PROTOTIPE *MONITORING* PEMAKAIAN LISTRIK *VIA ANDROID*

PROYEK AKHIR

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

Disusun Oleh :

Dwi Retno Angraini	NIM : 003 15 08
Mochamad Rivai	NIM : 003 15 16

POLITEKNIK MANUFAKTUR NEGERI
BANGKA BELITUNG
TAHUN 2018

LEMBAR PENGESAHAN

PROTOTIPE *MONITORING* PEMAKAIAN LISTRIK *VIA ANDROID*

Oleh :

Dwi Retno Angraini /0031508

Mochamad Rivai /0031516

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

Menyetujui,

Pembimbing 1	Pembimbing 2
Yudhi, M.T.	Surojo, M.T.

Penguji 1	Penguji 2	Penguji 3
Eko Sulistyo, M.T.	Aan Febriansyah, M.T.	Ocsirendi, M.T.

PERNYATAAN BUKAN PLAGIAT

Yang bertanda tangan dibawah ini :

Nama Mahasiswa 1 : Dwi Retno Angraini NIRM: 0031508
Nama Mahasiswa 2 : Mochamad Rivai NIRM: 0031508

Dengan judul : " PROTOTIPE *MONITORING* PEMAKAIAN LISTRIK *VIA ANDROID*"

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

Sungailiat, 16 Juni 2018

Nama Mahasiswa	Tanda Tangan
1. Dwi Retno Angraini	.....
2. Mochamad Rivai	.....

ABSTRAK

Energi listrik adalah salah satu kebutuhan utama masyarakat. PT PLN menggunakan kWh meter sebagai perhitungan pemakaian energi listrik yang digunakan oleh pelanggan. Seiring dengan kemajuan teknologi kWh meter konvensional telah diperbaharui menjadi kWh meter prabayar. Akan tetapi terdapat kekurangan pada kWh meter prabayar yang dirasakan oleh masyarakat, misalnya ketika pengguna lupa untuk mengisi saldo pulsa maka listrik akan mati tentu saja ini sangat merepotkan dan tidak efisien karna pengguna harus sering memantau sisa pulsa yang ada pada kWh meter prabayar agar listrik tidak padam karna kehabisan pulsa. Untuk mengatasi masalah ini dan mempermudah pelanggan dalam mengontrol pemakaian listrik maka dibuatlah "Prototipe Monitoring Pemakaian Listrik Via Android". Pada alat ini terdapat 2 bagian yaitu, bagian pertama box kontrol yang berisikan rangkaian-rangkaian kontrol yang berfungsi sebagai pengontrolan mikrokontroller komponen-komponen yang digunakan pada alat ini. Pada box kedua yang memiliki fungsi sebagai terminal output beban yang dikontrol baik itu secara otomatis melalui mikrokontroller dan secara manual yang diakses melalui android. Dari hasil uji coba yang dilakukan diperoleh hasil data dan diperoleh hasil bahwa untuk mempermudah pelanggan dalam memonitoring terdapat beberapa cara salah satunya dengan mengakses melalui android. Serta untuk menghemat pemakaian listrik, pelanggan dapat menggunakan mode manual atau otomatis sesuai kebutuhan atau yang diinginkan pelanggan melalui android.

Kata Kunci : KWh meter prabayar, Android, Mode manual dan otomatis.

ABSTRACT

Electric energy is one of the people's needs, PT PLN uses kWh meter as the calculation of electric energy consumption used by the customer. Along with advances in conventional kWh meter technology has been refurbished into prepaid kWh meter felt by the community, for example when the user forgot to fill the balance of the pulse then the electricity will die, of course this is very inconvenient and inefficient because the user must of ten monitor the remaining pulse in the pre-paid kWh meter , so that electricity is not extinguished due to run out of pulses. To overcome this problem and make it easier for customers to control the electricity usage then made "Prototipe Monitoring Pemakaian Listrik Via Android". In this tool there are two parts, the first box control box containing control circuits that serve as a control microcontroller component used in this tool. In the second box which has a function as a controlled load output terminal either through the microcontroller and manually accessed via android from the results of experiments performed detained data and obtained result that to facilitate customers in monitoring there are several ways one of them by accessing through android. As well as to save electricity consumption, customers can use manual or automatic mode according to the needs or the customer wants through android.

