license type:  
Student

## Why Buy a Student License?

With a student license, you can send more data to ThingSpeak, and you can send data as fast as once per second. With MATLAB apps built into ThingSpeak, you can add custom analyses and visualizations to your data. You also have access to additional features such as private channel sharing and MQTT subscriptions.



	FREE For small non-commercial projects	STUDENT For students at degree-granting institutions <sup>(1)</sup>
Scalable for larger projects	✗ No. Annual usage is capped.	✓
Number of messages	3 million/year (~8,200/day) <sup>(2)</sup>	33 million/year per unit (~90,000/day per unit) <sup>(2)</sup>
Message update interval limit	Every 15 seconds	Every second
Number of channels	4	10 per unit
MATLAB Compute Timeout	20 seconds	20 seconds
Private channel sharing	Limited to 3 shares	Unlimited
Technical Support	Community Support ( <a href="https://www.mathworks.com/matlabcentral/topics/thingspeak.html">https://www.mathworks.com/matlabcentral/topics/thingspeak.html</a> )	Community Support ( <a href="https://www.mathworks.com/matlabcentral/topics/thingspeak.html">https://www.mathworks.com/matlabcentral/topics/thingspeak.html</a> )
Max image size	✗ Image feature unavailable	5 MB
Messages used per image	✗	100

<sup>(1)</sup>Not for government, commercial, academic-institutional or other organizational use.  
<sup>(2)</sup>Based on constant update rate.

## Pricing calculator

How many channels?  **Currently:**  **To be added:**

How often will they collect data? **Every**  **Minutes**

Calculated number of channels needed: 10

Calculated Message Use**
2,628,000/year
216,000/month
7,200/day
300/hour
5/minute

**Our Recommendation:**  
Thingspeak units: 1  
License type: **Student**  
Maximum number of channels allowed: 10

\* Typically one channel per device  
\*\* Assumes constant message rate

## Purchase

License type: **Student**

ThingSpeak units:

x USD 45.00 price/unit/year

**Total: USD 45.00/year**

You will be taken to the MathWorks store to complete your purchase.

Questions about ThingSpeak purchasing?  
[Licensing FAQ \(/pages/license\\_faq\)](#) [Contact us](#)

## Academic License

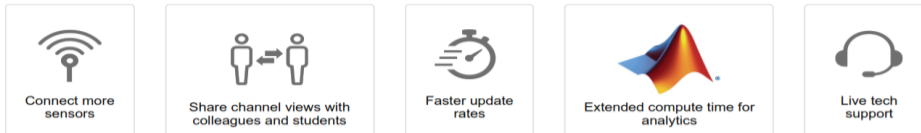
ThingSpeak academic licenses are for academic research and teaching. Pricing is based on the number of channels required and on a count of messages to be processed and stored in a one-year period. ThingSpeak is available as a free service for small non-commercial academic projects (<3 million messages/year or ~8,200 messages/day) with limits on certain features. For larger projects and for expanded limits, an academic license is available. See the table and Pricing Calculator below for details. ThingSpeak is bought in units, where one unit allows 33 million messages to be processed and stored in a one-year period (~90,000 messages/day). One unit (academic license type) also provides the ability to create up to 250 channels on ThingSpeak. See Licensing FAQ (/pages/license\_faq) for more details.

Change country:  
Indonesia

Select a different license type:  
Academic

## Why Buy an Academic License?

With an academic license, you can send more data to ThingSpeak, and you can update ThingSpeak as fast as once per second. Using the MATLAB apps built into ThingSpeak, you can develop custom visualizations of your data and schedule analytics to run on your sensor data as it is passed to ThingSpeak. You also have access to live technical support, private channel sharing and MQTT subscriptions.



	FREE For small non-commercial projects	ACADEMIC For academic use by faculty, staff, or researchers at degree-granting institutions <sup>(1)</sup>
Scalable for larger projects	✗ No. Annual usage is capped.	✓
Number of messages	3 million/year (~8,200/day) <sup>(2)</sup>	33 million/year per unit (~90,000/day per unit) <sup>(2)</sup>
Message update interval limit	Every 15 seconds	Every second
Number of channels	4	250 per unit
MATLAB Compute Timeout	20 seconds	60 seconds
Private channel sharing	Limited to 3 shares	Unlimited
Technical Support ( <a href="https://www.mathworks.com/matlabcentral/topics/thingspeak.html">https://www.mathworks.com/matlabcentral/topics/thingspeak.html</a> )	Community Support	Standard MathWorks support
Max image size	✗ Image feature unavailable	5 MB
Messages used per image	✗	100

<sup>(1)</sup>Not for government, commercial or other organizational use.  
<sup>(2)</sup>Based on constant update rate.

## Pricing calculator

How many channels?  **Currently:** 0 **To be added:**

How often will they collect data? **Every**  **Minutes**

**Calculated number of channels needed: 10**

Calculated Message Use**
2,628,000/year
216,000/month
7,200/day
300/hour
5/minute

**Our Recommendation:**  
Thingspeak units: 1  
License type: **Academic**  
Maximum number of channels allowed: 250

\* Typically one channel per device  
\*\* Assumes constant message rate

## Purchase

**License type:** Academic

**ThingSpeak units:**

x USD 275.00 price/unit/year

**Total:** **USD 275.00/year**

You will be taken to the MathWorks store to complete your purchase.

Questions about ThingSpeak purchasing?  
[🔗 Licensing FAQ \(/pages/license\\_faq\)](#)   [✉ Contact us](#)

Blog (<https://blogs.mathworks.com/iot/>) | Documentation (<https://www.mathworks.com/help/thingspeak/>) | Tutorials (<https://www.mathworks.com/help/thingspeak/getting-started-with-thingspeak.html>) | Terms ([https://www.mathworks.com/mw\\_account/company/aboutus/policies\\_statements/online-services-agreement.html](https://www.mathworks.com/mw_account/company/aboutus/policies_statements/online-services-agreement.html)) | Privacy Policy ([https://www.mathworks.com/company/aboutus/policies\\_statements](https://www.mathworks.com/company/aboutus/policies_statements))  
[f](http://www.facebook.com/thingspeak) (<http://www.facebook.com/thingspeak>) [t](http://www.twitter.com/thingspeak) (<http://www.twitter.com/thingspeak>) [R](http://feeds.feedburner.com/internetofthings) (<http://feeds.feedburner.com/internetofthings>) © 2022 The MathWorks, Inc. (<https://www.mathworks.com/>)