

MONITORING DAN PENGONTROLAN PENAMPUNG AIR

PROYEK AKHIR

Laporan akhir ini dibuat dan diajukan untuk memenuhi salah satu syarat kelulusan
Diploma III Politeknik Manufaktur Negeri Bangka Belitung



Disusun Oleh :

Della Devinda

NIRM 0031535

Mutiara Herdila

NIRM 0031554

POLITEKNIK MANUFAKTUR NEGERI

BANGKA BELITUNG

2018

LEMBAR PENGESAHAN

MONITORING DAN PENGONTROLAN PENAMPUNG AIR

Oleh :

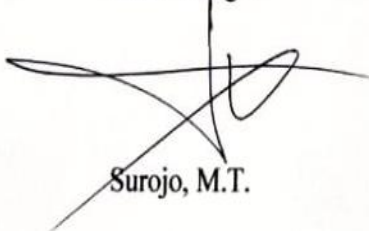
Della Devinda / 0031535

Mutiara Herdila / 0031554

Laporan akhir ini telah disetujui dan disahkan sebagai salah satu syarat kelulusan
Program Diploma III Politeknik Manufaktur Negeri Bangka Belitung

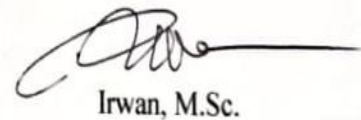
Menyetujui,

Pembimbing 1



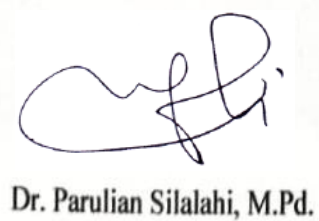
Surojo, M.T.

Pembimbing 2



Irwan, M.Sc.

Penguji 1



Dr. Parulian Silalahi, M.Pd.

Penguji 2



Eko Sulistyono, M.T.

Penguji 3



Ocsirendi, M.T.

PERNYATAAN BUKAN PLAGIAT

Yang bertanda tangan di bawah ini :

Nama Mahasiswa 1 : Della Devinda

NIRM : 0031535

Nama Mahasiswa 2 : Mutiara Herdila

NIRM : 0031554

Dengan Judul : Monitoring dan Pengontrolan Penampung Air

Menyatakan bahwa laporan akhir ini adalah hasil kerja kami sendiri dan bukan merupakan plagiat. Pernyataan ini kami buat dengan sebenarnya dan bila ternyata di kemudian hari ternyata melanggar pernyataan ini, kami bersedia menerima sanksi yang berlaku.

Sungailiat, Agustus 2018

Nama Mahasiswa	Tanda Tangan
1. Della Devinda
2. Mutiara Herdila

ABSTRAK

Penelitian ini dilakukan untuk membuat sebuah alat monitoring dan pengontrolan penampung air dengan memanfaatkan konektivitas internet dalam pengendaliannya. Kontrol dilakukan dengan menggunakan aplikasi yang sudah dibangun pada Android. Ini bertujuan agar memberikan kemudahan bagi pengguna dalam melakukan pengontrolan dan memberikan rasa aman kepada pengguna karena dapat mengurangi tingkat bahaya seperti korsleting. Alat ini menggunakan beberapa perangkat untuk mengintegrasikan proses antara hardware dan aplikasi. Esp8266 digunakan sebagai modul wifi sekaligus sebagai broker atau sarana pertukaran data antara sensor dan aplikasi . Sensor Ultrasonik digunakan sebagai pendeteksi ketinggian air. Voltage Regulator digunakan untuk memberikan tegangan 3,3V yang dibutuhkan oleh Esp8266. Output dari alat ini adalah relay yang berfungsi untuk mengaktifkan pompa, lcd untuk menampilkan hasil monitoring ketinggian air. Aplikasi yang dibangun menggunakan Blynk. Hasil dari pengujian yang telah dilakukan adalah alat mampu mengatur pengisian air menggunakan Android dalam dua kondisi yaitu manual dan otomatis. Dalam kondisi manual sinyal yang dikirimkan dengan perintah inputan on/off, sedangkan saat otomatis sinyal yang dikirimkan berdasarkan kondisi dari water level yang berasal dari pembacaan sensor ultrasonik. Penggunaan sensor ultrasonik memberikan hasil yang baik dan sangat efektif digunakan sebagai pendeteksi ketinggian air.

Kata kunci : *arduino nano, blynk, esp8266, sensor Ultrasonic*

ABSTRACT

This research was conducted to make a monitoring tool and control water storage by utilizing internet connectivity in its control. Control is done using an application that has been built on Android. It aims to make it easier for users to control and provide a sense of security to users because it can reduce the level of danger such as short circuit. This tool uses several devices to integrate the process between hardware and applications. Esp8266 is used as a wifi module as well as a broker or means of data exchange between sensors and applications. The Ultrasonic Sensor is used as a water level detector. Voltage Regulator is used to provide the 3.3V voltage required by ESP8266. The output of this device is a relay that functions to activate the pump, LCD to display the results of monitoring water level. Application built using Blynk. The results of the tests that have been carried out are tools capable of regulating the filling of water using Android in two conditions, namely manual and automatic. In the manual condition the signal is sent with the on / off input command, while the automatic signal sent is based on the condition of the water level coming from the ultrasonic sensor reading. The use of ultrasonic sensors provides good results and is very effective to be used as a water level detector.

Keywords: *arduino nano, blynk, ESP8266, Ultrasonic sensor*

KATA PENGANTAR

Alhamdulillah atas berkat dan limpahan rahmat dari Allah SWT sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul “*Monitoring dan Pengontrolan Penampung Air*”. Tugas akhir ini disusun dengan maksud memenuhi salah satu persyaratan guna menyelesaikan studi di Program Studi D3 Teknik Elektronika.

Selama proses penyusunan tugas akhir ini, penulis banyak mendapatkan bantuan baik secara langsung maupun tidak langsung dari berbagai pihak. Proses penyusunan yang memakan waktu cukup lama bukanlah menjadi hal yang membuang-buang waktu, melainkan menjadi pembelajaran yang sangat berharga bagi penulis dalam mencapai hasil yang maksimal. Oleh sebab itu, penulis ingin mengucapkan terima kasih kepada :

1. Keluarga besar yang selalu senantiasa memberikan kasih sayang, doa, semangat, dukungan moril dan materil.
2. Bapak Sugeng Ariyono, B.Eng, M.Eng, Ph.D selaku Direktur Politeknik Manufaktur Negeri Bangka Belitung.
3. Bapak Surojo, M.T. selaku pembimbing I yang telah meluangkan banyak waktu, memberi saran dan solusi dari masalah yang penulis hadapi selama proses penyusunan Makalah Proyek Akhir ini dan Bapak Irwan, M.Sc. selaku pembimbing II yang telah meluangkan banyak tenaga dan pikiran di dalam memberikan pengarahan pada penulisan makalah Proyek Akhir ini.
4. Seluruh staf pengajar di Politeknik Manufaktur Negeri Bangka Belitung.
5. Rekan-rekan mahasiswa Politeknik Manufaktur Negeri Bangka Belitung yang telah banyak membantu selama menyelesaikan Proyek Akhir.
6. Pihak-pihak lain yang telah memberikan bantuan secara langsung maupun tidak langsung dalam pembuatan Proyek Akhir ini yang tidak dapat disebutkan satu per satu.

Penulis menyadari bahwa penulisan makalah Proyek Akhir ini masih jauh dari sempurna dikarenakan penulis adalah manusia biasa yang tidak luput dari kesalahan. Karena yang benar hanya datang dari ALLAH dan yang salah datang dari penulis sendiri. Oleh karena itu, sangat diharapkan segala petunjuk, kritik, dan saran yang membangun dari pembaca agar dapat menunjang pengembangan dan perbaikan penulisan selanjutnya. Akhir kata penulis mohon maaf atas kekurangan dalam penulisan makalah Proyek Akhir ini dan penulis dengan senang hati menerima saran dan kritik yang membangun dari pembaca.

Semoga proyek akhir ini dapat berguna untuk menambah wawasan dan pengetahuan bagi rekan-rekan mahasiswa.

Sungailiat, Agustus 2018

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN	ii
PERNYATAAN BUKAN PLAGIAT	iii
ABSTRAK	iv
ABSTRACT	v
KATA PENGANTAR	vi
DAFTAR ISI	viii
DAFTAR TABEL	xi
DAFTAR GAMBAR	xii
DAFTAR LAMPIRAN	xiv
BAB I PENDAHULUAN	
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah dan Batasan Masalah	2
1.2.1 Rumusan Masalah	2
1.2.2 Batasan Masalah	2
1.3 Tujuan Proyek Akhir	2
BAB II DASAR TEORI	
2.1 Tinjauan Pustaka	3
2.2 Dasar Teori	5
2.2.1 Alat <i>Monitoring</i> dan Pengontrolan Penampung Air	5
2.2.2 <i>Internet of things</i> (IOT).....	5
2.2.3 Arduino Nano.....	5
2.2.4 Sensor Ultrasonik HC-SR04	6
2.2.5 Modul ESP8266	7

2.2.6 Modul <i>Relay</i> (1 Channel)	8
2.2.7 Catu Daya Adaptor	9
2.2.8 <i>LCD I2C</i>	10
2.2.9 <i>Voltage Regulator</i>	11
BAB III METODE PELAKSANAAN	
3.1 Survei, Pengumpulan data dan pengolahan data	13
3.1.1 Pengumpulan data primer	13
3.1.2 Pengumpulan data sekunder	13
3.1.3 Bahan yang digunakan	14
3.1.4 Alat yang digunakan	14
3.2 Perancangan alat <i>monitoring</i> dan pengontrolan penampung air	15
3.2.1 Perancangan <i>hardware</i> alat <i>monitoring</i> dan pengontrolan penampung air	15
3.2.2 Perancangan <i>software</i> alat <i>monitoring</i> dan pengontrolan penampung air	16
3.3 Pembuatan <i>hardware</i> dan <i>software</i>	17
3.3.1 Pembuatan <i>hardware</i> alat <i>monitoring</i> dan pengontrolan penampung air	17
3.3.2 Pembuatan <i>software</i> alat <i>monitoring</i> dan pengontrolan penampung air	18
3.4 Uji coba	18
3.5 Kesimpulan	19
BAB IV PEMBAHASAN	
4.1 Perancangan dan pembuatan <i>hardware</i> mekanik	20
4.1.1 Perancangan konstruksi	20
4.1.2 Pembuatan konstruksi	21
4.1.3 Perancangan dan pembuatan wadah	22
4.1.4 Perancangan dan pembuatan pompa	23
4.2 Perancangan dan Pembuatan <i>hardware</i> elektrik	23
4.2.1 Perancangan <i>Box Panel</i>	23
4.2.2 Pembuatan <i>Box Panel</i>	24
4.3 Adaptor	25
4.3.1 Perancangan Rangkaian Adaptor	25
4.3.2 Pembuatan Adaptor	25
4.3.3 Pengujian Adaptor	26

4.4 Modul Relay	26
4.4.1 Perancangan Modul Relay	26
4.4.2 Pembuatan Modul Relay	27
4.4.3 Pengujian Modul Relay	27
4.5 Sensor Ultrasonik HC-SR-04	28
4.5.1 Perancangan Sensor Ultrasonik HC-SR-04	28
4.5.2 Pembuatan Sensor Ultrasonik HC-SR-04	28
4.5.3 Pengujian Sensor Ultrasonik HC-SR-04.....	28
4.6 Rangkaian LCD 2x16	32
4.6.1 Perancangan Rangkaian LCD 2x16	32
4.6.2 Pembuatan Rangkaian LCD 2x16.....	33
4.6.3 Pengujian LCD 2x16.....	33
4.7 Perancangan dan Pembuatan <i>Software</i>	34
4.7.1 Flowchart program arduino.....	35
4.7.2 Pembuatan program arduino IDE.....	36
4.7.3 Perancangan Blynk	37
4.8 Pembuatan <i>interface monitoring</i> dan pengontrolan penampung air	40
4.8.1 <i>Design interface</i> aplikasi <i>monitoring</i> dan pengontrolan penampung air	40
4.8.2 Diagram <i>activity</i> program <i>Blynk</i>	42
4.9 Hasil pengujian.....	43
4.9.1 Analisa dan Hasil pengujian keseluruhan sistem	43
BAB V PENUTUP	
5.1 Kesimpulan	50
5.2 Saran	50
DAFTAR PUSTAKA	51
LAMPIRAN	

DAFTAR TABEL

Tabel 2.1 Tinjauan Pustaka	4
Tabel 3.1 Bahan yang digunakan	14
Tabel 3.2 Alat yang digunakan	14
Tabel 4.1 Hasil pengujian rangkaian Adaptor	26
Tabel 4.2 Hasil pengujian modul relay	27
Tabel 4.3 Data hasil pengukuran jarak.....	30
Tabel 4.4 Pengujian keseluruhan	44

DAFTAR GAMBAR

Gambar 2.1 Bentuk fisik arduino nano	5
Gambar 2.2 Bentuk fisik sensor ultrasonik	7
Gambar 2.3 Modul ESP8266	8
Gambar 2.4 Bentuk fisik modul <i>relay</i> 1 channel	9
Gambar 2.5 Rangkaian Adaptor	10
Gambar 2.6 Bagian-bagian <i>LCD</i>	10
Gambar 2.7 Bentuk fisik <i>LCD</i>	11
Gambar 2.8 Bentuk fisik <i>Voltage regulator</i>	11
Gambar 3.1 Diagram alir metode pelaksanaan	12
Gambar 3.2 Perancangan <i>hardware</i>	15
Gambar 4.1 Rancangan kontruksi	20
Gambar 4.2 Kontruksi	21
Gambar 4.3 Kontruksi akhir	22
Gambar 4.4 Wadah penampung air	22
Gambar 4.5 Wadah sumber utama	22
Gambar 4.6 Pompa yang digunakan	23
Gambar 4.7 Desain box	23
Gambar 4.8 <i>Hardware</i> rangkaian	25
Gambar 4.9 Rangkaian skematik Adaptor	25
Gambar 4.10 Adaptor.....	25
Gambar 4.11 Rangkaian pengujian Adaptor	26
Gambar 4.12 Rangkaian skematik modul relay	26
Gambar 4.13 Blok diagram pengujian relay	27
Gambar 4.14 Rangkaian skematik sensor ultrasonik hc-sr04	28
Gambar 4.15 Desain kontruksi	28

Gambar 4.16 Pengukuran 1	30
Gambar 4.17 Pengukuran 2	31
Gambar 4.18 Pengukuran 3.....	31
Gambar 4.19 Pengukuran 4.....	31
Gambar 4.20 Pengukuran 5	32
Gambar 4.21Rangkaian skematik LCD 2x16	32
Gambar 4.22 Layout rangkaian LCD 2x16.....	33
Gambar 4.23 LCD 2x16.....	33
Gambar 4.24 Pengujian tampilan LCD 2x16.....	34
Gambar 4.25 Diagram Blok	35
Gambar 4.26 Flowchart arduino	36
Gambar 4.27 Tampilan awal pada aplikasi <i>Blynk</i>	37
Gambar 4.28 Tampilan <i>log in</i>	38
Gambar 4.29 Penginisialan nama <i>Project</i>	38
Gambar 4.30 Tampilan pada menu <i>new project</i>	39
Gambar 4.31 Aksesoris pada <i>Widget Box</i>	39
Gambar 4.32 <i>Interface Monitoring</i> dan Pengontrolan penampung air	40
Gambar 4.33 Desain aplikasi	41
Gambar 4.34 Hasil akhir <i>Interface Aplikasi</i>	41
Gambar 4.35 Diagram <i>Activity</i> aplikasi <i>Blynk</i>	42
Gambar 4.36 Pengujian perintah auto.....	45
Gambar 4.37 Pengujian perintah auto.....	45
Gambar 4.38 Pengujian perintah manual.....	46
Gambar 4.39 Pengujian perintah manual.....	46
Gambar 4.40 Pengujian perintah auto.....	46
Gambar 4.41 Pengujian perintah auto.....	47
Gambar 4.42 Pengujian perintah manual.....	47
Gambar 4.43 Pengujian perintah manual.....	48
Gambar 4.44 Pengujian perintah auto.....	48
Gambar 4.45 Pengujian perintah auto.....	48

DAFTAR LAMPIRAN

Lampiran 1 Daftar Riwayat Hidup

Lampiran 2 Program

Lampiran 3 Data Sheet

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Air merupakan suatu kebutuhan sangat penting bagi kehidupan manusia dan makhluk hidup di bumi hingga detik ini. Terlebih lagi untuk air tawar yang tentu mengalami keterbatasan dalam penyediaannya. Tanpa air, manusia mustahil untuk bertahan hidup dalam waktu yang lama. Dikarenakan jumlah air tawar yang terus menipis serta semakin lamanya waktu musim kemarau, banyak orang-orang melakukan berbagai cara untuk menghemat pemakaiannya.

Banyaknya pekerjaan yang menyita waktu membuat masyarakat melupakan pekerjaan rumah seperti mematikan mesin pengisian air. Karena banyaknya pekerjaan yang menyita waktu ini, tidak memungkinkan masyarakat untuk menunggu selama pengisian air, oleh sebab itu masyarakat biasanya melakukan pengisian air secara manual dengan menyalakan mesin lalu ditinggal. Hal ini akan menyebabkan air pada tandon penampung akan meluap dan membuat penggunaan air menjadi sangat boros. Maka dengan permasalahan ini munculah ide untuk membuat suatu alat monitoring dan pengontrolan penampung air dengan menggunakan aplikasi *Android*. Dengan adanya alat WLC (*water level control*) ini diharapkan orang-orang tidak perlu khawatir meninggalkan rumah dalam keadaan air yang terisi pada bak penampungan. Alat ini juga akan memberikan manfaat lain, yaitu mengurangi resiko *overheating* pada mesin air yang terlalu lama dihidupkan atau lupa mematikan mesin. Dengan menggunakan aplikasi *Android* sebagai hasil laporan dan pengontrolan dari penggunaan air diharapkan alat ini dapat mengatasi permasalahan borosnya air serta bisa membuat para masyarakat dapat mengerjakan pekerjaan lain tanpa mengurangi efisiensi waktu bekerja. Alat ini juga diharapkan dapat mengatasi borosnya penggunaan air dan listrik.

1.2 Rumusan dan Batasan Masalah

1.2.1 Rumusan Masalah

Rumusan masalah yang dibuat pada proyek akhir ini adalah :

1. Bagaimana membuat alat yang bisa dikontrol serta di *monitoring* dengan menggunakan aplikasi *Android*?
2. Bagaimana prinsip kerja WLC (*water level control*) dalam menghemat air dan listrik?

1.2.2 Batasan Masalah

Proyek Akhir ini dibatasi oleh beberapa ruang lingkup permasalahan sebagai berikut :

1. Toleransi dari ketinggian level air adalah 10%.
2. Batasan jumlah air adalah 20 Liter.
3. Pengontrolan dilakukan dengan menggunakan *Android*.

1.3 Tujuan Proyek Akhir

1. Membuat alat yang bisa dikontrol serta di *monitoring* dengan menggunakan aplikasi *Android*.
2. Mengetahui prinsip kerja WLC (*water level control*) dalam menghemat air dan listrik.
3. Menerapkan kemajuan teknologi dalam kehidupan sehari-hari.