Keywords : Prepaid kWh meter, Android, Manual and aoutomatice mode.

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat ALLAH Subhanahu Wa Ta'ala atas berkat rahmat dan hidayah-Nya jualah, sehingga pada akhirnya penulis dapat menyelesaikan karya tulis proyek akhir ini dengan baik.

Karya Tulis Proyek Akhir ini disusun sebagai salah satu persyaratan dan kewajiban mahasiswa untuk menyelesaikan kurikulum program pendidikan Diploma III di Politeknik Manufaktur Negeri Bangka Belitung.

Penulis mencoba untuk menerapkan ilmu pengetahuan yang telah didapatkan selama 3 tahun mengecap pendidikan di Politeknik Manufaktur Negeri Bangka Belitung dan pengelaman yang penulis dapatkan selama melaksanakan Program Kerja Lapangan yang dilaksanaka sebelum melaksanakan Proyek Akhir ini.

Pada kesempatan ini penulis menyampaikan rasa terimakasih yang sebesar–besarnya kepada orang-orang yang telah berperan sehingga dapat terselesaikannya Proyek Akhir ini, sebagai berikut :

1. Keluarga besar (Ayah, Ibu, Kakak-kakak penulis, Nenek, Kakek, Teman-teman dll) yang selalu senantiasa memberikan kasih sayang, doa, dukungan moril maupun materi dan semangat.
2. Bapak Sugeng Ariyono, M.Eng., Ph.D selaku Direktur Politeknik Manufaktur Negeri Bangka Belitung.
3. Bapak Yudhi, M.T. selaku pembimbing I yang telah meluangkan banyak waktu, tenaga, pikiran dan yang paling penting Motivasi didalam memberikan pengarahan dalam penulisan karya tulis Proyek Akhir ini dan Bapak Surojo, M.T. selaku dosen pembimbing II yang telah banyak memberi saran-saran dan solusi dari masalah-masalah yang penulis hadapi selama proses penyusunan karya tulis Proyek Akhir ini.
5. Seluruh staf pengajar di Politeknik Manufaktur Negeri Bangka Belitung.

6. Rekan-rekan mahasiswa Politeknik Manufaktur Negeri Bangka Belitung yang telah banyak membantu selama menyelesaikan Proyek Akhir.
7. Rekan-rekan sepermainan di tempat penulis tinggal, yang selalu memberikan semangat mental dan moril disaat penulis jenuh.
8. Pihak-pihak lain yang telah memberikan bantuan secara langsung maupun tidak langsung dalam pembuatan Proyek Akhir ini yang tidak dapat disebutkan satu persatu.

Penulis menyadari bahwa penulisan 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 dating 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 karya tulis Proyek Akhir ini dan penulis dengan senang hati menerima saran dan kritik yang membangun dari pembaca.

Besar harapan penulis semoga makalah tugas akhir dan alat yang dibuat dapat memberikan manfaat bagi pihak yang berkepentingan pada khususnya dan bagi perkembangan ilmu teknologi pada umumnya .

Sungailiat, 26 Juni 2018

Penulis

DAFTAR ISI

BAB III METODE PELAKSANAAN

BAB IV PEMBAHASAN

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR LAMPIRAN

BAB I
PENDAHULUAN

1.1 Latar Belakang

Energi listrik adalah salah satu kebutuhan utama masyarakat. Dengan berkembangnya teknologi yang sangat pesat secara otomatis kebutuhan energi listrik akan terus meningkat, terutama dalam penggunaan peralatan listrik untuk rumah tangga yang bisa dirasakan manfaatnya oleh masyarakat untuk meringankan pekerjaan dan mempermudah aktivitas setiap hari. [1]

PT PLN menggunakan *kWh meter* sebagai suatu media perhitungan pemakaian energi listrik yang digunakan oleh pemilik rumah. Seiring dengan kemajuan teknologi *kWh meter* konvensional telah diperbaharui menjadi *kWh meter* prabayar. Penggunaan *kWh meter* prabayar ini pengguna tidak perlu lagi membayar tagihan pemakaian listrik ke kantor PT PLN, pengguna hanya perlu membeli token. Akan tetapi terdapat kekurangan pada *kWh meter* prabayar yang dirasakan oleh masyarakat, misalnya ketika pengguna lupa untuk mengisi saldo pulsa maka listrik akan mati tentu saja ini sangat merepotkan dan tidak efisien karna pengguna harus sering memantau sisa pulsa yang ada pada *kWh meter* prabayar agar listrik tidak padam karna kehabisan pulsa.[2]

Berdasarkan hal tersebut, diperlukan adanya suatu sistem monitoring yang dapat memonitoring penggunaan listrik. Monitoring ini akan memudahkan pengguna untuk mengetahui saldo pada *kWh meter* serta pengguna akan dapat berhemat pemakaian listrik pada saat beban maksimal yang telah ditentukan tercapai maka beban terbesar pada stop kontak akan dimatikan. Alat ini dapat menghitung daya yang terpakai dan daya yang tersisa yang akan ditampilkan pada *LCD* dan layar *smartphone* selain itu alat ini juga memiliki akses untuk mematikan stop kontak secara otomatis dimana akan memutuskan aliran listrik pada stop kontak yang memiliki beban terbesar diantara stop kontak lainnya yang berfungsi agar pengguna dapat menekan pemakaian energi listrik. [6]

1.2 Rumusan Masalah

Rumusan masalah yang dibuat pada proyek akhir ini adalah:

1. Bagaimana pengambilan data perubahan arus dan tegangan dari sensor arus serta tegangan.
2. Bagaimana mikrokontroler memproses data dari semua masukan menjadi keluaran berupa nilai daya yang terpakai dan daya yang tersisa dalam bentuk *kWh.*
3. Bagaimana cara dapat mengontrol pemakaian yang berlebihan dan memutuskan aliran listrik pada stop kontak yang memiliki beban besar.
4. Bagaimana cara mengirim data dan kontrol pemutusan aliran listrik dari mikrokontroler ke *smartphone.*

1.3 Batasan Masalah

Batasan masalah dalam penyelesaian proyek akhir ini adalah *KWh meter* prabayar yang di keluarkan oleh PT PLN (Persero) tidak dapat dibongkar sehingga *kWh meter* prabayar dari PT PLN (Persero) tidak dapat digunakan dalam proyek akhir ini yang menyebabkan penulis harus membuat sendiri simulator *kWh meter* prabayar.

1.4 Tujuan Proyek Akhir

Tujuan pengerjaan proyek akhir ini adalah:

1. Menghasilkan perhitungan daya beban yang didapatkan dari hasil pengambilan data perubahan yang dihasilkan dari PZEM-004T berupa nilai arus beban dan nilai tegangan beban.
2. Menghasilkan data yang berupa sisa pulsa dan daya pemakaian yang diproses oleh mikrokontroller ke *smartphone.*
3. Membuat program otomatis dan fitur kontrol *manual* yang bisa diakses melalui *smartphone* yang digunakan sebagai pemutus aliran listrik yang mengalir ke beban, dengan beban yang memiliki daya terbesar diantara beban lainnya.