BAB II

DASAR TEORI

2.1 Tinjauan Pustaka

Mencari referensi dari beberapa sumber yang berkaitan dengan judul yang diambil. Berikut ini beberapa referensi yang berkaitan dengan penelitian:

Mengimplementasikan sistem *automatic level control* terdiri dari mikrokontroller, sensor ultrasonik, LCD, relay dan pompa air. Mikrokontroller merupakan pusat kendali dari seluruh rangkaian, dimana mikrokontroller akan mengambil data yang dikirimkan oleh sensor ultrasonik kemudian ditampilkan oleh LCD. Data yang ditampilkan oleh LCD adalah data ketinggian air. Sensor ultrasonik berfungsi sebagai pengukur ketinggian air. Relay berfungsi untuk menghidupkan dan mematikan pompa air secara otomatis yang dikendalikan mikrokontroller. [1]

Telah melakukan penelitian yang memanfaatkan elektroda logam yang didukung beberapa komponen lainnya yang berupa SCR, transistor, dan relay yang difungsikan sebagai penjaga level air didalam sebuah tandon. Mendeteksi posisi tinggi permukaan air pada tandon melalaui elektroda yang kemudian diterjemahkan oleh driver yang terdiri dari komponen utama yang berupa SCR dan transistor yang kemudian memberikan perintah kepada sistem pengendali akhir. [2]

Mengimplementasikan *IoT* kedalam *Monitoring* dan Pengontrolan Penampungan Air dengan membuat sistem dengan menggunakan mikrokontroller AV ATmega8. Sistem ini mempermudah dalam hal pemantauan kendali ketinggian air di bendungan dan sungai yang menerapkan sistem ini. Sensor ultrasonik digunakan agar ketika kondisi level air tertentu pintu air dapat dikontrol. [3]

Berdasarkan penelitian sebelumnya, telah banyak penelitian mengenai *monitoring* dan pengontrolan penampung air. Maka dilakukan

penelitian lebih lanjut perihal. *Monitoring* dan Pengontrolan Penampung Air. Berikut ini adalah Tabel 2.1 yang berisi tinjauan pustaka dari penelitian yang pernah dilakukan.

Tabel 2.1 Tinjauan Pustaka

PENELITI	JUDUL PENELITIAN	KETERANGAN
Ilfan Arifin, 2015	<i>Automatic Water Level Control</i> Berbasis Mikrokontroller Dengan Sistem Ultrasonik	Mikrokontroller yang digunakan AT Mega 328, LCD menampilkan nilai ketinggian air dalam satuan cm
Yolan Priatna, 2015	<i>Water level Control</i> Berbasis Elektroda	Penggunaan SCR, transistor dan relay yang difungsikan sebagai penjaga level air didalam sebuah tandon
Nia Anita Sari Lubis, 2017	Sistem <i>Monitoring</i> Level Ketinggian Air Bendungan	Sistem diterapkan pada bendungan air atau sungai dan mikrokontroller yang digunakan AV Atmega8
Della Devinda dan Mutiara Herdila, 2018	<i>Monitoring</i> dan Pengontrolan Penampung Air	Mikrokontroller yang digunakan adalah Arduino Nano, sensor ultrasonik digunakan sebagai pendeteksi ketinggian air, android digunakan sebagai pengontrol air dan aplikasi yang digunakan adalah blynk

2.2 Dasar Teori

2.2.1 Alat Monitoring dan Pengontrolan Penampung Air

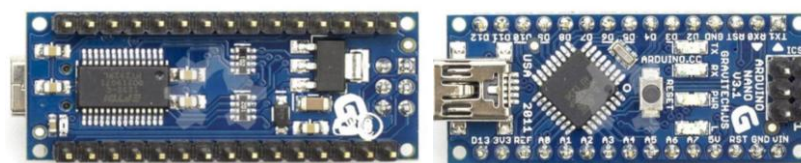
Alat monitoring dan pengontrolan penampung air merupakan suatu alat yang berfungsi sebagai pengontrol dan monitoring air, alat ini berprinsip *IoT* dimana memerlukan koneksi internet dan dilengkapi dengan fitur pengontrolan dan monitoring dengan menggunakan aplikasi *Android*. Alat ini berbasis pada arduino yang dikombinasikan dengan modul ESP8266 dan *output* yang dikeluarkan ke pompa air dan *LCD*. Sumber dari alat ini adalah *adaptor*.

2.2.2 *Internet of Things (IoT)*

Internet of Things atau *IoT* adalah sebuah istilah yang dimaksudkan dalam penggunaan internet yang lebih besar, mengadopsi komputasi yang bersifat *mobile* dan konektivitas kemudian menggabungkannya kedalam kehidupan sehari-hari. *IoT* berkaitan dengan *DoT (Disruption of Things)* dan sebagai pengantar perubahan atau transformasi penggunaan internet dari sebelumnya *Internet of People* menjadi *Internet of M2M (Maching-to-Machine)*. [4]

2.2.3 Arduino Nano

Arduino Nano adalah salah satu papan pengembangan mikrokontroler yang berukuran kecil, lengkap dan mendukung penggunaan breadboard. Arduino Nano diciptakan dengan basis mikrokontroler ATmega328 atau ATmega 168. Arduino Nano tidak menyertakan colokan DC berjenis Barrel Jack, dan dihubungkan ke komputer menggunakan port USB Mini-B. Arduino Nano dirancang dan diproduksi oleh perusahaan Gravitech. Bentuk fisik Arduino Nano bagian depan dan belakang ditunjukkan pada Gambar 2.1 dibawah ini.



Gambar 2.1 Bentuk Fisik Arduino Nano

Arduino Nano dapat diaktifkan melalui koneksi USB Mini-B, atau melalui catu daya eksternal dengan tegangan belum teregulasi antara 6-20V yang dihubungkan melalui pin 30 atau pin VIN, atau melalui catu daya eksternal dengan tegangan teregulasi 5V melalui pin 27 atau pin 5V. Sumber daya akan secara otomatis dipilih dari sumber tegangan yang lebih tinggi. *Chip* FTDI FT232L pada Arduino Nano akan aktif apabila memperoleh daya melalui USB, ketika Arduino Nano diberikan daya dari luar (*Non-USB*) maka *chip* FTDI tidak aktif dan pin 3,3V pun tidak tersedia (tidak mengeluarkan tegangan), sedangkan *LED* TX dan RX pun berkedip apabila pin digital 0 dan 1 berada pada posisi *HIGH*.

ATMega168 memiliki 16 KB flash memory untuk menyimpan kode (2 KB digunakan untuk bootloader); Sedangkan ATMega328 memiliki flash memory sebesar 32 KB, (juga dengan 2 KB digunakan untuk bootloader). ATMega168 memiliki 1 KB *memory* pada SRAM dan 512 *byte* pada EEPROM (yang dapat dibaca dan ditulis dengan perpustakaan EEPROM); Sedangkan ATMega328 memiliki 2 KB *memory* pada SRAM dan 1 KB pada EEPROM.

Masing-masing dari 14 pin digital pada Arduino Nano dapat digunakan sebagai input atau output, dengan menggunakan fungsi *pinMode()*, *digitalWrite()*, dan *digitalRead()*. Semua pin beroperasi pada tegangan 5V. Setiap pin dapat memberikan atau menerima arus maksimum 40 mA dan memiliki resistor *pull-up internal* (yang terputus secara *default*) sebesar 20-50 KOhm. [6]

2.2.4 Sensor Ultrasonik HC-SR04

Sensor ultrasonik adalah komponen yang prinsip kerjanya didasarkan prinsip dari pantulan suatu gelombang suara sehingga dapat digunakan untuk menafsirkan eksistensi sebuah benda spesifik yang ada dalam frekuensinya. Ukuran frekuensi sensor ultrasonik yaitu sekitar 40 KHz sampai 400 KHz. Sensor ultrasonik dibentuk dari dua buah unit, yaitu yang pertama adalah unit penerima dan yang kedua adalah unit pemancar. Kedua unit dalam sensor ultrasonik ini memiliki struktur yang sangatlah sederhana, yaitu suatu kristal *piezoelectric* yang terhubung dengan mekanik jangkar; disambungkan hanya dgn sebuah diafragma

penggetar. Kemudian kepada plat logam diberikan tegangan bolak balik yang mempunyai frekuensi kerja 40 KHz sampai dengan 400 KHz.

Jauh dan dekatnya benda yang terdeteksi serta kualitas dari sensor penerima ataupun sensor pemancarnya, merupakan faktor penentu besar amplitudo signal elektrik yang dihasilkan unit sensor penerimanya. Operasi *scanning* yang dijalankan oleh sensor tersebut memakai metode pantulan dengan memperhitungkan selisih jarak diantara objek sasaran dan sensor. Cara menghitung jarak tersebut ialah dengan mengalikan separuh waktu yang dipakai oleh *signal* ultrasonik untuk berjalan dari rangkaian TX hingga ditangkap kembali oleh rangkaian Rx, dengan kecepatan rambat dari signal ultrasonik tersebut pada media rambat yang dipakainya (dalam hal ini adalah udara). Waktu tersebut dihitung saat pemancar aktif hingga diperoleh adanya input dari rangkaian penerima. Apabila dalam batas waktu yang ditentukan, rangkaian penerima tak juga menerima sinyal input diartikan bahwa tak ada yang menghalangi di depannya. [7] Gambar 2.2 dibawah ini menunjukkan bentuk fisik dari sensor ultrasonik.



Gambar 2.2 Bentuk Fisik Sensor Ultrasonik

2.2.5 Modul ESP8266

ESP8266 merupakan modul *wifi* yang berfungsi sebagai perangkat tambahan *mikrokontroler* seperti Arduino agar dapat terhubung langsung dengan *wifi* dan membuat koneksi TCP/IP.

Modul ini membutuhkan daya sekitar 3.3V dengan memiliki tiga mode *wifi* yaitu *Station*, *Access Point* dan *Both* (Keduanya). Modul ini juga dilengkapi dengan prosesor, memori dan GPIO dimana jumlah pin bergantung dengan jenis ESP8266 yang kita gunakan. Sehingga modul ini bisa berdiri sendiri tanpa

menggunakan *mikrokontroler* apapun karena sudah memiliki perlengkapan layaknya *mikrokontroler*.

Firmware default yang digunakan oleh perangkat ini menggunakan *AT Command*, selain itu ada beberapa *Firmware SDK* yang digunakan oleh perangkat ini berbasis *opensource* yang diantaranya adalah sebagai berikut :

- a) *NodeMCU* dengan menggunakan *basic programming lua*
- b) *MicroPython* dengan menggunakan *basic programming python*
- c) *AT Command* dengan menggunakan perintah perintah *AT command*

Untuk pemrogramannya sendiri bisa menggunakan *ESPlorer* untuk *Firmware* berbasis *NodeMCU* dan menggunakan *putty* sebagai terminal *control* untuk *AT Command*.

Selain itu perangkat ini bisa diprogram menggunakan *Arduino IDE*. Dengan menambahkan *library ESP8266* pada *board manager* kita dapat dengan mudah memprogram dengan *basic* program *arduino*. Ditambah lagi dengan harga yang cukup terjangkau, modul ini sering digunakan untuk membuat berbagai proyek. Bentuk fisik dari *Node MCU* terdapat pada Gambar 2.3 dibawah ini.



Gambar 2.3 Bentuk Fisik Modul ESP8266

2.2.6 Modul Relay

Terdapat relay pada modul ini, dengan tegangan 3.3V/5V DC untuk melakukan kontrol terhadap relay. Sangat baik untuk melakukan *switch* pada perangkat AC maupun DC yang membutuhkan arus dan tegangan yang besar.

Relay yang digunakan ialah SPDT atau *single pole double throw* dengan arus dan tegangan maksimal 10A/250V AC. *Ground* pada koil *relay* terpisah dengan *ground* pada sinyal input. Namun keduanya dapat disatukan dengan memberikan jumper pada header.

Features:

Power supply range from 5V~7.5V

Onboard Photocoupler isolation

Equiped with high-current relay, AC250V 10A ; DC30V 10A.

Relay Output Indicator LED

TTL logic interface can be directly connected to microcontroller (Arduino, 8051, AVR, PIC, DSP, ARM, ARM, MSP430,etc). [5]

Bentuk fisik modul relay ditunjukkan pada Gambar 2.4 dibawah ini.



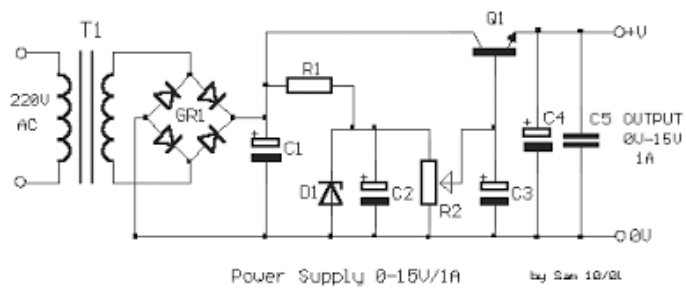
Gambar 2.4 Bentuk Fisik Modul Relay

2.2.7 Catu Daya Adaptor

Adaptor adalah perangkat elektronik yang dapat merubah tegangan listrik (AC) yang tinggi menjadi tegangan listrik (DC) yang rendah, tetapi ada juga adaptor yang dapat merubah tegangan listrik yang rendah menjadi tegangan listrik yang tinggi. Adaptor, *accumulator* (aki), dan baterai merupakan salah satu contoh penyuplai daya (Power supply). Keuntungan dari adaptor dibanding dengan baterai maupun *accumulator* adalah sangat praktis berhubungan dengan ketersediaan tegangan, karena adaptor dapat di ambil dari sumber tegangan AC yang ada di rumah, dimana pada zaman sekarang ini setiap rumah sudah menggunakan listrik. Selain itu, adaptor mempunyai jangka waktu yang tidak terbatas jika ada tegangan AC, tegangan AC ini sudah merupakan kebutuhan primer dalam kehidupan manusia.

Secara umum adaptor adalah alat elektronika yang dapat menyesuaikan atau merubah tegangan listrik, maksudnya adalah merubah sumber tegangan listrik utama yaitu dari PLN menjadi tegangan listrik yang dapat digunakan untuk disesuaikan dengan perangkat elektronik yang akan dipakai, misalnya seperti Televisi, Radio, gadget dan lain lain.

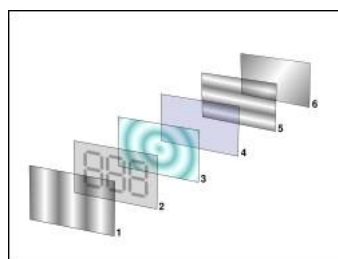
Rangkaian Adaptor adalah rangkaian yang berfungsi untuk mengubah arus AC menjadi arus DC. Kelebihan dari rangkaian ini adalah arus yang dihasilkan cukup stabil dan besarnya tegangan yang dihasilkan bisa kita atur dengan cara menyesuaikan komponen yang digunakan dengan output tegangan yang kita kehendaki. Adaptor banyak digunakan dalam berbagai alat sebagai catu daya, seperti Amplifier, TV mini, radio, tape dan lain-lain. [8] Gambar rangkaian adaptor ditunjukkan pada Gambar 2.5 dibawah ini.



Gambar 2.5 Rangkaian Adaptor

2.2.8 LCD I2C

Liquid Crystal Display (LCD) adalah sebuah peralatan elektronik yang berfungsi untuk menampilkan output sebuah sistem dengan cara membentuk suatu citra atau gambaran pada sebuah layar. Secara garis besar komponen penyusun LCD terdiri dari kristal cair (*liquid crystal*) yang diapit oleh 2 buah elektroda transparan dan 2 buah filter polarisasi (*polarizing filter*). [9] Bagian-bagian pada LCD ditunjukkan pada Gambar 2.6 dan bentuk fisik LCD ditunjukkan pada Gambar 2.7 dibawah ini.



Gambar 2.6 Bagian-Bagian LCD

Keterangan:

1. Film dengan *polarizing filter vertical* untuk memolarisasi cahaya yang masuk.
2. *Glass substrate* yang berisi kolom-kolom elektroda *Indium tin oxide* (ITO).
3. *Twisted nematic liquid crystal* (kristal cair dengan susunan terpilin).
4. *Glass substrate* yang berisi baris-baris elektroda *Indium tin oxide* (ITO).
5. Film dengan *polarizing filter horizontal*
6. Reflektor cahaya

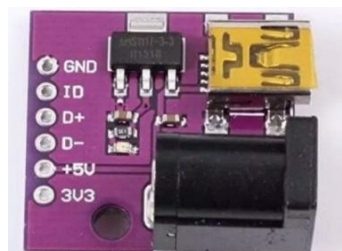


Gambar 2.7 Bentuk Fisik LCD

2.2.9 Voltage Regulator

Voltage regulator atau dalam bahasa Indonesianya pengatur tegangan merupakan rangkaian elektronika yang berfungsi mempertahankan atau memastikan tegangan pada nilai tertentu secara otomatis.

Tentunya tegangan yang stabil dan bebas gangguan tersebut sangat dibutuhkan untuk mengoperasikan peralatan elektronik yang berbasis digital seperti mikroprosesor ataupun mikrokontrol. Tegangan output yang ada pada *voltage regulator* tidak dipengaruhi oleh perubahan tegangan input, beban, maupun suhu. Gambar 2.8 menunjukkan bentuk fisik dari *voltage regulator*.

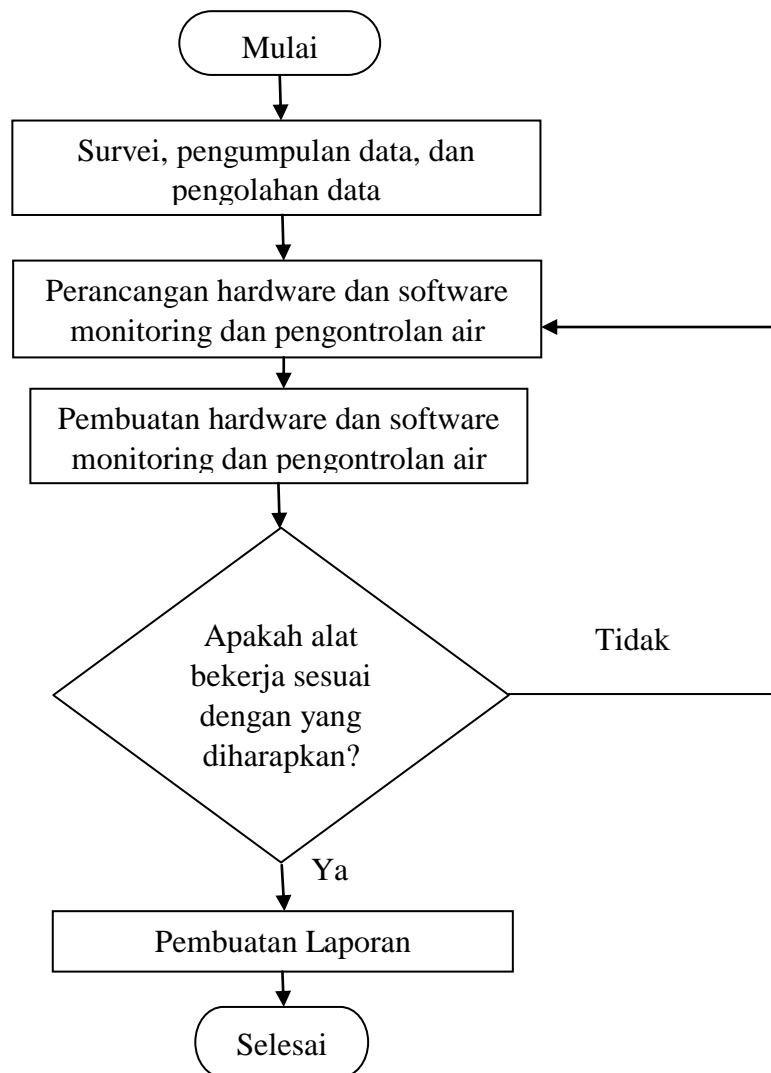


Gambar 2.8 Bentuk Fisik Voltage Regulator

BAB III

METODE PELAKSANAAN

Metode pelaksanaan yang digunakan dalam proyek akhir ini adalah dengan merancang kegiatan-kegiatan dalam bentuk diagram alir, dengan tujuan agar tindakan yang dilakukan lebih terarah dan terkontrol sehingga target-target yang diharapkan dapat tercapai. Diagram alir ini dapat dilihat pada Gambar berikut.



Gambar 3.1 Diagram Alir Metode Pelaksanaan

Dari gambar 3.1 dapat dijelaskan metode penelitian proyek akhir ini meliputi beberapa data sebagai berikut:

3.1 Survei, Pengumpulan Data dan Pengolahan Data

Pelaksanaan proyek akhir ini dimulai dengan melakukan survei, pengumpulan data, dan pengolahan data. Hal ini bertujuan untuk mendapatkan ide permasalahan yang dapat dijadikan rumusan permasalahan dalam merancang proyek akhir ini. Data yang kami kumpulkan dapat berupa data primer maupun data sekunder. Berikut ini adalah metode pengumpulan data secara primer dan sekunder tersebut, yaitu:

3.1.1 Pengumpulan Data Primer

Pengumpulan data primer adalah data didapatkan secara langsung dari objek penelitian. Metode ini dilakukan dengan cara melakukan bimbingan / konsultasi dengan pihak-pihak lain agar tujuan yang diterapkan pada proyek akhir tercapai sesuai keinginan. Tujuannya adalah untuk mengkoordinasikan rancangan yang telah disetujui untuk dibuat, juga untuk mengatasi permasalahan yang timbul selama proses pembuatan proyek akhir, termasuk pembuatan laporan dan presentasi proyek akhir.

3.1.2 Pengumpulan Data Sekunder

Pengumpulan data sekunder adalah data yang secara tidak langsung dari objek penelitian. Metode ini dilakukan dengan cara pengumpulan data dari internet yang berhubungan dengan proyek akhir. Tujuannya adalah untuk mendapatkan daftar peralatan dan komponen-komponen yang tepat untuk digunakan pada proyek akhir yang akan dibuat. Selain itu juga untuk membandingkan literatur dari berbagai sumber yang berasal dari buku-buku referensi dan jurnal yang akan berkaitan dengan proyek akhir dari internet dengan rancangan proyek akhir yang akan dibuat. Tujuannya adalah untuk mengetahui kemungkinan rancangan yang dapat diajukan sebagai proposal proyek akhir.

3.1.3 Bahan yang digunakan

Berikut adalah bahan yang digunakan dan dirangkum pada tabel 3.1 dibawah ini :

Tabel 3.1 Bahan yang digunakan

No.	Bahan	Kuantitas
1.	Arduino Nano	1
2.	Sensor ultrasonik	1
3.	Modul ESP8266	1
4.	Modul relay	1
5.	Adaptor DC 5V	1
6.	<i>LCD I2C</i>	1
7.	<i>Voltage Regulator</i>	1
8.	Pompa	1
9.	Selang	1
10.	Wadah	1

3.1.4 Alat yang digunakan

Berikut adalah alat yang digunakan, terdiri dari perangkat lunak, alat ukur dan alat perkakas yang dirangkum pada tabel 3.2 dibawah ini :

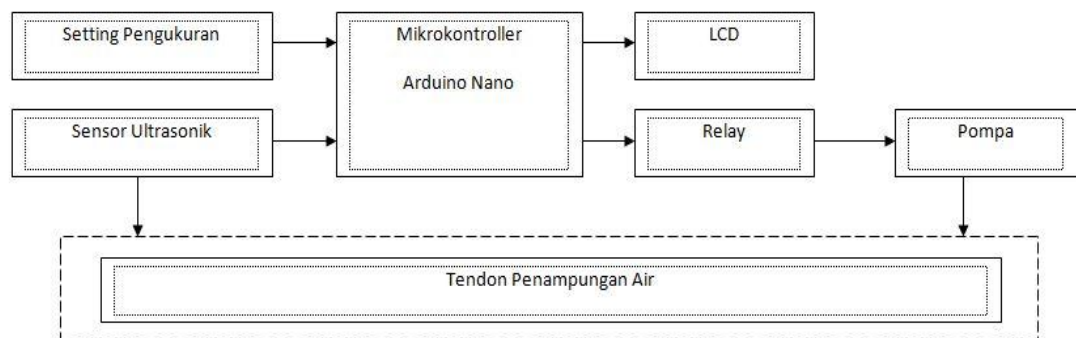
Tabel 3.2 Alat yang digunakan

No.	Jenis Alat	Nama
1.	Perangkat Lunak	Aplikasi Blynk Arduino IDE 1.6.12 Solder
2.	Alat Perkakas	Timah Solder Cutter Obeng
3.	Catu Daya	Adaptor

3.2 Perancangan alat monitoring dan pengontrolan penampung air

3.2.1 Perancangan *Hardware* alat *monitoring* dan pengontrolan penampung air

Perancangan *hardware* bertujuan agar dapat mencapai target yang diinginkan dan semua sistem kontrol dapat bekerja dengan baik. Bertujuan juga untuk menghubungkan antara komponen dengan *mikrokontroler* dan *packaging* yang melindungi *mikrokontroler* dan komponen didalamnya. Adaptor 5V digunakan sebagai daya untuk keseluruhan sistem. Terdapat 1 inputan yaitu sensor ultrasonik kemudian akan diproses di *Arduino Uno*. Output dari *hardware* ini adalah relay, pompa dan lcd 12c. Diagram blok *Hardware* ditunjukkan pada gambar 3.2 dibawah ini.



Gambar 3.2 Perancangan *Hardware* [10]

Adapun perancangan *hardware* sebagai berikut:

1. Perancangan *hardware mechanical*

Perancangan *hardware mechanical* terdiri dari:

a) Perancangan konstruksi penampung air (tendon)

Konstruksi penampung air berguna untuk sebagai wadah air yang telah dikombinasikan dengan *electrical wiring* yang aman dengan menggunakan prinsip isolator yang baik.

b) Perancangan *box* panel kontrol

Box panel kontrol terbuat dengan berbahan dasar plastik berukuran 10cmx10cmx5cm yang dimodifikasi sedemikian rupa sehingga sesuai dengan kebutuhan.

2. Perancangan *hardware electrical*

Perancangan *hardware electrical* terdiri dari:

a) Adaptor

Adaptor yang digunakan memiliki spesifikasi DC 5V/3A.

b) Arduino

Arduino yang digunakan yaitu Arduino Nano.

c) Pompa air

Pompa air yang digunakan adalah pompa air DC yang memiliki spesifikasi yang cukup.

d) Sensor Ultrasonik

Sensor ultrasonik yang digunakan sensor ultrasonic HC-SR04.

e) LCD 16x2

LCD 16x2 digunakan sebagai tampilan *interface*.

f) *Voltage Regulator*

Voltage Regulator 3,3V.

Perancangan *hardware* bertujuan untuk menghubungkan antara komponen dengan arduino. Adaptor 5V digunakan sebagai daya untuk keseluruhan sistem.

3.2.2 Perancangan *Software alat monitoring dan pengontrolan penampung air*

Setelah perancangan *hardware* selesai, maka langkah selanjutnya adalah merancang pemilihan *software* yang sesuai dengan rancangan *hardware*. Perancangan *software* ini bertujuan untuk mengontrol sistem kerja alat. *Software* yang digunakan pada proyek akhir ini adalah *Arduino IDE Version 1.6.12*. Diagram blok keseluruhan sistem terdiri dari Input, mikrokontroler dan Output.

Adapun *software* ini berfungsi untuk mengontrol berbagai komponen dalam monitoring dan pengontrolan penampungan air, yaitu sebagai berikut:

a) Modul ESP

Modul ESP digunakan sebagai penghubung sistem android dengan server arduino pada koneksi internet.

b) Arduino Nano

Arduino Nano digunakan sebagai otak dan pengolah data input output dan pengatur komunikasi modul ESP.

c) Relay

Relay digunakan untuk menghidupkan dan mematikan pompa.

d) Pompa Air

Pompa Air digunakan sebagai alat yang memompa pergerakan air dari sumber ke wadah penampungan.

e) Sensor Ultrasonik

Sensor Ultrasonik digunakan untuk mendeteksi ketinggian air.

f) *LCD I2C*

LCD I2C digunakan untuk menampilkan data.

g) *Voltage Regulator*

Voltage Regulator digunakan untuk mengeluarkan tegangan 3,3V yang dibutuhkan modul ESP.

3.3 Pembuatan *Hardware* dan *Software*

3.3.1 Pembuatan *Hardware* alat *monitoring* dan pengontrolan penampung air

Setelah membuat suatu rancangan alat, maka selanjutnya dibuat alat sesuai dengan desain yang telah ditentukan sebelumnya. Adapun pembuatan *hardware* sebagai berikut:

1. Pembuatan *hardware mechanical*

Pembuatan *hardware mechanical* dilakukan dengan cara menyediakan berbagai peralatan yang diperlukan kemudian membuat bagian-bagian alat satu per satu sesuai dengan perancangan yang telah ditentukan sebelumnya.

2. Pembuatan *hardware electrical*

Adapun hal-hal yang dilakukan dalam membuat *hardware electrical* yaitu:

- Menyediakan semua peralatan dan komponen-komponen elektronika yang akan digunakan untuk membuat alat ini.

- Setelah semua peralatan dan komponen tersedia selanjutnya merangkai komponen elektronika seperti *LCD* dan modul *ESP*. Berikut adalah komponen-komponen elektronika yang akan dipasang pada *box* panel kontrol yaitu:
 - Adaptor
 - Arduino Nano
 - Modul ESP
 - Relay
 - *LCD* 16x4
 - Saklar *On-Off*
 - *Power Suply*
- *Assembly* bagian-bagian tersebut menjadi satu-kesatuan rangkaian kontrol alat pengontrolan dan *monitoring* air.

3.3.2 Pembuatan *Software* alat *monitoring* dan pengontrolan penampung air

Setelah pembuatan *hardware monitoring* dan pengontrolan penampung air sesuai dengan desain yang telah ditentukan sebelumnya. Selanjutnya dilakukan pembuatan *software* alat *monitoring* dan pengontrolan penampung air. Pembuatan *software*, meliputi:

- Menyediakan *software Arduino IDE Version 1.6.12*.
- Menginstal *software Arduino IDE Version 1.6.12* ke dalam komputer.
- Membuat program koneksi internet dari hubungan arduino ke Andorid.
- Menginstal *software Blynk* di android.
- Membuat program Koneksi ke server *Blynk*.

3.4 Uji Coba

Pada tahap selanjutnya adalah tahap uji coba alat yang sudah dibuat sebelumnya. Dalam suatu uji coba alat biasanya mengalami *trial and error*. Untuk itu *monitoring* dan pengontrolan penampung air dan program yang sudah dibuat harus diuji cobakan agar proses kerja yang diinginkan tercapai dan mendapatkan hasil pengukuran yang tepat. Apabila dalam uji coba mengalami gangguan (*error*)

dan tidak bekerja. sesuai yang diinginkan maka proses selanjutnya adalah perbaikan pada sistem yang mengalami gangguan tersebut.

Adapun uji coba yang dilakukan terbagi dengan 2 cara yaitu:

- Uji Coba Perbagian
 - a. Uji coba *LCD* apakah berfungsi atau tidak.
 - b. Uji coba Sensor Ultrasonik.

- Uji Coba Keseluruhan

Uji coba keseluruhan dari alat ini adalah untuk mendapatkan sistem yang baik. Pengujian secara keseluruhan dilihat dari tingkat toleransi ketinggian air dan sistem jaringan komunikasi internet.

Dari hasil uji coba akan diketahui apakah proses kerja dari alat ini telah sesuai dengan instruksi yang sudah dirancang. Jika sudah sesuai maka proses pembuatan proyek akhir ini akan berlanjut dengan pembuatan laporan, namun jika belum sesuai maka dilakukan lagi analisa dari perancangan alat sampai didapatkan hasil yang diinginkan.

3.5 Kesimpulan

Berdasarkan hasil dari uji coba yang telah dilakukan dapat ditarik kesimpulan bahwa pada tahap ini semua aktivitas selama pembuatan proyek akhir dapat dilaporkan dalam bentuk makalah proyek akhir. Laporan ini dibuat berdasarkan format yang telah ditentukan sesuai dengan panduan yang sudah ada, yang berisikan mulai dari latar belakang masalah, perancangan alat proyek akhir berupa kontruksi dari besi untuk penempatan wadah penampung air, pembuatan alat proyek akhir, pengujian kerja alat *monitoring* dan pengontrolan penampung air yang telah dibuat, petunjuk penggunaan dan hasil pencapaiannya, termasuk kekurangan / kesalahan (*error*) yang mungkin didapat sebagai hasil uji coba kerja alat *monitoring* dan pengontrolan penampung air.

BAB IV PEMBAHASAN

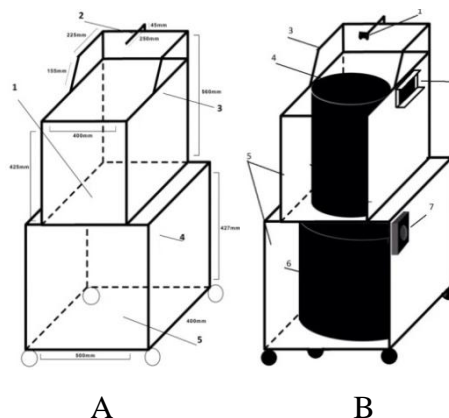
Bab ini akan membahas mengenai teknik rancangan, hasil dan analisis pengujian yang telah dilakukan serta observasi. Pengujian dilakukan untuk mengetahui performa dari alat yang telah dibuat. Adapun pengujian yang dilakukan adalah sebagai berikut:

- Pengujian Modul Relay.
- Pengujian Sensor Ultrasonik.
- Pengujian LCD.
- Pengujian secara keseluruhan.

4.1 Perancangan dan Pembuatan *Hardware* Mekanik

4.1.1 Perancangan Kontruksi

Perancangan kontruksi penampung air ini dibuat menjadi dua bagian yaitu bagian atas kontruksi digunakan untuk menempatkan wadah penampung air dan pada bagian bawah sebagai tempat ember untuk sumber air utama. Pada samping kiri atas terdapat *box* panel dan besi penyangga untuk meletakkan sensor ultrasonik. Bagian atas dan bawah kontruksi berbentuk balok yang disusun bertingkat. Untuk menunjang kontruksi digunakan roda sebanyak empat buah untuk memudahkan pemindahan kontruksi.



Gambar 4.1 Rancangan Kontruksi

Berikut merupakan penjelasan mengenai rancangan konstruksi gambar 4.1 A panel pada gambar diatas:

1. Tempat peletakan wadah tandon penampung
2. Tempat peletakkan sensor ultrasonik
3. Tempat peletakkan *box* panel kontrol
4. Tempat peletakkan terminal
5. Tempat peletakkan bak utama sumber air

Berikut merupakan penjelasan mengenai rancangan konstruksi gambar 4.1 B panel pada gambar diatas:

1. Sensor Ultrasonik
2. *Box* panel kontrol
3. Besi pegangan konstruksi
4. Tandon penampung air
5. Triplek penutup sisi-sisi konstruksi
6. Bak utama sumber air
7. Terminal

4.1.2 Pembuatan Kontruksi

Kontruksi dibuat dari besi holo dan dinding penutup terbuat dari triplek milamin, dengan ukuran kontruksi bawah panjang 500 mm, lebar 400 mm dan tinggi 427 mm, bagian atas berukuran panjang 400 mm, lebar 400 mm dan tinggi 425 mm. Pegangan pada bagian kiri dan kanan atas berukuran panjang 225 mm, lebar 155 mm ditunjukkan pada Gambar 4.2.



Gambar 4.2 Kontruksi

Berikut merupakan kontruksi akhir alat pada proyek akhir seperti yang ditunjukkan pada Gambar 4.3 dibawah ini.



Gambar 4.3 Kontruksi Akhir

4.1.3 Perancangan dan Pembuatan Wadah

Pada perancangan ini terdapat dua buah wadah yang digunakan. Wadah pertama digunakan untuk tempat penampung air berbahan plastik dan berbentuk tabung dengan ukuran tinggi 42 cm dan diameter 30 cm. Wadah ini bisa menampung air dengan kapasitas 27,5 liter ditunjukkan pada Gambar 4.4.



Gambar 4.4 Wadah Penampung Air

Wadah kedua ini digunakan sebagai *prototipe* tedmon yang berdiameter 35 cm dan tinggi 33 cm sebagai bak sumber utama dan tempat saluran pompa air ditunjukkan pada Gambar 4.5.



Gambar 4.5 Wadah Sumber Utama

4.1.4 Perancangan dan Pembuatan Pompa

Pompa yang digunakan adalah pompa yang biasa dipakai pada akuarium. Ukuran pompa 12 x 6.5 cm. Pompa digunakan sebagai alat yang memompa pergerakan air dari sumber utama ke wadah penampung, Pompa ini juga dilengkapi dengan *driver* pompa sebagai pengontrolan pompa dari arduino. Pompa yang digunakan ditunjukkan pada Gambar 4.6 dibawah ini:

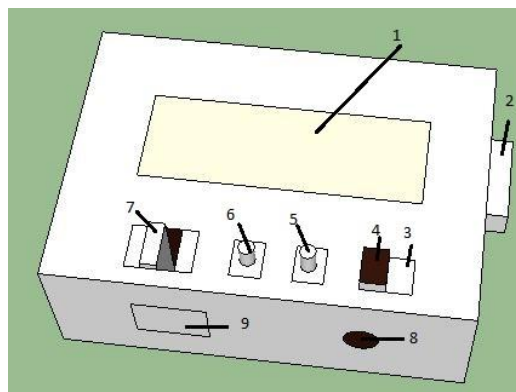


Gambar 4.6 Pompa yang Digunakan

4.2 Perancangan dan Pembuatan Hardware Elektrik

4.2.1 Perancangan *Box* Panel

Box digunakan untuk meletakkan *lcd*, saklar, tombol on/off, dan dibagian samping kanan terdapat colokan untuk sensor ultrasonik. Selain itu semua komponen diletakkan di dalam *box* panel tersebut. *Casing box* ini terbuat dari bahan atom plastik dengan panjang 10 cm, lebar 7.7 cm dan tinggi 3.5 cm. Untuk bagian bawah terdapat slot untuk kabel USB berukuran 3 cm x 1.5 cm dan slot untuk kabel adaptor. Gambar 4.7 ini merupakan desain *box* panel yang digunakan.



Gambar 4.7 Desain *Box*

Berikut merupakan penjelasan mengenai keterangan box panel pada gambar diatas:

1. *LCD*
2. Colokan untuk sensor ultrasonik
3. Tombol OFF
4. Tombol ON
5. Tombol ON/OFF manual
6. Tombol mode
7. ON/OFF layar *LCD*
8. Slot untuk kabel adaptor
9. Slot untuk kabel USB

4.2.2 Pembuatan *Box Panel*

Bagian ini akan dibahas mengenai perangkat keras yang digunakan untuk mengirim informasi berupa status ketinggian pompa dan status kondisi pompa. Perancangan perangkat keras (*hardware*) ini berupa penyusunan komponen elektronik menjadi sebuah rangkaian yang dapat bekerja sesuai dengan fungsinya. Hal yang dilakukan dalam pengujian yaitu menyambungkan alat ke *power supply* sebagai sumber keseluruhan sistem. *Led* pada modul relay dan *Voltage Regulator* akan menyala sebagai indikator alat sudah mendapatkan daya. Ketika *mode* auto diaktifkan maka pengguna bisa memilih batas maksimum dan minimum air yang ingin diisi namun pada *mode* manual pengguna melakukan kontrol sendiri dengan menekan *button on/off* pada aplikasi, kemudian data tersebut akan dikirimkan ke *server blynk* dan selanjutnya dari *web server* akan dikirimkan ke ESP8266 sebagai *broker*. Di dalam proses pengiriman data ini, ESP8266 harus memiliki akses point dan koneksi internet yang stabil. Setelah data diterima ESP8266 maka selanjutnya sinyal akan dikirimkan ke Arduino Nano sebagai elemen proses dari alat ini. Lalu Arduino akan memberikan sinyal *driver* pompa dan *driver* pompa akan langsung mengontrol kapan pompa harus hidup dan pompa harus mati. Informasi ketinggian air akan ditampilkan pada *interface monitoring* dan pengontrolan penampung air.