BAB II
DASAR TEORI

2.1 Deskripsi Umum Energi Listrik

Energi merupakan energi utama yang dibutuhkan bagi peralatan listrik atau energi yang tersimpan dalam arus listrik dengan satuan *ampere* (A) dan tegangan listrik dengan satuan *volt* (V) dan dengan satuan *time* (t) dengan ketentuan kebutuhan konsumsi daya listrik dengan satuan *watt* (W) Satuan besaran energi adalah *watt* jam atau kilo-*watt* jam. Energi listrik dinyatakan pada Persamaan 2.1.

$$W = V*I*t \quad \ldots\ldots(1)$$

Keterangan : W = Energi Listrik (*Watt*-jam)
V = Tegangan Listrik (*Volt*)
I = Arus Listrik (*Ampere*)
t = Waktu (jam)

Alat yang digunakan untuk mengukur energi listrik dinamakan *kWh meter*. *kWh meter* adalah alat penghitung pemakaian energi listrik. Alat ini bekerja menggunakan metode induksi medan magnet dimana medan magnet tersebut menggerakkan piringan yang terbuat dari alumunium. Pada piringan alumunium itu terdapat as yang mana as tersebut akan menggerakkan *counter digit* sebagai tampilan jumlah *kWh*-nya. *KWh meter* memiliki 3 kumparan yaitu 1 kumparan tegangan degan koil yang diameternya tipis dan 2 kumparan arus dengan koil yang diameternya tebal. Pada *kWh meter* juga terdapat magnet permanen yang tugasnya menetralkan piringan alumunium dari induksi medan magnet.

Dengan seiringnya perkembangan zaman dalam bidang teknologi, maka *kWh meter* sudah banyak dikembangkan, misalnya *kWh meter* prabayar. Untuk

kWh meter prabayar adalah *kWh meter* yang menggunakan *keypad*, *mikrokontroller* dan *LCD* sebagai *input*, proses dan *output*. Dalam proses yang dilakukan oleh *kWh meter* ini menggunakan *mikrokontroller*. Yang melakukan proses perhitungaan saldo *kWh*, pengaktifan *kWh* dan pengecekan terhadap penggunaan *kWh*.

2.2 Deskripsi Teori Daya Listrik

Satuan daya listrik dalam SI adalah Watt, yang didifinisikan sebagai berubahnya energi terhadap waktu dalam bentuk tegangan dan arus. Daya dalam watt diserap oleh suatu beban pada setiap saat sama dengan jatuh tegangan pada beban tersebut (*volt*) dikalikan dengan arus yang mengalir lewat beban (*Ampere*), atau Daya listrik terbagi menjadi tiga jenis, yaitu daya aktif, daya reaktif dan daya nyata :

1. Daya Aktif (Watt) Adalah Daya yang berupa daya kerja seperti daya mekanik, panas, cahaya, dan sebagainya. Daya aktif dinyatakan dalam satuan Watt (W). Rumus dari daya aktif adalah sebagai berikut : [2]

$$P = V \times I \times \cos \varphi \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(2)$$

Keterangan : P = Daya Aktif (Watt)
V = Tegangan Listrik (*Volt*)
I = Arus Listrik (*Ampere*)
Cos φ = Faktor Daya

2. Daya Reaktif (VAR) merupakan daya yang diperlukan oleh peralatan listrik yang bekerja dengan sistem elektromagnet. Daya reaktif dinyatakan dalam satuan VAR. Rumus dari daya reaktif adalah sebagai berikut : [2]

$$Q = V \times I \times \sin \varphi \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(3)$$

Keterangan : Q = Daya Reaktif (VAR)
V = Tegangan Listrik (*Volt*)
I = Arus Listrik (*Ampere*)
Sin φ = Faktor Reaktif

3. Daya Nyata (VA) Adalah penjumlahan vektor dari daya aktif dan reaktif. Daya ini dinyatakan dalam satuan VA. Rumus dari daya nyata adalah sebagai berikut : [2]