Berikut merupakan gambar *hardware* rangkaian keseluruhan yang ditunjukkan pada Gambar 4.8 dibawah ini.

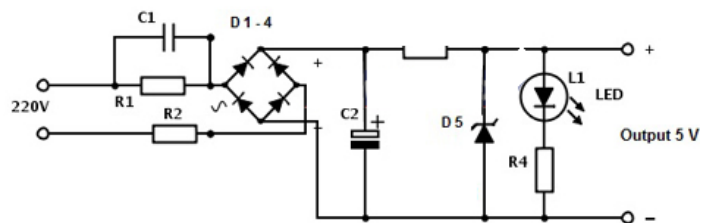


Gambar 4.8 *Hardware* Rangkaian

4.3 Adaptor

4.3.1 Perancangan Rangkaian Adaptor

Pada proyek akhir ini rangkaian catu daya yang digunakan adalah rangkaian adaptor. Tegangan output yang diinginkan yaitu 5V DC untuk Mikrokontroler Arduino Nano dan komponen lainnya seperti sensor Ultrasonik HC-04 dan Relay. Adapun skematik dari rangkaian catu daya yang digunakan ditunjukkan pada gambar 4.9 berikut.



Gambar 4.9 Rangkaian Skematik Adaptor

4.3.2 Pembuatan Adaptor

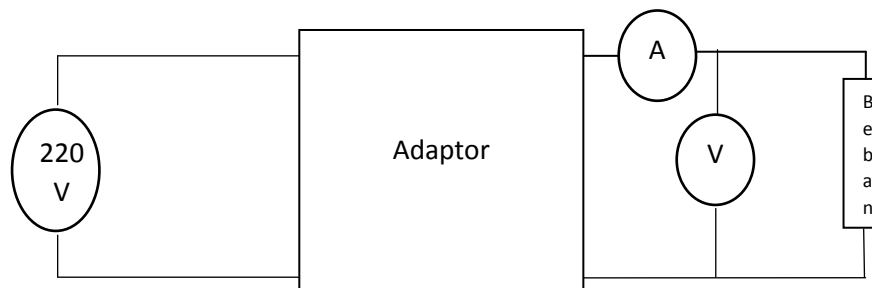
Uji coba rangkaian Adaptor dilakukan untuk menguji apakah rangkaian Adaptor yang dibuat sesuai dengan yang diinginkan.



Gambar 4.10 Adaptor

4.3.3 Pengujian Adaptor

Uji coba rangkaian Adaptor dilakukan untuk menguji apakah rangkaian Adaptor yang dibuat sesuai dengan yang diinginkan. Berikut adalah skema dalam pengujian Adaptor yang ditunjukkan pada gambar 4.11.



Gambar 4.11 Rangkaian Pengujian Adaptor

Berikut merupakan hasil pengujian rangkaian Adaptor yang ditunjukkan pada tabel 4.1 di bawah ini.

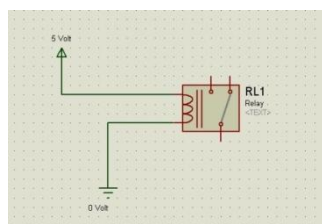
Tabel 4.1 Hasil Pengujian Rangkaian Adaptor

Perancangan	Hasil Pengukuran	Persentase Error
5V DC	5V DC	0%
3A	3A	0%

4.4 Modul Relay

4.4.1 Perancangan Modul Relay

Perancangan Modul Relay dilakukan untuk mengetahui gambaran mengenai cara pemakaian modul relay. Berikut adalah gambar skematik dari rangkaian modul relay menggunakan software ISIS Proteus ditunjukkan pada gambar 4.12.



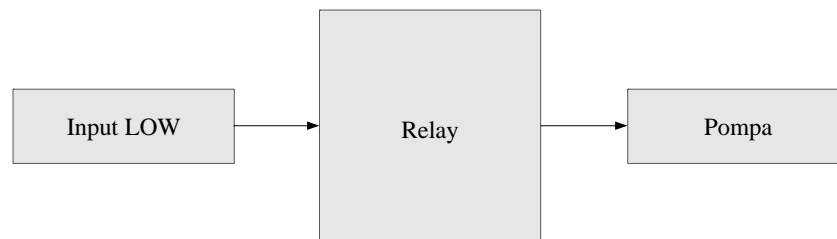
Gambar 4.12 Rangkaian skematik modul relay

4.4.2 Pembuatan Modul Relay

Pembuatan Modul Relay diputuskan dengan membeli modul yang telah jadi dan biasa dijual secara umum. Hal ini dikarenakan modul yang dijual secara umum dapat langsung digunakan tanpa harus membuat rangkaian modul relay.

4.4.3 Pengujian Modul Relay

Pengujian relay dilakukan dengan menyalakan LED pada modul relay dan juga pengujian terminal Outputan yang berupa saklar NO (*Normally Open*) dan NC (*Normally Close*). Pada saat pengujian modul relay di coba dengan mengaktifkan lampu relay dan hasilnya LED dari relay menyala serta terminal outputannya bekerja sesuai dengan fungsinya masing masing. Berikut blok diagram pengujian masing-masing koil pada relay terdapat pada gambar 4.13.



Gambar 4.13 Blok diagram pengujian relay

Berdasarkan pengujian didapati hasil sebagaimana terdapat pada tabel 4.2. Dari data dapat dianalisa koil pada modul relay berfungsi dengan baik. Ditunjukkan lampu dan koli berpindah ketika diberikan inputan dari arduino.

Seperti kita ketahui relay menyala dalam kondisi LOW sehingga ketika Arduino yang mana telah diprogram memberikan inputan low pada keluaran yang terhubung ke channel pada relay maka relay yang diberikan inputan akan menyala atau bekerja. Sehingga dapat digunakan dalam proyek akhir ini.

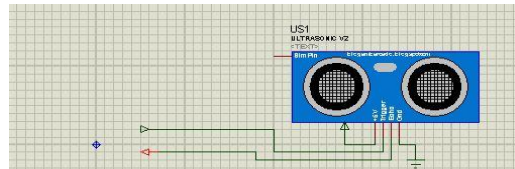
Tabel 4.2 Hasil Pengujian Modul Relay

Channel Relay	Input	Hasil Pengujian	Kondisi
1	LOW	LED Menyala	Baik

4.5 Sensor Ultrasonik HC-SR04

4.5.1 Perancangan Sensor Ultrasonik HC-SR04

Sensor Ultrasonik HC-SR04 digunakan sebagai pendeteksi ketinggian air. Adapun skematik dari rangkaian Ultrasonik yang digunakan ditunjukkan pada gambar 4.14 berikut.



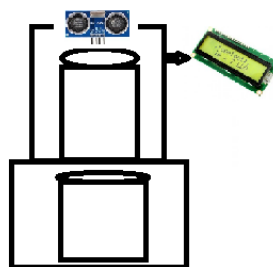
Gambar 4.14 Rangkaian Skematik Sensor Ultrasonik HC-SR04

4.5.2 Pembuatan Sensor Ultrasonik HC-SR04

Pembuatan sensor Ultrasonik HC-SR04 diputuskan dengan membeli modul yang telah jadi dan biasa dijual secara umum. Hal ini dikarenakan sensor Ultrasonik HC-SR04 yang dijual secara umum sudah dapat langsung digunakan tanpa perlu membuat rangkaian tambahan.

4.5.3 Pengujian Sensor Ultrasonik

Sensor ultrasonik bekerja berdasarkan perbandingan dari berapa waktu yang ditangkap setelah gelombang itu dipancarkan. Semakin jauh benda maka waktu pantulan akan semakin lama sedangkan jika semakin dekat benda maka pantulan akan semakin cepat. Gambar 4.15 menunjukkan desain konstruksi dan penempatan sensor ultrasonik pada prototype. Sensor ultrasonik diletakkan diatas wadah penampung air.



Gambar 4.15 Desain Kontruksi

Adapun *coding* program yang dibuat sebagai pengujian sensor ultrasonik untuk memastikan sensor tersebut bekerja sesuai dengan yang diinginkan, ialah sebagai berikut :

```
#include <NewPing.h>
#define TRIGGER_PIN 12
#define ECHO_PIN 11
#define MAX_DISTANCE 42
NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
int tinggi = 0;
int tinggi2 = 0;
void lcdTampil()
{
    tinggi = map(sonar.ping_cm(), 11, MAX_DISTANCE, 28, 0);
    if (tinggi < MAX_DISTANCE)
        tinggi2=tinggi;
    char lcd1[20];
    sprintf(lcd1, "level=%d cm ", tinggi2);
    lcd.setCursor(0, 0);
    lcd.print(lcd1);
}
void setup()
{
    lcd.init();
    lcd.backlight();
}
void loop()
{
    lcdTampil();
    delay(200);
}
```

Tabel 4.3 menunjukkan data hasil pengukuran jarak menggunakan sensor ultrasonik pada bidang datar. Hasil dari pengukuran sensor ultrasonik akan ditampilkan dalam bentuk persentase pada *LCD*.

Tabel 4.3 Data Hasil Pengukuran Jarak

No	Jarak yang diukur secara manual	Jarak yang diukur oleh sensor	Error (%)
1	11,8cm	11cm	0,8
2	23cm	23cm	0
3	29,5cm	29cm	0,5
4	17,9cm	18cm	0,1
5	17cm	17cm	0
Rata-Rata Error (%)			0,28

Berikut merupakan hasil pengukuran pertama yang telah dilakukan seperti yang ditunjukkan pada Gambar 4.16.



Gambar 4.16 Pengukuran 1

Berikut merupakan hasil pengukuran kedua yang telah dilakukan seperti yang ditunjukkan pada Gambar 4.17.



Gambar 4.17 Pengukuran 2

Berikut merupakan hasil pengukuran ketiga yang telah dilakukan seperti yang ditunjukkan pada Gambar 4.18.



Gambar 4.18 Pengukuran 3

Berikut merupakan hasil pengukuran keempat yang telah dilakukan seperti yang ditunjukkan pada Gambar 4.19.



Gambar 4.19 Pengukuran 4

Berikut merupakan hasil pengukuran kelima yang telah dilakukan seperti yang ditunjukkan pada Gambar 4.20.



Gambar 4.20 Pengukuran 5

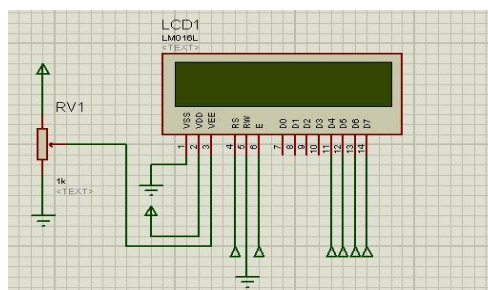
Berdasarkan hasil pengujian yang telah dilakukan dapat diketahui bahwa error dapat dihitung dengan rumus :

$$|Error(\%)| = \text{Pengukuran manual} - \text{Pembacaan sensor} \dots\dots\dots[1]$$

4.6 Rangkaian LCD 2x16

4.6.1 Perancangan Rangkaian LCD 2x16

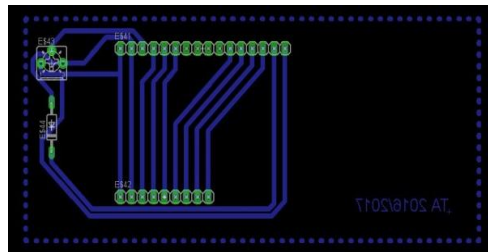
Perancangan dimulai dengan proses pembuatan skematik rangkaian LCD 2x16 menggunakan *software* ISIS Proteus. LCD 2x16 berfungsi sebagai penunjuk *display* perintah. Gambar skematik dari rangkaian LCD 2x16 ditunjukkan pada gambar 4.21 berikut.



Gambar 4.21 Rangkaian Skematik LCD 2x16

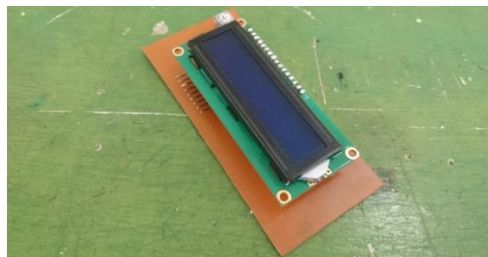
4.6.2 Pembuatan Rangkaian LCD 2x16

Proses pembuatan rangkaian LCD 2x16 ini dilakukan dengan membuat *layout* rangkaian LCD 2x16 sesuai dengan skematik yang telah dibuat. Berikut adalah *layout* dari rangkaian LCD 2x16 ditunjukkan pada gambar 4.22.



Gambar 4.22 *Layout* Rangkaian LCD 2x16

Adapun rangkaian LCD 2x16 yang dibuat ditunjukkan pada gambar 4.23 berikut.



Gambar 4.23 LCD 2x16

4.6.3 Pengujian LCD 2x16

Pengujian *LCD* dilakukan dengan tujuan untuk mendapatkan parameter berupa tampilan karakter pada *LCD* sesuai dengan keinginan. Pengujian ini dilakukan dengan memprogram karakter atau tulisan yang ingin ditampilkan pada *LCD* dan kemudian dicocokkan dengan tampilan yang ada pada layar *LCD* tersebut. Adapun *coding* program yang dibuat sebagai pengujian *LCD* untuk memastikan *LCD* bekerja sesuai dengan yang diinginkan, ialah sebagai berikut :

```
void setup()
{
  lcd.init();
  lcd.init();
  lcd.backlight();
}
```



```

lcd.setCursor(1,0);
lcd.print("Mutiara herdila");
lcd.setCursor(2,1);
lcd.print("della devinda");
lcd.setCursor(0,2);
}
void loop()
{
lcd.setCursor(1, 0);
}

```

Adapun hasil yang ditampilkan oleh LCD, ditunjukkan pada Gambar 4.24 berikut:



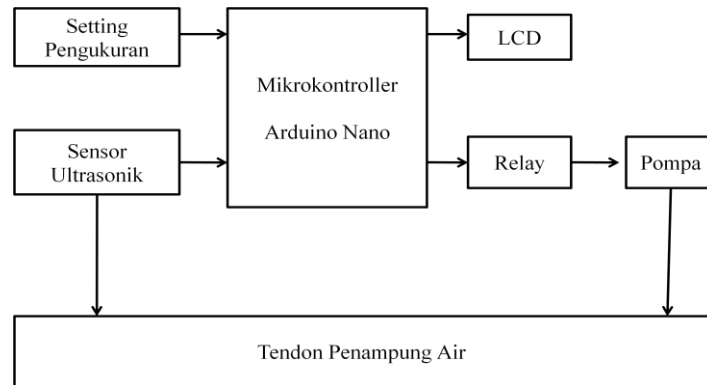
Gambar 4.24 Pengujian Tampilan *LCD* 16x2

Dari hasil pengujian yang dilakukan dapat disimpulkan bahwa *LCD* dapat bekerja sesuai dengan yang diinginkan.

4.7 Perancangan dan Pembuatan *Software*

Perangkat lunak berfungsi sebagai sarana interaksi yang menghubungkan atau menjembatani pengguna komputer dengan perangkat keras. Perangkat lunak akan memproses data atau perintah hingga mendapatkan hasil atau menjalankan sebuah perintah-perintah. Pada *monitoring* dan pengontrol penampung air berbasis *Internet of Things* ini menggunakan ESP8266 sebagai sarana pertukaran data antara Android dan *interface* aplikasi *monitoring* dan pengontrol penampungan air. Mikrokontroler yang digunakan adalah Arduino Nano.

Diagram blok keseluruhan sistem ditunjukkan pada Gambar 4.25 dibawah ini:

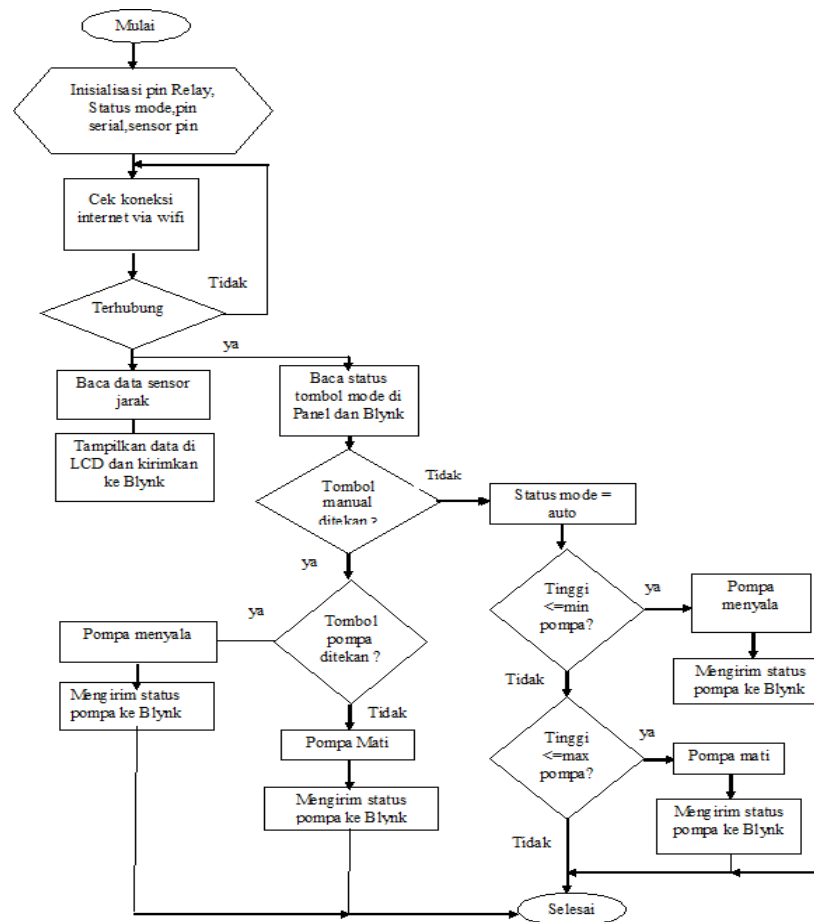


Gambar 4.25 Diagram Blok

4.7.1 Flowchart Program Arduino

Pemrograman dilakukan pada *software* Arduino IDE untuk pengolahan data yang akan dilakukan oleh mikrokontroler arduino nano. Pada *flowchart* Arduino yang pertama kali dilakukan adalah memulai program yang telah dibangun, kemudian melakukan penyediaan tempat-tempat pengolahan data dalam *storage* dalam hal ini yaitu inisialisasi pin relay, status mode pin serial serta pin sensor jarak. Kemudian akan dilakukan pengecekan koneksi dengan wifi yang digunakan ESP8266 dengan yang ada pada pemrograman. Setelah terhubung akan dilakukan proses pembacaan data sensor jarak yang akan diproses dan status ditampilkan pada *LCD* dan dikirimkan ke *Blynk*. Apabila koneksi telah terhubung ke ESP8266 pembacaan status tombol mode di *Blynk* juga dilakukan. Terdapat dua pilihan mode dalam program yang telah dibangun ini yaitu mode manual dan mode auto. Jika tombol manual ditekan pompa akan ikut ditekan dan pompa akan menyala serta mengirimkan status pompa ke *Blynk*. Namun, jika pompa tidak ditekan maka pompa akan mati dan informasi status pompa akan dikirimkan ke *Blynk*. Apabila tidak menekan tombol manual maka akan dilakukan proses status mode yaitu Auto. Pada mode auto pengaturan batas tinggi minimum dan maksimum ketinggian air bisa diatur. Jika pengguna melakukan *settings* pada tinggi kurang dari atau sama dengan batas minimum pompa maka pompa akan menyala dan mengirimkan status informasi pompa ke *Blynk*, jika tidak maka *decision* akan

menampilkan pilihan tinggi kurang dari atau sama dengan batas maksimum pompa, pompa akan mati dan status pompa akan dikirimkan ke *Blynk*. Setelah selesai dilakukan proses, *decision*, dan *preddifined process* maka program akan berakhir. *Flowchart* arduino ditunjukkan pada Gambar 4.26.



Gambar 4.26 *Flowchart* Arduino

4.7.2 Pembuatan Program Arduino IDE

Implementasi pemrograman Arduino IDE dibagi menjadi tiga yaitu ini insialisasi, *void setup* dan *void loop*. Pada program arduino yang harus dilakukan yaitu menambahkan *library* ESP8266 pada *board manager* agar dapat memprogram ESP8266 dengan *basic* program arduino. Selain itu penambahan *library blynk* pada dasarnya adalah kumpulan fungsi-fungsi yang bisa melakukan suatu pekerjaan tertentu.

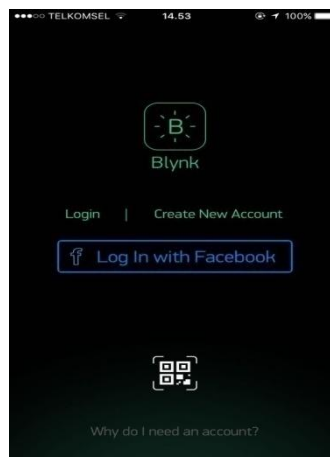
Bagian *void setup* digunakan untuk menginisialisasi variabel-variabel yang akan digunakan dan pada bagian *void setup* dan dijalankan satu kali yaitu pada saat modul ESP8266 aktif atau menyala. Fungsi pin *Mode* pada *void setup()* digunakan untuk menginisiasi sebuah pin tersebut yang akan digunakan sebagai *input* atau *output*.

4.7.3 Perancangan *Blynk*

Blynk adalah *platform* untuk aplikasi OS *Mobile* (*Ios* dan *Android*) yang bertujuan untuk kendali *module* *Arduino* dan *ESP8266*. Aplikasi ini digunakan untuk membuat antarmuka grafis untuk proyek yang akan diimplementasikan hanya dengan metode *drag and drop widget*.

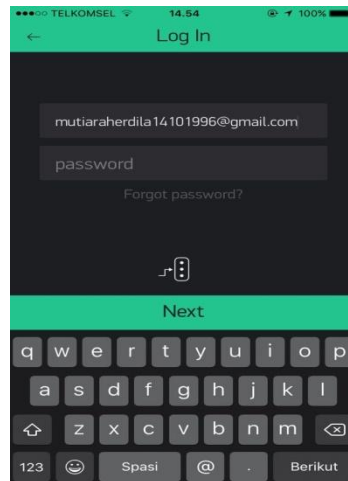
Untuk merancang *Blynk* yang pertama kali dilakukan adalah mendownload aplikasi *Blynk* pada *Android*. Kemudian buka aplikasi dan *login* menggunakan *e-mail* yang sudah kita miliki. Buat *new project* dan pilih *module* *ESP8266* yang akan digunakan sebagai sarana terhubung ke internet. Setelah itu *drag and drop* rancangan proyek yang kita inginkan. Kemudian *Blynk* akan mengirimkan *Token Auth* melalui *e-mail* yang akan digunakan untuk program yang di download ke *module*.

Software yang digunakan untuk memprogram serta mendownload program ke module adalah *Arduino IDE*. Gambar 4.27 menunjukkan tampilan ketika masuk ke aplikasi *Blynk*.



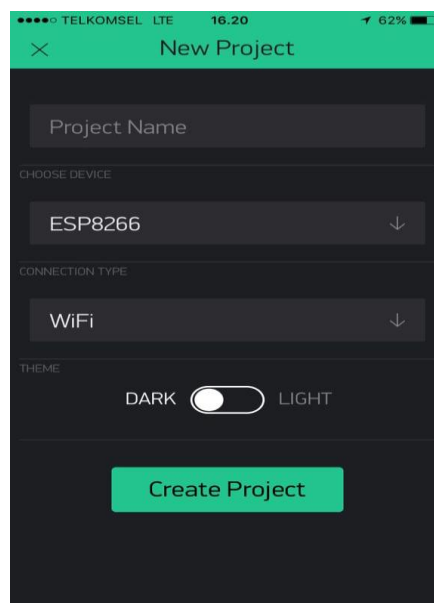
Gambar 4.27 Tampilan awal pada aplikasi *Blynk*

Berikut merupakan gambar yang menunjukkan ketika akan *log in* menggunakan *e-mail* seperti yang ditunjukkan pada gambar 4.28 dibawah ini.



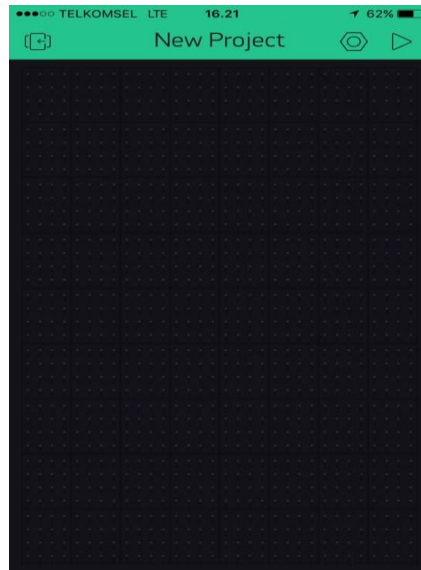
Gambar 4.28 Tampilan *Log In*

Berikut merupakan Gambar 4.29 yang menunjukkan pembuatan serta penamaan project yang kita inginkan



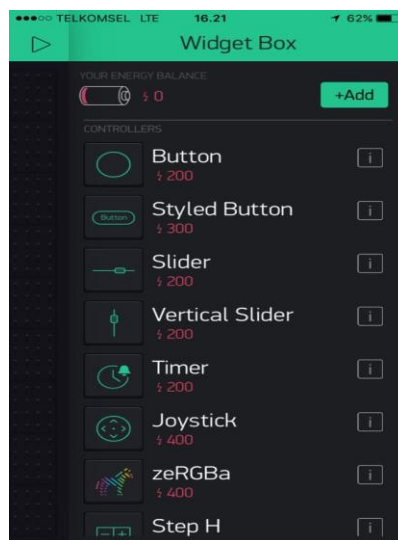
Gambar 4.29 Penginisialan nama *Project*

Berikut adalah halaman *new project* yang ditunjukkan pada Gambar 4.30 merupakan *interface* untuk kita membuat desain aplikasi yang diinginkan.



Gambar 4.30 Tampilan pada menu *new project*

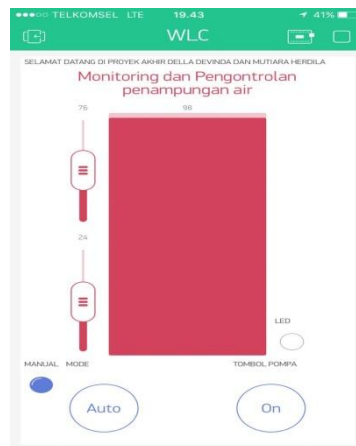
Berikut adalah halaman *new project* yang merupakan *interface* untuk kita membuat desain aplikasi yang diinginkan dengan menggeser layar ke kanan dan akan muncul *Widget Box* yang berisi aksesoris yang kita inginkan pada Gambar 4.31.



Gambar 4.31 Aksesoris pada *Widget Box*

4.8 Pembuatan *Interface Monitoring dan Pengontrolan Penampung Air*

Interface Monitoring dan Pengontrolan Penampung Air berfungsi sebagai tampilan status *mode* dalam kondisi manual atau otomatis, status ketinggian air dan pompa dalam keadaan mati atau menyala. Pada tampilan ini pengguna bisa mengetahui status ketinggian air dengan pengontrolan menggunakan mode manual atau auto (otomatis) seperti pada Gambar 4.32 dibawah ini.



Gambar 4.32 *Interface Monitoring dan Pengontrolan penampung air*

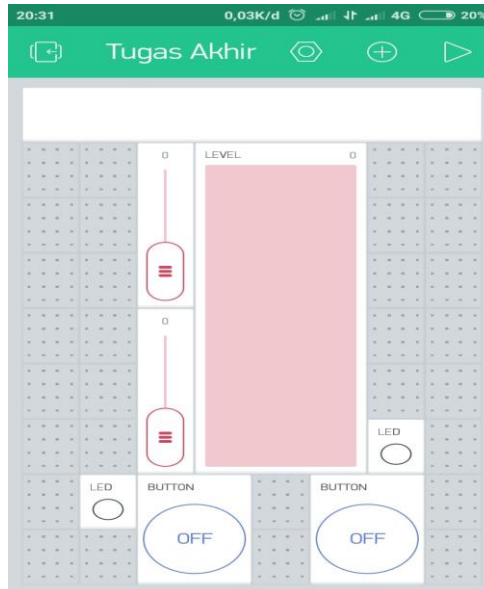
4.8.1 *Design Interface Aplikasi Monitoring dan Pengontrolan Penampung Air*

Interface aplikasi *monitoring dan pengontrolan penampung air* ini dibangun untuk mengoptimalkan fungsi alat dengan menambahkan aplikasi Android yang dapat mengontrol penampung air dan hasil *monitoring* penampungan air ditampilkan pada *interface* aplikasi *monitoring dan pengontrolan penampung air*. Pada perancangan *interface* aplikasi *monitoring dan pengontrolan penampung air* ini digunakan 1 labelled value, 2 *button*, 1 *slider settings*, 1 *level settings* dan 2 *led settings*. Berikut penjelasan mengenai labelled value, button, slider settings, level settings dan led settings:

- *Button1* : *Auto / Manual*
- *Button2* : *On / Off*
- *Labelled value* : *Nama Project*
- *Slider settings* : *Min / Max*
- *Led settings1* : *Indikator manual atau auto*
- *Led settings2* : *Indikator On atau Off*

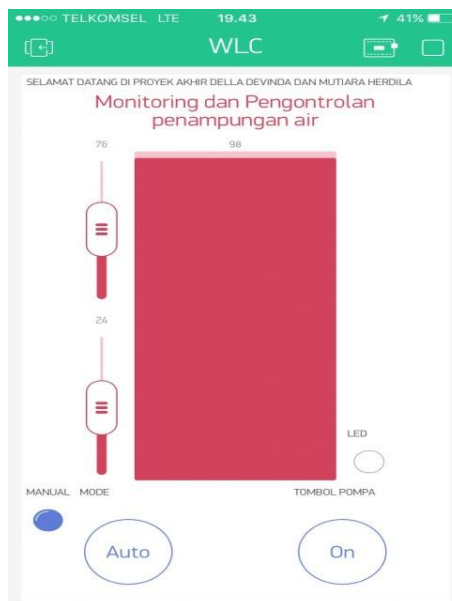
- *Level settings* : *Monitoring* ketinggian air

Desain perancangan aplikasi Monitoring dan Pengontrol Penampung Air ditunjukkan pada Gambar 4.33.



Gambar 4.33 Desain Aplikasi

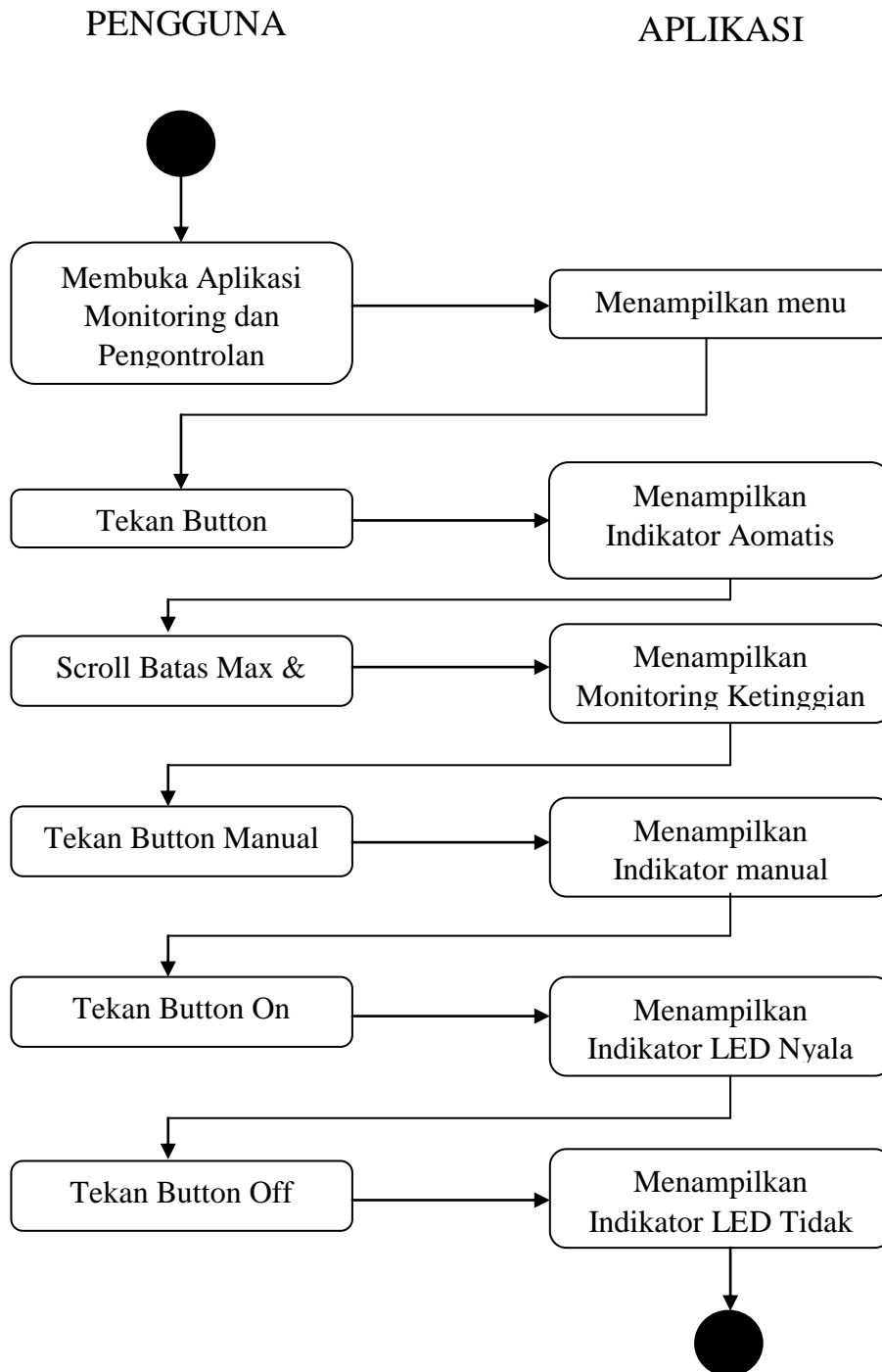
Berikut adalah hasil akhir interface aplikasi *monitoring* dan pengontrolan penampung air seperti yang ditunjukkan pada Gambar 4.34 dibawah ini.



Gambar 4.34 Hasil akhir *Interface* Aplikasi

4.8.2 Diagram Activity Program *Blynk*

Berikut adalah bentuk tampilan dari diagram *activity* dari aplikasi *blynk* seperti yang ditunjukkan pada Gambar 4.24 dibawah ini.



Gambar 4.35 Diagram *Activity* aplikasi *Blynk*

Pada diagram Activity aplikasi akan berjalan apabila aplikasi pada *monitoring* dan pengontrolan penampung air dibuka dan sudah terkoneksi dengan internet. Ketika membuka aplikasi sistem akan menampilkan tampilan menu utama yang dapat kita kontrol. Selain itu pada aplikasi akan menampilkan *button* Auto atau *button* Manual. Ketika kita tekan *button* Auto maka pada *interface* aplikasi akan menampilkan indikator Auto, kemudian pengguna dapat melakukan *scroll* terhadap batas maksimal dan minimal air. Setelah itu akan menampilkan *monitoring* ketinggian air. Apabila menekan *button* manual maka akan menampilkan indikator manual, pengisian air akan dilakukan secara manual yaitu dengan menekan *button on* pompa ketika pompa dalam keadaan *on* menu utama akan menampilkan indikator berupa *led* menyala. Ketika *button off* pompa ditekan maka akan menampilkan indikator berupa *led* tidak menyala.

4.9 Hasil Pengujian

Dalam pengambilan data *monitoring* dan pengontrolan penampung air ini akan dilakukan pada hari yang berbeda dalam pengendalian menggunakan aplikasi yang sudah dirancang. Hasil tersebut akan dibandingkan dengan status ketinggian air dan pompa apakah sama atau tidak dan untuk pengujian sensor Ultrasonik apakah sensor tersebut dapat bekerja sesuai dengan respon yang diterima. Pengujian secara keseluruhan merupakan pengujian dimana semua komponen dalam penelitian ini telah terintegrasi satu sama lain.

4.9.1 Analisa dan Hasil Pengujian Keseluruhan Sistem

Setelah pembuatan *software* maupun *hardware* selesai, selanjutnya masuk ke tahap akhir yaitu pengujian keseluruhan sistem yang dibuat. Pengujian ini bertujuan untuk melakukan pengisian air tandon dengan perintah dari android. Untuk mengukur ketinggian air tandon yang sudah terisi dilakukan secara manual. Hasil pengukuran ini dibandingkan dengan pengukuran otomatis, jika hasilnya sama maka sistem tersebut dikatakan berhasil, jika tidak dilakukan perbaikan. Untuk pengujian nya dilakukan tahapan sebagai berikut:

1. Pengguna membuka aplikasi *Blynk* pada *android*.
2. Pilih mode yang akan digunakan.
3. Pengguna menekan tombol otomatis maka aplikasi akan menampilkan indikator otomatis.
4. Pengguna memilih ketinggian maksimum dan minimum dengan menggunakan scroll bar pada *android*.
5. Pengguna menekan tombol manual maka aplikasi menampilkan indikator manual.
6. Pengguna menekan tombol on pompa maka aplikasi menampilkan berupa led menyala.
7. Pengguna menekan tombol off pompa maka aplikasi menampilkan berupa led tidak menyala.

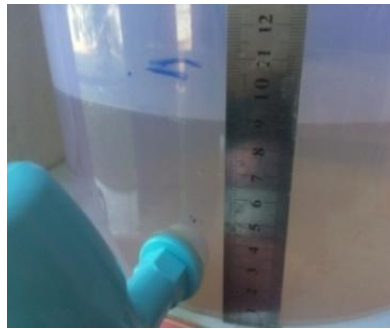
Adapun hasil pengujian seperti yang ditunjukkan pada tabel 4.4 dibawah ini.

Tabel 4.4 pengujian keseluruhan

No	Mode yang dipilih	Pengukuran ketinggian		Error (cm)
		Air		
		Aplikasi Blynk	Manual	
1	Auto	9,3cm	9,0cm	0,3
2	Manual	14,0cm	14,5cm	0,5
3	Auto	20,4cm	20,2cm	0,2
4	Manual	24,1cm	24,0cm	0,1
5	Auto	28,0cm	28,4cm	0,4
Rata-rata error (cm)				0,3

1. Analisa pengujian perintah “Auto”

Pengambilan data pertama dengan memberikan instruksi berupa mode Auto. Dimana pengguna akan menscroll batas *max* secara otomatis. Pada aplikasi diberikan level ketinggian air sebesar 9,3cm. Namun pada *hardware* menunjukkan level ketinggian air sebesar 9,0cm. Pada pengambilan data pertama ini terdapat error sebesar 0,3cm seperti yang ditunjukkan pada gambar dibawah ini.



Gambar 4.36 Pengujian Perintah “Auto”

Berikut merupakan gambar tampilan LCD saat kondisi Auto yang ditunjukkan pada gambar dibawah ini.



Gambar 4.37 Pengujian Perintah “Auto”

2. Analisa pengujian perintah “Manual”

Pengambilan data kedua adalah pengguna akan melakukan pengisian air melalui aplikasi secara manual dengan menekan tombol on pompa ketika akan memulai pengisian air dan menekan tombol off pompa ketika pengisian air selesai dilakukan. *Monitoring* level ketinggian air menunjukkan sebesar 14,0cm. Kemudian pada hardware level ketinggian air

menunjukkan 14,5cm. Pada pengambilan data kedua ini terjadi error sebesar 0,5cm seperti ditunjukkan pada gambar dibawah ini.



Gambar 4.38 Pengujian Perintah “Manual”

Berikut merupakan gambar tampilan LCD saat kondisi Manual yang ditunjukkan pada gambar dibawah ini.



Gambar 4.39 Pengujian Perintah “Manual”

3. Analisa pengujian perintah “Auto”

Pengambilan data ketiga adalah dengan memberikan instruksi berupa mode Auto. Dimana pengguna akan menscroll batas *max* secara otomatis. Pada aplikasi diberikan level ketinggian air sebesar 20,4cm. Namun pada *hardware* menunjukkan level ketinggian air sebesar 20,2cm. Pada pengambilan data ketiga ini terdapat error sebesar 0,2cm seperti ditunjukkan pada gambar dibawah ini.



Gambar 4.40 Pengujian perintah “Auto”

Berikut merupakan gambar tampilan LCD saat kondisi Auto yang ditunjukkan pada gambar dibawah ini.



Gambar 4.41 Pengujian perintah “Auto”

4. Analisa pengujian perintah “Manual” :

Pengambilan data keempat ini pengguna akan melakukan pengisian air melalui aplikasi secara manual dengan menekan tombol on pompa ketika akan memulai pengisian air dan menekan tombol off pompa ketika pengisian air selesai dilakukan. Monitoring level ketinggian air menunjukkan pada aplikasi sebesar 24,1cm dan status pada hardware juga menunjukkan level ketinggian air yang sama yaitu 24,0cm. Pada pengambilan data ini terdapat error sebesar 0,1cm seperti ditunjukkan pada gambar dibawah ini.



Gambar 4.42 Pengujian Perintah “Manual”

Berikut merupakan gambar tampilan LCD saat kondisi Auto yang ditunjukkan pada gambar dibawah ini.



Gambar 4.43 Pengujian Perintah “Manual”

5. Analisa pengujian perintah “Auto”

Pengambilan data kelima yang merupakan pengambilan data terakhir diberikan instruksi berupa mode auto. Tampilan *monitoring* level ketinggian air menunjukkan pada aplikasi ketinggian air sebesar 28,0cm dan status pada *hardware* juga menunjukkan level ketinggian air yang sama yaitu 28,4cm. Pada pengambilan data kelima ini terdapat error sebesar 0,4cm. seperti ditunjukkan pada gambar dibawah ini.



Gambar 4.44 Pengujian Perintah “Auto”

Berikut merupakan gambar tampilan LCD saat kondisi Auto yang ditunjukkan pada gambar dibawah ini.



Gambar 4.45 Pengujian Perintah “Auto”

Berdasarkan hasil pengambilan data yang sudah dilakukan diatas dapat dipastikan bahwa percobaan pengambilan data berupa status ketinggian air dan status kondisi pompa pada keadaan sebenarnya dengan aplikasi berhasil. Percobaan dilakukan sebanyak 5 kali dengan mode yang berbeda-beda.

Koneksi dengan internet sangat diperlukan dalam pembuatan *monitoring* dan pengontrolan penampung air ini. Apabila koneksi yang digunakan pada ESP8266 mempunyai kecepatan data yang sangat cepat maka *Arduino* akan cepat membaca data yang dikirimkan oleh Android. Dalam hal ini ketika salah satu tombol ditekan pada aplikasi dan dikirimkan ke ESP8266 maka akan *delay* normalnya hanya 5 detik selama ESP8266 *mendownload* data tersebut. Namun apabila koneksi sangat lamban maka akan diperlukan waktu 10-15 menit dalam melakukan proses *download* data.

$$\text{Kecepatan} = \frac{\text{jumlah file data}}{\text{Waktu}} \dots\dots\dots[2]$$

Waktu

Jumlah data = 1kilo *byte*

Waktu = 5 *second*

Kecepatan = 1 kilo *byte* / 5 *second* = 0,2 kilo *byte per second*

Semakin kecil jumlah file data yang digunakan maka akan semakin cepat pula waktu dan kecepatan dalam melakukan proses *download* data.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan mulai dari tahap perancangan hingga uji coba sistem maka dapat ditarik kesimpulan sebagai berikut:

1. Alat Monitoring dan Pengontrolan Penampung Air ini dapat memonitoring dan mengontrol ketinggian air menggunakan *smartphone* melalui konektivitas internet yang tersedia tanpa batasan jarak dan waktu.
2. Pembacaan sensor ultrasonik pada alat ini memiliki rata-rata nilai *error* sebesar 0,28cm terhadap alat ukur yang digunakan dan dapat berubah-ubah sesuai dengan besarnya riak air didalam wadah penampung.
3. Pengiriman data-data sensor pada alat ini maupun instruksi memiliki kecepatan sekitar 0,2 kilo *byte per second*. Kecepatan dapat berubah-ubah sesuai dengan kecepatan konektivitas internet yang digunakan dan jumlah data yang dikirimkan.

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan diperoleh beberapa hal yang dapat dijadikan saran untuk dilakukan pengembangan lebih lanjut yaitu:

1. Membuat aplikasi monitoring yang *compatible* dengan *smartphone* menggunakan *App Inventor* ataupun *Virtuino* sehingga tidak bergantung pada aplikasi *Blynk* yang berbayar.
2. Menambahkan kamera pada sistem sehingga dapat menampilkan visualisasi ketinggian air secara langsung.

DAFTAR PUSTAKA

- [1] Arifin Ilfan, (2015), “*Automatic Water Level Control* Berbasis Mikrokontroller Dengan Sensor Ultrasonik”, skripsi, Universitas Negeri Semarang.
- [2] Asthon, K. (2009), “*That 'internet of things' thing*. RFiDJ.22(7):97-114. Barbaran.
- [3] Anita Sari Lubis, Nia, (2017), “*Sistem Monitoring Level Ketinggian Air Bendungan Menggunakan Sensor Ultrasonik Yang Dikontrol Melalui Android*”, tugas akhir, Universitas Sumatera Utara.
- [4] Bagja Priandana, Asep, (2017), *Esp8266-Primadona Baru IoT*, <https://blog.framework.id>, diakses pada tanggal 12 Juli 2018.
- [5] Elmech, (2016). Diambil dari : <https://elmechtechnology.com/product/relay-module-5V-1-channel>, diakses pada tanggal 8 Agustus 2018.
- [6] family-cybercode.blogspot.com/2016/01/mengenal-arduino-nano.html, diakses pada tanggal 8 Agustus 2018.
- [7] Fatma, (2018). Diambil dari : elektronikadasar.info/sensor-ultrasonik.html (8 Agustus 2018).
- [8] Muhammad Hanafiyah, Ali, (2013), <https://electrozone94.blogspot.com/2013/09/catu-daya-adaptor.html>, diakses pada tanggal 8 Agustus 2018.
- [9] Purnomosejati, (2011). Diambil dari : <https://purnomosejati.wordpress.com/2011/08/25/mengenal-komunikasi-I2Cinter-integrated-circuit>. (8-8-2018)
- [10] Tri Santoso, Herman, (2014), “*Water Level Control Menggunakan Elektroda Untuk Diterapkan Pada Rumah Tangga dan Home Industri*”, skripsi, Unversitas Muhammadiyah Ponorogo.

LAMPIRAN

LAMPIRAN 1
DAFTAR RIWAYAT HIDUP

LAMPIRAN 2

PROGRAM

LAMPIRAN 3

DATA SHEET

DAFTAR RIWAYAT HIDUP



1. Data Pribadi

Nama Lengkap : Della Devinda
Tempat & Tanggal Lahir : Penyamun, 13 April 1998
Alamat Rumah : Jl.Pangkal layang Dusun Muntabak Desa Penyamun
Jenis Kelamin : Perempuan
Agama : Islam
No.Telepon : 082188980424
E-mail : delladevinda@gmail.com

2. Riwayat Pendidikan

SDN 8 Muntabak : 2003-2009
SMPN 5 Sungailiat : 2009-2012
SMAN 1 Pemali : 2012-2015

3. Pengalaman Kerja

Praktek Kerja Lapangan di PT.Yamaha Music Manufacturing Asia	7 September 2017- 7 Januari 2018
---	----------------------------------

4. Pengetahuan Bahasa : Bahasa Indonesia

5. Hobi : Bersepeda, *Traveling*

Sungailiat, 02 Agustus 2018

Della Devinda

DAFTAR RIWAYAT HIDUP



1. Data Pribadi

Nama Lengkap : Mutiara Herdila
Tempat & Tanggal Lahir : Sungailiat, 13 Oktober 1996
Alamat Rumah : Jl. A.Yani Jalur II Komp.PU (Depan kantor KPU)
Jenis Kelamin : Perempuan
Agama : Islam
No.Telepon : 081297311714
E-mail : mutiaraherdila14101996@gmail.com

2. Riwayat Pendidikan

SDN 15 Sungailiat : 2002-2008
SMPN 2 Sungailiat : 2008-2011
SMAN 1 Sungailiat : 2011-2014

3. Pengalaman Kerja

Praktek Kerja Lapangan di PT.Yamaha Music Manufacturing Asia	7 September 2017- 7 Januari 2018
---	----------------------------------

4. Pengetahuan Bahasa : Bahasa Indonesia

5. Hobi : *Traveling and shopping*

Sungailiat, 02 Agustus 2018

Mutiara Herdila

PROGRAM

```
#define BLYNK_PRINT Serial
#include <ESP8266_Lib.h>
#include <BlynkSimpleShieldEsp8266.h>

#include <NewPing.h>

#define TRIGGER_PIN 12 // Arduino pin tied to trigger pin on the ultrasonic
sensor.
#define ECHO_PIN 11 // Arduino pin tied to echo pin on the ultrasonic sensor.
#define MAX_DISTANCE 57 // maksimum ketinggian yg diukur dalam satuan
cm.
NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);

// variable tambahan
#define switchmode 8 //pin tombol 1/ tombol mode
#define switchpompa 9 //pin tombol 2/ tombol on off pompa
int bacaswitchmode;//variabel menyimpan status tombol 1
int bacaswitchpompa;//variabel menyimpan status tombol 2
long previous1=HIGH;//variabel pelengkap untuk tombol1
long previous2=HIGH;//variabel pelengkap untuk tombol2
long time=0;//variable untuk delay tombol 1 dan 2
long delaylcd=0;//variable untuk delay lcd

#include <Wire.h>

#include <LiquidCrystal_I2C.h>
```

```
LiquidCrystal_I2C lcd(0x27, 16, 2);
```

```
char auth[] = "d7e3f2062d334e25bc1ec5fb1ccb49cb";
```

```
char ssid[] = "Della devinda";
```

```
char pass[] = "11223344";
```

```
#include <SoftwareSerial.h>
```

```
SoftwareSerial EspSerial(2, 3); // RX, TX
```

```
#define ESP8266_BAUD 9600
```

```
ESP8266 wifi(&EspSerial);
```

```
BlynkTimer timer;
```

```
int tombolMode = 0;
```

```
int tombolPompa = 1;
```

```
int statusPompa = 0;
```

```
int pompa = 5;

int tinggi = 0;
int tinggi2 = 0;
int tinggi3 = 0;
float cm;
int maxPompa = 80;
float liter;

void kirimSensor()
{

    tinggi = map(sonar.ping_cm(), 28, 56.5, 100, 0);
    if (tinggi>=0 && tinggi <= 100)
    {

        Blynk.virtualWrite(V1, tinggi2);
    }
    Blynk.virtualWrite(V3, statusPompa);
    Blynk.virtualWrite(V6, tombolMode*255);
}

void lcdTampil()
{
    String mode;
    tinggi = map(sonar.ping_cm(), 28, 56.5, 100, 0);
    if (tinggi>=0 && tinggi <= 100)
    {
        tinggi2=tinggi;
    }
}
```

```
liter = (706.5*(tinggi2*0.000283));
cm = tinggi2*0.28;
if(cm>=28)cm=28;
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Lev:"+String(tinggi2));
lcd.print("%");
lcd.setCursor(9,0);
lcd.print("V:"+String(liter,1));
lcd.print("L");
lcd.setCursor(0,1);
if(tombolMode==1)mode="M";
else{mode="A";}
lcd.print("M:"+String(mode));
lcd.setCursor(4,1);
lcd.print("T:"+String(cm,1));
lcd.print("CM");
}
```

```
void setup()
{
  pinMode(switchmode, INPUT_PULLUP);
  pinMode(switchpompa, INPUT_PULLUP);

  pinMode(pompa, OUTPUT);
  digitalWrite(pompa, HIGH);
  delay(10);

  lcd.init();
  lcd.backlight();
}
```

```
Serial.begin(9600);
```

```
EspSerial.begin(ESP8266_BAUD);
```

```
Blynk.begin(auth, wifi, ssid, pass, "blynk-cloud.com", 8442);
```

```
timer.setInterval(200L, kirimSensor);
```

```
timer.setInterval(200L, lcdTampil);
```

```
}
```

```
BLYNK_WRITE(V0)
```

```
{
```

```
  tombolMode = param.asInt(); // assigning incoming value from pin V0 to a  
  variable
```

```
}
```

```
BLYNK_WRITE(V2)
```

```
{
```

```
  tombolPompa = param.asInt(); // assigning incoming value from pin V2 to a  
  variable
```

```
}
```

```
//BLYNK_WRITE(V4)
```

```
//{
```

```

//
// minPompa = param.asInt(); // assigning incoming value from pin V4 to a
variable
//
//}
BLYNK_WRITE(V5)
{

    maxPompa = param.asInt(); // assigning incoming value from pin V5 to a
variable

}

void loop()
{
    Blynk.run();
    timer.run();

    bacaswitchmode=digitalRead(switchmode);//baca status tombol 1/tombol mode
    bacaswitchpompa=digitalRead(switchpompa);//baca status tombol 2/tombol on
off pompa

    if (millis()-delaylcd>1000)//tiap 500ms menampilkan data ke lcd
    {
        lcdTampil();
        delaylcd=millis();//(reset nilai waktu 500ms diatas)
    }

    if (bacaswitchmode==LOW && previous1== HIGH && millis()-
time>200)//kondisi ketika tombol 1 ditekan selama 200ms

```

```

{
  if(tombolMode==1){tombolMode=0;}//jika tombol mode=1 maka tombol mode
dirubah ke 0
  else{tombolMode=1;}//sebaliknya
  time=millis();//nilai time = millis (reset nilai waktu 200ms diatas)
}

if (tombolMode == 1)//mode manual diaktifkan
{

  if (bacaSwitchpompa==LOW && previous2== HIGH&& millis()-
time>200)//kondisi ketika tombol 2 ditekan selama 200ms
  {
    if (tombolPompa==0){tombolPompa=1;} //jika tombol tombolpompa=1 maka
tombolpompa dirubah ke 0
    else {tombolPompa=0;} //sebaliknya
    time=millis();//nilai time = millis (reset nilai waktu 200ms diatas)
  }

  if (tombolPompa==0){statusPompa=255;}
  if (tombolPompa==1){statusPompa=0;}

  digitalWrite(pompa, tombolPompa);//mengaktifkan pompa sesuai dengan nilai
tombolpompa

}

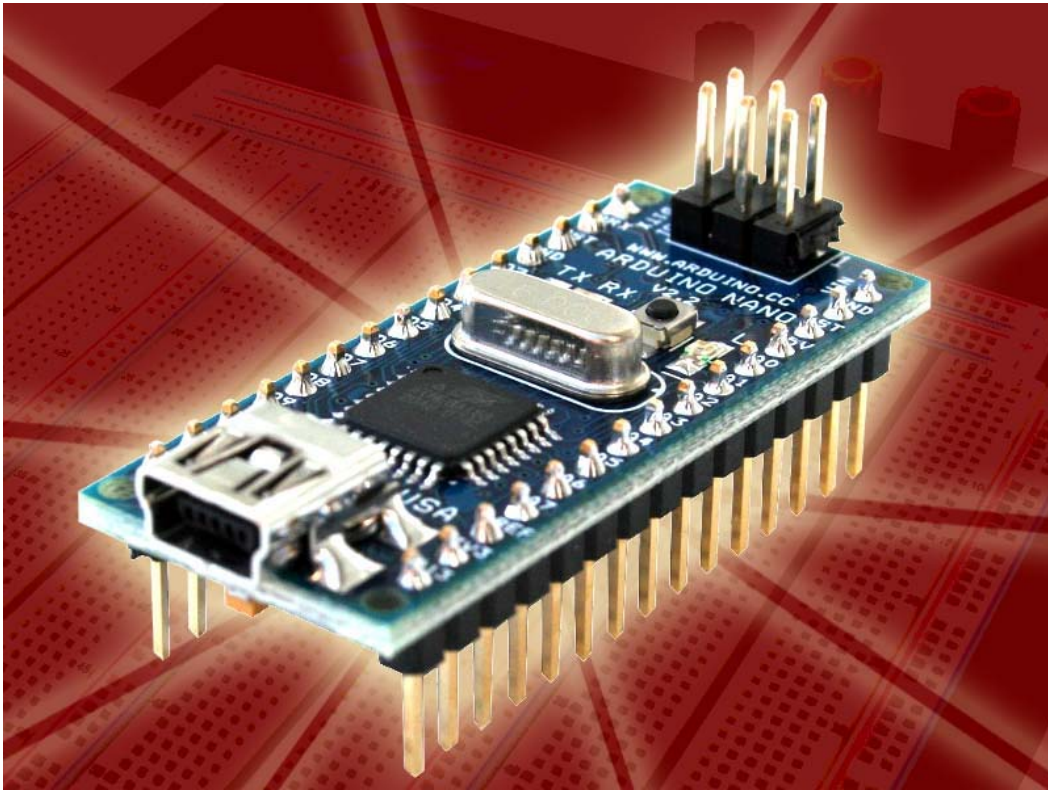
if (tinggi2 <= 100 && tombolMode == 0)
{
  if (tombolMode == 0 && tinggi2 >= maxPompa)

```

```
{  
    digitalWrite(pompa, HIGH);  
    statusPompa=0;  
}  
else {digitalWrite(pompa, LOW);}  
}  
//previous adalah kondisi terakhir switchmode dan switchpompa  
previous1=bacaswitchmode;  
previous2=bacaswitchpompa;  
}
```


Arduino Nano (V2.3)

User Manual



Released under the Creative Commons Attribution Share-Alike 2.5 License

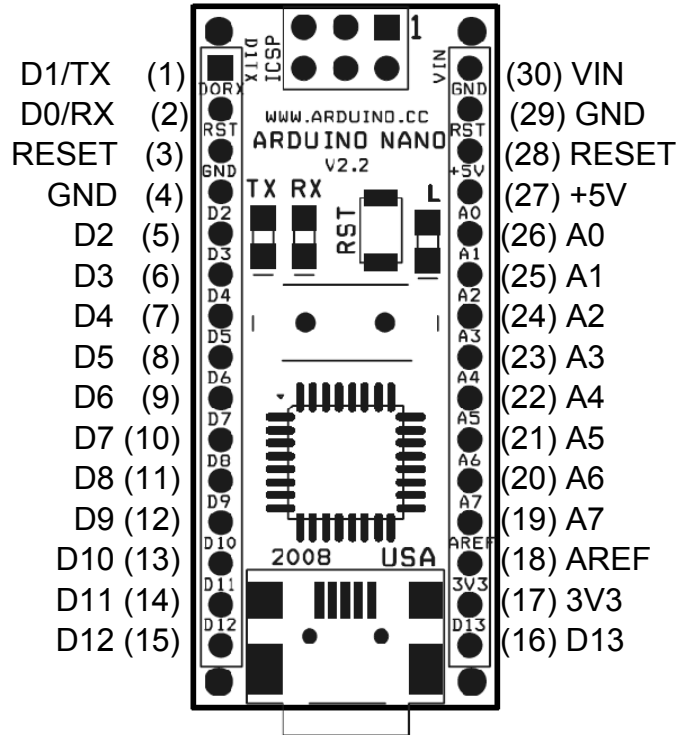
<http://creativecommons.org/licenses/by-sa/2.5/>

More information:

www.arduino.cc

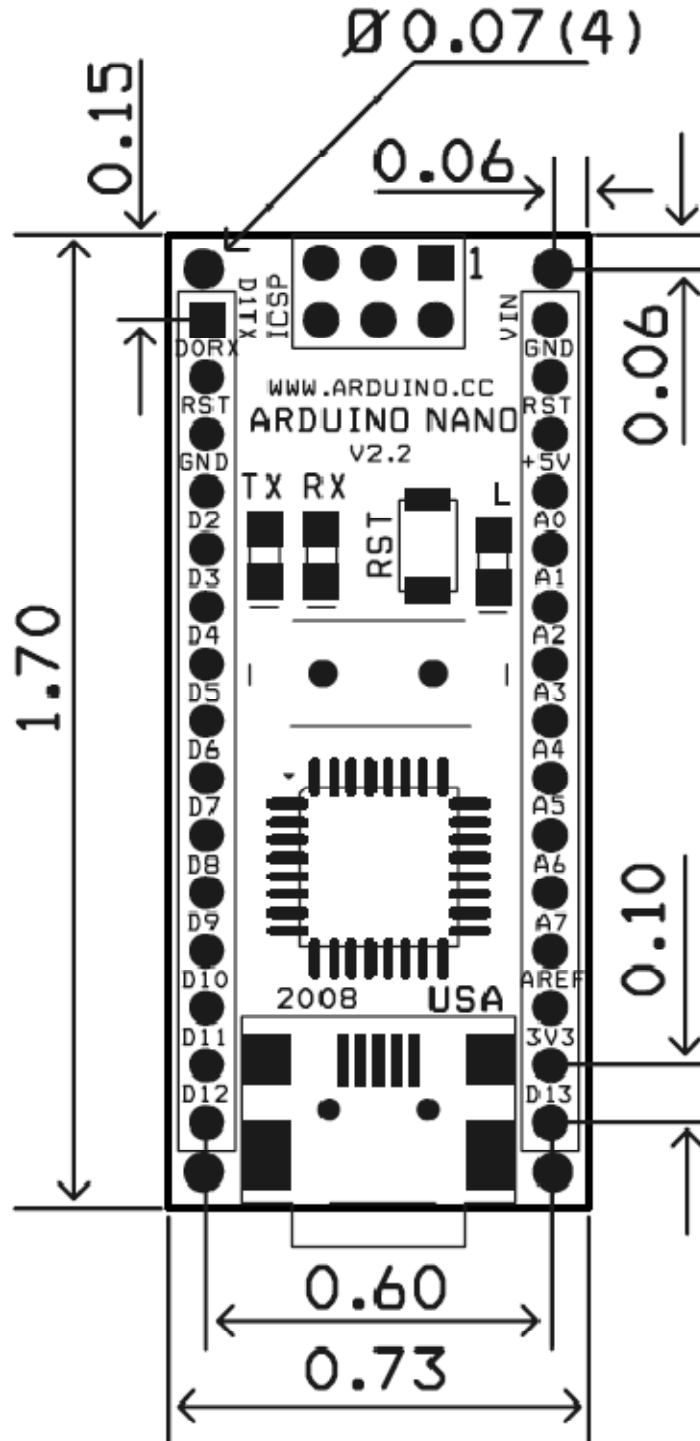
Rev. 2.3

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



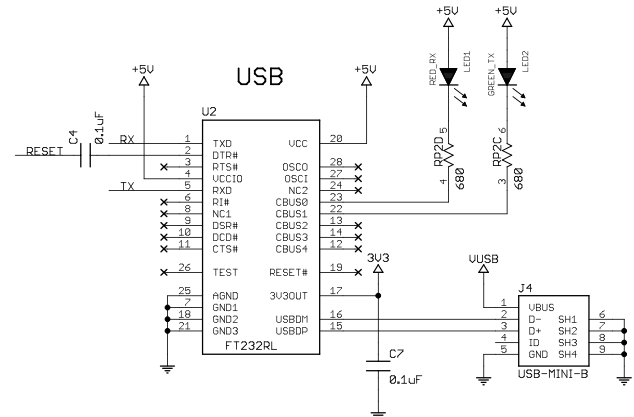
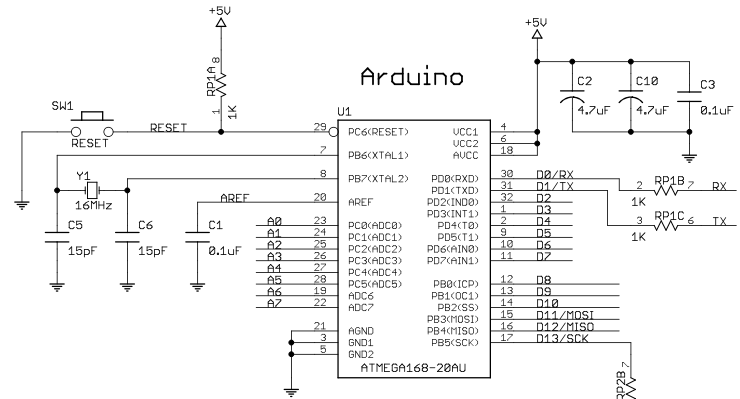
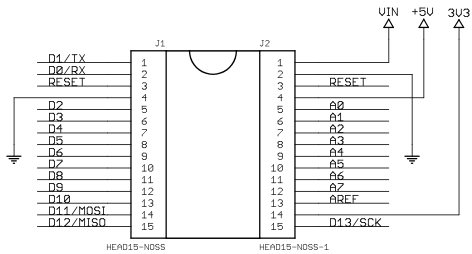
ALL DIMENSIONS ARE IN INCHES

Arduino Nano Bill of Material

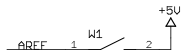
Item Number	Qty.	Ref. Dest.	Description	Mfg. P/N	MFG	Vendor P/N	Vendor
1	5	C1,C3,C4,C7,C9	Capacitor, 0.1uF 50V 10% Ceramic X7R 0805	C0805C104K5RACTU	Kemet	80-C0805C104K5R	Mouser
2	3	C2,C8,C10	Capacitor, 4.7uF 10V 10% Tantalum Case A	T491A475K010AT	Kemet	80-T491A475K010	Mouser
3	2	C5,C6	Capacitor, 18pF 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	APT2012SRCPRV	Kingbright	604-APT2012SRCPRV	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	Cystal, 16MHz +/-20ppm HC-49/US Low Profile	ABL-16.000MHZ-B2	Abracon	815-ABL-16-B2	Mouser

Arduino Nano Schematic

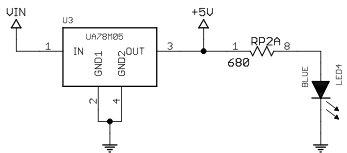
Copyright 2008 under the Creative Commons Attribution Share-Alike 2.5 License
<http://creativecommons.org/licenses/by-sa/2.5/>



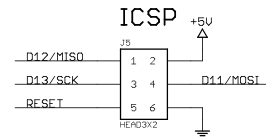
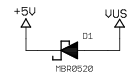
+5V AREF OPTION



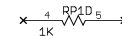
+5V REG



+5V AUTO SELECTOR



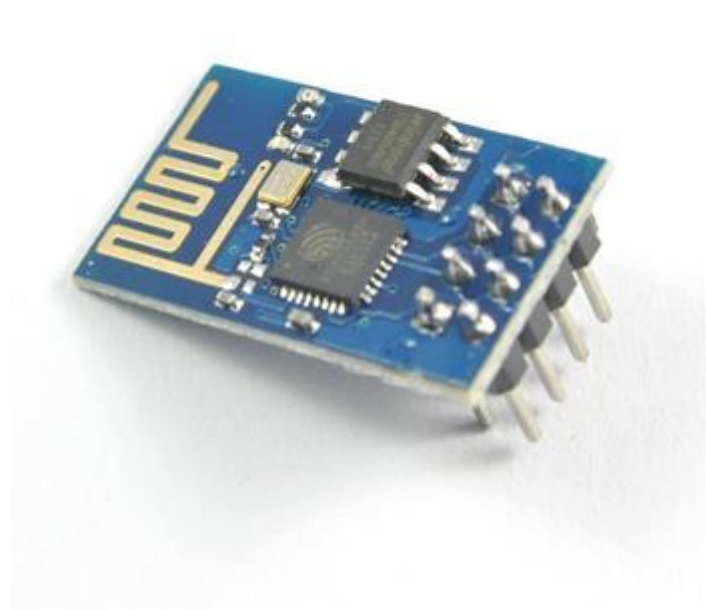
NOT USED



v2.3 - Modify FT232RL to use +5V

TITLE: Arduino Nano	
Document Number:	REV: 2.3
Date: 6/26/2008 8:35:54 PM	Sheet: 1/1

ESP8266 Serial Esp-01 WIFI Wireless



Building the gcc toolchain

have a look at the github wiki <https://github.com/esp8266/esp8266-wiki/wiki>

Code examples

have a look at the github wiki <https://github.com/esp8266/esp8266-wiki/wiki>

Running the module

The modules pins only allow 3.3v (use a multi meter to check your serial lines if you are not sure)

Connect CH_PD to VCC to make it boot

Uploading code

The modules pins only allow 3.3v (use a multi meter to check your serial lines if you are not sure)

see <https://github.com/esp8266/esp8266-wiki/wiki/Uploading>

links

Internal space links

https://git.nurdspace.lan/esp8266/led_example/source/

External

SDK documentation (all chineses) [DOCS](#)

VM file [\[1\]](#) Password: i90l

Forum about the module <http://www.esp8266.com/>

seedstudio.com/depot/WiFi-Serial-Transceiver-Module-w-ESP8266-p-1994.html

ESP8266 ROM Bootloader utility <https://github.com/themadinventor/esptool>

Datasheet

[English Datasheet](#)

[http://www.seeedstudio.com/document/pdf/ESP8266%20Specifications\(Chinese\).pdf](http://www.seeedstudio.com/document/pdf/ESP8266%20Specifications(Chinese).pdf) (Chinese)

Introduction

Yue Xin intelligent high performance wireless connectivity platform --ESCP SOC, designers bring the Gospel to the mobile platform, it At the lowest cost to provide maximum usability for WiFi capabilities embedded in other systems offer unlimited possibilities.

Technical Overview

ESP8266 is a complete and self-contained Wi-Fi network solutions that can carry software applications, or through Another application processor uninstall all Wi-Fi networking capabilities. ESP8266 when the device is mounted and as the only application of the application processor, the flash memory can be started directly from an external Move. Built-in cache memory will help improve system performance and reduce memory requirements. Another situation is when wireless Internet access assume the task of Wi-Fi adapter, you can add it to any microcontroller-based design, the connection is simple, just by SPI / SDIO interface or central processor AHB bridge interface. Processing and storage capacity on ESP8266 powerful piece, it can be integrated via GPIO ports sensors and other applications specific equipment to achieve the lowest early in the development and operation of at least occupy system resources. The ESP8266 highly integrated chip, including antenna switch balun, power management converter, so with minimal external circuitry, and includes front-end module, including the entire solution designed to minimize the space occupied by PCB. The system is equipped with ESP8266 manifested leading features are: energy saving VoIP quickly switch between the sleep / wake patterns, with low-power operation adaptive radio bias, front-end signal processing functions, troubleshooting and radio systems coexist characteristics eliminate cellular / Bluetooth / DDR / LVDS / LCD interference.

Characteristics

802.11 b / g / n

Wi-Fi Direct (P2P), soft-AP

Built-in TCP / IP protocol stack

Built-in TR switch, balun, LNA, power amplifier and matching network

Built-in PLL, voltage regulator and power management components

802.11b mode + 19.5dBm output power

Built-in temperature sensor

Support antenna diversity

off leakage current is less than 10uA

Built-in low-power 32-bit CPU: can double as an application processor

SDIO 2.0, SPI, UART

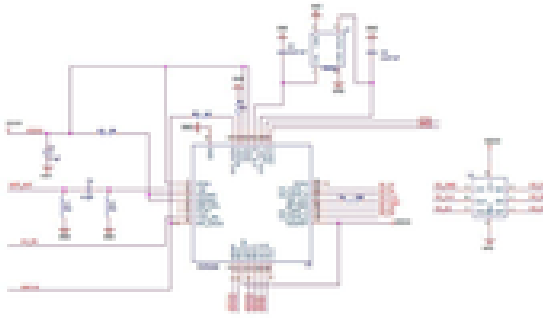
STBC, 1x1 MIMO, 2x1 MIMO

A-MPDU, A-MSDU aggregation and the 0.4 Within wake

2ms, connect and transfer data packets

standby power consumption of less than 1.0mW (DTIM3)

Schema



Ultra-low power technology

ESP8266 specifically for mobile devices, wearable electronics and networking applications design and make the machine to achieve the lowest energy consumption, together with several other patented technology. This energy-efficient construction in three modes: active mode, sleep mode and deep sleep mode type. When ESP8266 using high-end power management technology and logic systems to reduce non-essential functions of the power conversion regulate sleep patterns and work modes, in sleep mode, it consumes less than the current 12uA, is connected, it consumes less power to 1.0mW (DTIM = 3) or 0.5mW (DTIM = 10). Sleep mode, only calibrated real-time clock and watchdog in working condition. Real-time clock can be programmed to wake ESP8266 within a specific period of time. Through programming, ESP8266 will automatically wake up when detected certain to happen. ESP8266 automatic wake-up in the shortest time, this feature can be applied to the SOC for mobile devices, so before you turn Wi- Fi SOC are in a low-power standby mode. To meet the power requirements of mobile devices and wearable electronics products, ESP8266 at close range when the PA output power can be reduced through software programming to reduce overall power consumption in order to adapt to different applications.

Maximum integration

ESP8266 integrates the most critical components on the board, including power management components, TR switch, RF balun, a peak power of + 25dBm of PA, therefore, ESP8266 only guarantee the lowest BOM cost, and easy to be embedded in any system. ESP8266 BOM is the only external resistors, capacitors, and crystal.

ESP8266 application subject

Smart Power Plug

Home Automation

mesh network

industrial wireless control

Baby Monitor

Network Camera

sensor networks

wearable electronics

wireless location-aware devices

Security ID tag

wireless positioning system signals

Specifications

Power

The following data are based on a 3.3V power supply, ambient temperature 25C and use the internal regulator measured. [1] All measurements are made in the absence of the SAW filter, the antenna interface is completed. [2] all transmit data based on 90% duty cycle, continuous transmission mode in the measured.

Mode	Min	Typical	Max	Units
802.11b, CCK 1Mbps, POUT=+19.5dBm		215		mA
802.11b, CCK 11Mbps, POUT=+18.5dBm		197		mA
802.11g, OFDM 54Mbps, POUT=+16dBm		145		mA
802.11n, MCS7, POUT =+14dBm		135		mA
802.11b, packet size of 1024 bytes, -80dBm		60		mA
802.11b, packet size of 1024 bytes, -70dBm		60		mA
802.11b, packet size of 1024 bytes, -65dBm		62		mA
Standby		0.9		uA
Deep sleep		10		mA
Saving mode DTIM 1		1.2		mA
Saving mode DTIM 3		0.86		mA
Shutdown		0.5		uA

RF specifications

The following data is at room temperature, the voltage of 3.3V and 1.1V, respectively, when measured

Description	Min	Typical	Max	Units
Input Frequency	2412		2484	MHz
Input resistance		50		Ω
Input reflection			-10	dB
At 72.2Mbps, PA output power	14	15	16	dBm
11b mode, PA output power	17.5	18.5	19.5	dBm
Sensitivity				
CCK, 1Mbps <small>[OBJ]</small>		-98		dBm
CCK, 11Mbps <small>[OBJ]</small>		-91		dBm
6Mbps (1/2 BPSK) <small>[OBJ][OBJ]</small>		-93		dBm
54Mbps (3/4 64-QAM) <small>[OBJ][OBJ]</small>		-75		dBm
HT20, MCS7 (65Mbps, 72.2Mbps) <small>[OBJ][OBJ][OBJ]</small>		-71		dBm
Adjacent suppression				
OFDM, 6Mbps		37		dB

OFDM, 54Mbps		21		dB
HT20, MCS0		37		dB
HT20, MCS7		20		dB

CPU and memory

CPU Interface

The chip embedded in an ultra-low-power 32-bit micro-CPU, with 16 compact mode. Can be connected to the CPU via the following interfaces:

connecting storage controllers can also be used to access external code memory RAM / ROM interface (iBus)

Also attached storage controller data RAM interface (dBus)

Access Register of AHB interface

JTAG debug interface

Storage Controller

Storage controller contains ROM and SRAM. CPU can iBus, dBus and AHB interface to access the storage controller. Any one of these interfaces can apply for access to ROM or RAM cells, memory arbiter to determine the running order in the order of arrival.

AHB and AHB module

AHB module acts as arbiter, through the MAC, and SDIO host CPU control AHB interface. Since sending Address different, AHB data requests may arrive the following two slaves in one: APB module, or flash memory controller (usually in the case of off-line applications) to the received request is a high speed memory controllers often request, APB module receives register access is often Request. APB module acts as a decoder, but only you can access the ESP8266 main module programmable registers. Since the sending address different, APB request may reach the radio receiver, SI / SPI, hosts SDIO, GPIO, UART, real-time clock (RTC), MAC or digital baseband.

Interface

ESP 8266 contains multiple analog and digital interfaces, as follows:

Main SI / SPI control (optional)

Main Serial Interface (SI) can run at two, three, four-wire bus configuration, is used to control the EEPROM or other I2C / SPI devices. Multiple devices share the two-wire I2C bus. Multiple SPI devices to share the clock and data signals, and according to the chip select, each controlled by software alone GPIO pins. SPI can be used to control external devices, such as serial flash, audio CODEC or other slave devices, installation, effectively giving it three different pins, making it the standard master SPI device.

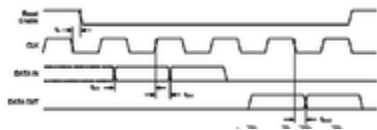
SPI_EN0

SPI_EN1

SPI_EN2

SPI slave is used as the primary interface, giving SPI master and slave SPI support. In the built-in applications, SPI_EN0 is used as an enable signal, the role of external serial flash, download firmware and / or MIB data to baseband. In host-based applications, the firmware and you can choose one MIB data downloaded via the host interface both. This pin is active low when not should be left unconnected. SPI_EN1 often used for user applications, such as controlling the built-in applications or external audio codec sensor ADC. This pin is active low when not should be left unconnected. SPI_EN2 often used to control the EEPROM, storing individual data

(individual data), such as MIB information, MAC address, and calibration data, or for general purposes. This pin is active low when not should be left unconnected.



General Purpose IO

A total of up to 16 GPIO pins. The firmware can assign them different functions. Each GPIO can be configured internal pullup / pulldown resistors available software registers sampled input, triggering edge or level CPU interrupt input, trigger level wake-up interrupt input, open-drain or complementary push-pull output drivers, software register output source or sigma-delta PWM DAC. These pins are multiplexed with other functions, such as the main interface, UART, SI, Bluetooth co-existence and so on.

Digital IO pins

Digital IO pad is two-way, three states. It includes a three-state control input and output buffers. In addition, for low-power operation, IO can be set to hold state. For example, when we reduce the chip's power consumption, all the output enable signal can be set to maintain a low-power state. Hold function can be selectively implanted IO in need. When the IO help internal and external circuit driving, hold function can be used to hold last state. Hold function to pin introduce some positive feedback. Therefore, the external drive pin must be stronger than the positive feedback. However, the required driving force size is still small, in the 5uA of.

Variables	Symbol	Min	Max	Units
Input Low Voltage	Vil	-0.3	0.25xV10	V
Input High Voltage	Vih	0.75xV10	3.6	V
Input leakage current	IIL	-	50	nA
Output Low Voltage	VOL	-	0.1xV10	V
Output High Voltage	VOH	0.8xV10	-	V
Input pin capacitance	Cpad	-	2	pF
VDDIO	V10	1.7	3.6	V
Current	I _{max}	-	12	mA
Temperature	T _{amb}	-20	100	C

All digital IO pins must add an overvoltage protection circuit (snap back circuit) between the pin and ground. Usually bounce (snap back) voltage is about 6V, while maintaining the voltage is 5.8V. This prevents excessive voltage and generating ESD. Diodes also avoid reverse voltage output devices.

Firmware and software tools development kit

The firmware running on the ROM and SRAM chip, when the device is awake, firmware via SDIO sector Download the instructions from the host side. Firmware is fully compliant with 802.11 b / g / n / e / i WLAN MAC protocol and Wi-Fi Direct specification only supports basic services unit distributed control function (DCF) under (BSS) operation, but also follow the latest Wi-Fi P2P

protocol to support P2P groups operating (P2P group operation). Low level protocol functions automatically run by ESP8266, such as

RTS / CTS

Confirm

fragmentation and reassembly

polymerization

frame package (802.11h / [RFC 1042](#))

automatic beacon monitoring / scanning

P2P WiFi direct

With P2P discovery procedures, passive or active scanning once in the host command start, it will be done automatically. Perform power management, interaction with the host at least, this way, the task of effectively minimized.

Features

Laboratory features of the software developer's kit is as follows:

802.11 b / g / n / d / e / i / k / r support

Wi-Fi Direct (P2P) support

P2P discovery, P2P group master mode (Group Owner mode), P2P Power Management

Infrastructure Network (Infrastructure BSS) station (Station) mode / P2P mode / SoftAP mode

hardware accelerator

CCMP (CBC-MAC, counting mode)

TKIP (MIC, RC4) o WAPI (SMS4)

WEP (RC4)

CRC

WPA / WPA2 and WPS support

Other 802.11i security features:

Pre-Certification

TSN

Open interfaces ♣ prepared for a variety of upper EAP authentication methods, such as:

TLS

PEAP

LEAP

SIM

AKA

802.11n support (2.4GHz)

Support MIMO 1x1 and 2x1, STBC, A-MPDU and A-MSDU aggregation, 0.4s guard interval

WMM saving U-APSD

use with multi-queue QoS management, in line with 802.11e standard multimedia data traffic optimization methods

Follow the UMA, and certified by UMA

802.11h / RFC1042 frame encapsulation

hash DMA data transfer operation, the CPU usage to a minimum

antenna diversity and choice (by software management hardware)

the clock / power gating and follow the 802.11 standard power management combined, according to the current connection, enter OK dynamically adjusted to achieve the lowest energy

consumption

ratio can be adjusted to set an optimum algorithm for the missing data and the Tx power transfer rate based on the actual SNR and packet

Rate

MAC layer automatic retransmission and automatic response, to avoid packet loss occurs when the host is running slow

seamless roaming support

Configurable packet traffic arbitration and tailored, based on the slave processor design combining a series of Bluetooth chip vendors to provide flexible and precise time-Bluetooth coexistence support

support dual / single antenna Bluetooth coexistence with syncing WiFi / Bluetooth capability

Power Management

Chip can tune into the following states:

off (OFF): CHIP_PD pin is in a low power state. RTC failure. All registers are emptied.

deep sleep (DEEP_SLEEP): RTC open, other parts of the chip are closed. RTC internal recovery memory to save the basic WiFi connection information.

sleep (SLEEP): Only RTC running. Crystal oscillator stops. Any part of the wake (MAC, host, RTC timer, external interrupt) will make the wake of the chip.

Wake (WAKEUP): In this state, the system from a sleep state to start (PWR) status. Crystal oscillator and PLL are converted enabled state.

on state (ON): High-speed clock can run, And sent to each clock control register is enabled Modules. Each module, including the CPU, including the implementation of relatively low-level clock gating. When the system works, you can WAITI instructions to turn off the CPU's internal clock.



Clock Management

High Frequency Clock

ESP8266 on high frequency clock is used to drive two Tx and Rx mixer, which is generated by the internal oscillator and an external oscillator. Crystal frequency between 26MHz to 52MHz float. Although the internal crystal oscillator of the calibration range of the crystal so that the clock generator to meet the conditions, but in general, the crystal quality is still obtained a proper phase noise factors to be considered. When the crystal is used, or because of the frequency offset, rather than the best choice for quality, the maximum capacity of the data processing system and will reduce the sensitivity of the wifi. Please refer to the following instructions to measure the frequency offset.

Variables	Symbol	Min	Max	Units
Frequency	Fxo	52	MHz	
Load capacitance	Cl		32	pF

Dynamic capacitance	Cm	2	5	pF
Serial resistance	Rs	0	65	Ω
Frequency tolerance	Fxo	-15	15	ppm
Frequency vs Temperature (-25C ~ 75C)	Fxo,Temp	-15	15	ppm

External Reference Requirements

At 26MHz external clock frequency between 52MHz. In order to make a well-functioning radio receiver, the clock will Must have the following characteristics:

Variables	Symbol	Min	Max	Units
Clock amplitude	Vxo	0.2	1	Vpp
External clock accuracy	Fxo,EXT	-15	15	ppm
Phase Noise @ 1kHz offset, 40MHz clock			-120	dBc/Hz
Phase Noise @ 10kHz offset, 40MHz clock			-130	dBc/Hz
Phase Noise @ 100kHz offset, 40MHz clock			-138	dBc/Hz

Radio receivers

ESP8266 radio receiver mainly includes the following modules:

2.4GHz receiver

2.4GHz transmitter

High-speed clock generator and crystal oscillator

Real-time clock

bias and regulators

Power Management

Channel Frequency

According IEEE802.11bgn standard, RF transceiver supports the following channels:

Channel	Freq.	Channel	Freq.
1	2412	8	2447
2	2417	9	2452
3	2422	10	2457
4	2427	11	2462
5	2432	12	2467
6	2437	13	2472
7	2442	14	2484

2.4GHz receiver

2.4GHz RF signal receiver down into quadrature baseband signal, with two high-resolution, high-speed ADC and the latter into a digital signal. In order to accommodate different signal channels, a radio receiver integrated RF filters, automatic gain control (AGC), DC offset compensation circuit and a baseband filter.

2.4GHz transmitter

2.4 GHz transmitter orthogonal frequency baseband signals up to 2.4GHz, using high-power

CMOS power amplifier to drive the antenna. Further use of the digital calibration improves the linearity of the power amplifier to achieve the average power of + 19dBm in 802.11b transmission, the transmission reaches + 16dBm 802.11n average power, features super. To offset defects in the radio receiver is also calibrated by other measures such as:

carrier leakage

I / Q phase matching, and

baseband nonlinear

This will reduce the time and equipment required for testing.

Clock generator

The clock generator generates the receiver and transmitter 2.4GHz clock signal all of its components are integrated on the chip, Include:

inductor

varactor

closed-loop filter

Clock generator contains a built-in calibration circuit and self-test circuitry. Clock phase and quadrature phase noise through the optimal calibration algorithm processing patent on the chip, in order to ensure that the receiver and transmitter to achieve the best performance.

AT Commands

Tnx to <http://www.electrodragon.com/w/Wi07c>

Format

Baud rate at 57600

x is the commands

Set	Inquiry	Test	Execute
AT+<x>=<...>	AT+<x>?	AT+<x>=?	AT+<x>
AT+CWMODE=<mode>	AT+CWMODE?	AT+CWMODE=?	-
Set the network mode	Check current mode	Return which modes supported	-

Commands

carefully there are must be no any spaces between the " and IP address or port

Com mand s	Descri ption	Typ e	Set/Execute	Inquir y	test	Param eters	Examples
AT+RST	restart the module	basic	-	-	-	-	
AT+CW MODE	wifi mode	wifi	AT+CWMODE= <mode>	AT+CWM ODE?	AT+CWM ODE=?	1= Sta, 2= AP, 3=both	
AT+CWJ AP	join the AP	wifi	AT+ CWJAP =<ssid>,< pwd >	AT+ CWJAP?	-	ssid = ssid, pwd = wifi	

Command	Description	Type	Set/Execute	Inquiry	test	Parameters	Examples
						password	
AT+CWLAP	list the AP	wifi	AT+CWLAP				
AT+CWQAP	quit the AP	wifi	AT+CWQAP	-	AT+CWQAP=?		
AT+CWSAP	set the parameters of AP	wifi	AT+CWSAP=<ssid>,<pwd>,<chl>,<ecn>	AT+CWSAP?		ssid, pwd, chl = channel, ecn = encryption	Connect to your router: : AT+CWJAP="YOURSSID", "helloworld"; and check if connected: AT+CWJAP?
AT+CIPSTATUS	get the connection status	TCP/IP	AT+CIPSTATUS				
AT+CIPSTART	set up TCP or UDP connection	TCP/IP	1)single connection (+CIPMUX=0) AT+CIPSTART=<type>,<addr>,<port>; 2) multiple connection (+CIPMUX=1) AT+CIPSTART=<id><type>,<addr>,<port>	-	AT+CIPSTART=?	id = 0-4, type = TCP/UDP, addr = IP address, port= port	Connect to another TCP server, set multiple connection first: AT+CIPMUX=1; connect: AT+CIPSTART=4,"TCP","X1.X2.X3.X4",9999
AT+CIPSEND	send data	TCP/IP	1)single connection(+CIPMUX=0) AT+CIPSEND=<length>; 2) multiple connection (+CIPMUX=1) AT+CIPSEND=		AT+CIPSEND=?		send data: AT+CIPSEND=4,15 and then enter the data

Com mand s	Descri ption	Typ e	Set/Execute	Inquir y	test	Param eters	Examples
			<id>,<length>				
AT+CIPC LOSE	close TCP or UDP connecti on	TCP/I P	AT+CIPCLOSE= <id> or AT+CIPCLOSE			AT+CIPC LOSE=?	
AT+CIFS R	Get IP address	TCP/I P	AT+CIFSR			AT+ CIFSR=?	
AT+ CIPMUX	set mutiple connecti on	TCP/I P	AT+ CIPMUX=<mo de>	AT+ CIPMUX?		0 for single connecti on 1 for mutiple connecti on	
AT+ CIPSERV ER	set as server	TCP/I P	AT+ CIPSERVER= <mode>[,<por t>]			mode 0 to close server mode, mode 1 to open; port = port	turn on as a TCP server: AT+CIPSERVER=1,8888, check the self server IP address: AT+CIFSR=?
+IPD	received data						



Ultrasonic Ranging Module HC - SR04

Product features:

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules includes ultrasonic transmitters, receiver and control circuit. The basic principle of work:

- (1) Using IO trigger for at least 10us high level signal,
- (2) The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.
- (3) IF the signal back, through high level , time of high output IO duration is the time from sending ultrasonic to returning.

Test distance = (high level time×velocity of sound (340M/S) / 2,

Wire connecting direct as following:

- 5V Supply
- Trigger Pulse Input
- Echo Pulse Output
- 0V Ground

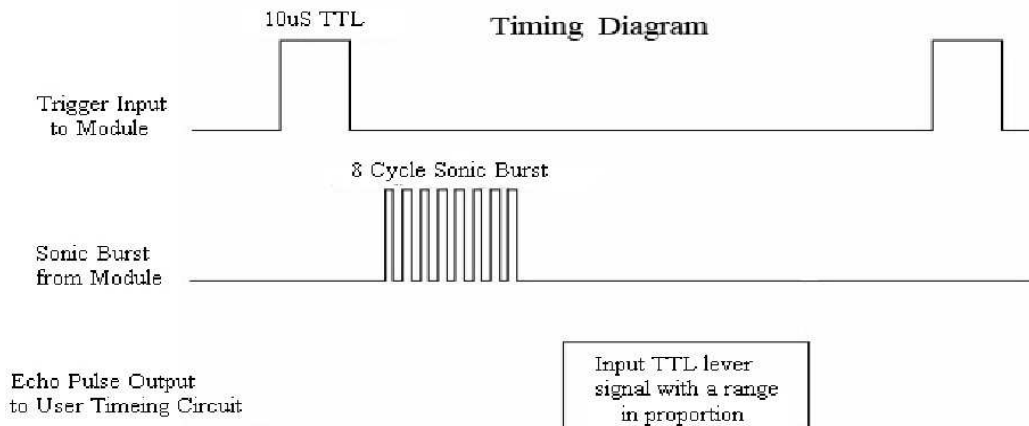
Electric Parameter

Working Voltage	DC 5 V
Working Current	15mA
Working Frequency	40Hz
Max Range	4m
Min Range	2cm
MeasuringAngle	15 degree
Trigger Input Signal	10uS TTL pulse
Echo Output Signal	Input TTL lever signal and the range in proportion
Dimension	45*20*15mm



Timing diagram

The Timing diagram is shown below. You only need to supply a short 10uS pulse to the trigger input to start the ranging, and then the module will send out an 8 cycle burst of ultrasound at 40 kHz and raise its echo. The Echo is a distance object that is pulse width and the range in proportion. You can calculate the range through the time interval between sending trigger signal and receiving echo signal. Formula: $\mu\text{S} / 58 = \text{centimeters}$ or $\mu\text{S} / 148 = \text{inch}$; or: the range = high level time * velocity (340M/S) / 2; we suggest to use over 60ms measurement cycle, in order to prevent trigger signal to the echo signal.



Attention:

- The module is not suggested to connect directly to electric, if connected electric, the GND terminal should be connected the module first, otherwise, it will affect the normal work of the module.
- When tested objects, the range of area is not less than 0.5 square meters and the plane requests as smooth as possible, otherwise ,it will affect the results of measuring.

www.ElecFreaks.com



Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

[SparkFun Electronics:](#)

[SEN-13959](#)

Datasheet

I2C 1602 Serial LCD Module



Product features:

The I2C 1602 LCD module is a 2 line by 16 character display interfaced to an I2C daughter board. The I2C interface only requires 2 data connections, +5 VDC and GND to operate

For in depth information on I2C interface and history, visit: <http://www.wikipedia/wiki/i2c>

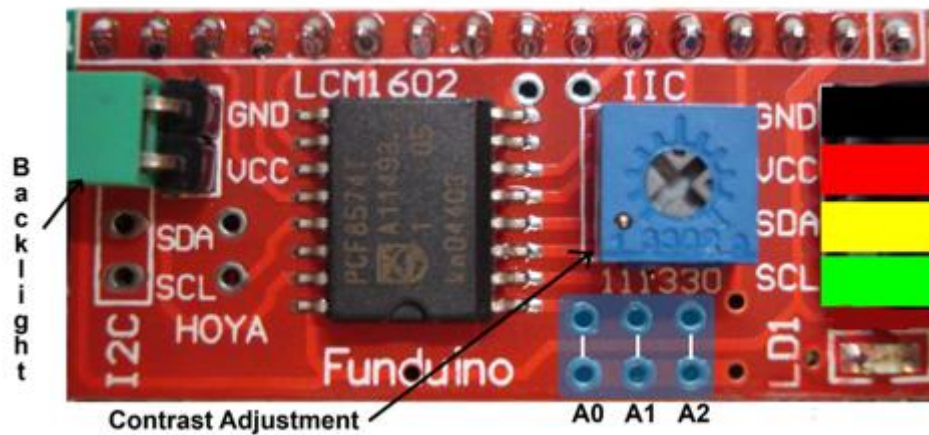
Specifications:

I2C Address Range	2 lines by 16 character 0x20 to 0x27 (Default=0x27, addressable)
Operating Voltage	5 Vdc
Backlight	White
Contrast	Adjustable by potentiometer on I2c interface
Size	80mm x 36mm x 20 mm
Viewable area	66mm x 16mm

Power:

The device is powered by a single 5Vdc connection.

Pinout Diagram:



Pin/Control Descriptions:

Pin #	Name	Type	Description
1	GND	Power	Supply & Logic ground
2	VCC	Power	Digital I/O 0 or RX (serial receive)
3	SDA	I/O	Serial Data line
4	SCL	CLK	Serial Clock line
A0	A0	Jumper	Optional address selection A0 - see below
A1	A1	Jumper	Optional address selection A1 - see below
A2	A2	Jumper	Optional address selection A2 - see below
Backlight		Jumper	Jumpered - enable backlight, Open - disable backlight
Contrast		Pot	Adiust for best viewing

Addressing:

A0	A1	A2	Address
Open	Open	Open	0x27
Jumper	Open	Open	0x26
Open	Jumper	Open	0x25
Jumper	Jumper	Open	0x24
Open	Open	Jumper	0x23
Jumper	Open	Jumper	0x22
Open	Jumper	Jumper	0x21
Jumper	Jumper	Jumper	0x20

Software:

Download the required LCD Arduino™ library for this device from:

<http://www.circuitattic.com/downloads/category/3-sample-code.html?download=9%3Aanother-i2c-library-easier-to-use>

Replace current liquid crystal library found in the Arduino library directory with the above
(Note: If you use the examples included with the library, be sure to change address to 0x27)

Simple example using library above.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#if defined(ARDUINO) && ARDUINO >= 100
#define printByte(args) write(args);
#else
#define printByte(args) print(args,BYTE);
#endif
LiquidCrystal_I2C lcd(0x27,16,2); // set the LCD address to 0x27 for a
//chars and 2 line display
void setup()
{
    lcd.init(); // initialize the lcd
    lcd.backlight();
    lcd.clear();
    delay(100);
    for(int i = 0; i < 3; i++)
    {
        lcd.backlight();
        delay(250);
        lcd.noBacklight();
        delay(250);
    }
    lcd.backlight();
}

void loop()
{
    int x=0;
    lcd.clear();
    lcd.setCursor(2,0); //Start at character 0 on line 0
    lcd.print("Hello World");
    lcd.setCursor(0,1); //Start at character 0 on line 1
    lcd.print(" opencircuit.nl");
    delay(3000); //Wait 3 seconds
    lcd.clear();
    lcd.setCursor(0,0); //Start at character 0 on line 0
    lcd.print("Cursor Blink");
    lcd.blink();
    delay(2000);
    lcd.setCursor(0,0);
    lcd.print("Cursor noBlink");
    lcd.noBlink();
    delay(2000);
}
```