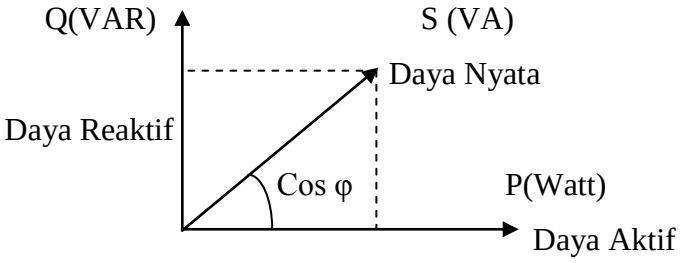
$$S = \sqrt{P^2 + Q^2} \quad \ldots\ldots(4)$$

Keterangan : S = Daya Semu (*VA*)
P = Daya Aktif (*Watt*)
Q = Daya Reaktif (*VAR*)

$$S = V \times I \quad \ldots\ldots(5)$$

Keterangan : S = Daya Semu (*VA*)
V = Tegangan (*Volt*)
I = Arus (*Ampere*)

Dari persamaan daya diatas terdapat Hubungan Daya dapat ditunjukkan pada gambar 2.1.



Gambar 2.1. Grafik hubungan daya aktif, reaktif dan daya nyata

2.3 *Android*

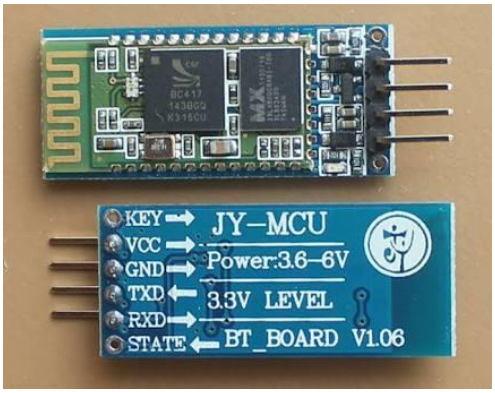
Pada *android* adalah salah satu *platform* sistem operasi yang digemari oleh masyarakat umum karena sifatnya yang *open source* sehingga memungkinkan pengguna untuk melakukan pengembangan. *Android* merupakan generasi baru *platform mobile* berbasis *linux* yang mencakup sistem operasi, *middleware*. Arsitektur *android* terdiri dari bagian-bagian seperti berikut : [3]

a. *Application dan widgets* : *layer* (lapisan) dimana pengguna hanya dapat berhubungan dengan aplikasi saja.
b. *Application Framework* : lapisan dimana para pengembang melakukan pembuatan aplikasi yang akan dijalankan di sistem operasi *android* dengan komponen-komponenya meliputi *views, contents provider, resoutce manager, notification manager, activity manager.*
c. *Libraries* : lapisan dimana fitur-fitur *android* yang berada diatas kernel meliputi *library C/C++* inti seperti Libc dan SSL.
d. *Android Run Timer* : lapisan yang membuat aplikasi *android* dapat dijalankan, yang dimana dalam prosesnya menggunakan implementasi *Linux* yang terbagi menjadi dua bagian yaitu *Core Libraries dan Dalvik virtual Machine.*
e. *Linux Kernel* : *Layer* yang berisi *file-file* sistem untuk mengatur *processing, memory, resource, driver,* dan sistem operasi *android* lainnya.

2.4 *Bluetooth HC-06*

Bluetooth HC-06 adalah sebuah modul *bluetooth* SPP (*Serial Port Protokol*) yang mudah digunakan untuk komunikasi serial dengan menggunakan *wireless* (nirkabel) yang mengkonversi *port* serial ke *Bluetooth. Bluetooth* ini memiliki komunikasi serial *UART* dalam penerimaan serta dalam pengiriman datanya. *Bluetooth HC-06* memungkinkan untuk dapat berkomunikasi secara langsung dengan mikrokontroler melalui jalur *pin* TX serta RX yang terdapat pada *pin out*-nya. *Bluetooth HC-06* hanya dapat dikonfigurasi sebagai *slave* tidak bisa digunakan sebagai *master*. *Data sheet* modul *bluetooth HC-06* dapat dilihat

pada halaman lampiran. Bentuk fisik dari *bluetooth HC-06* ditunjukkan pada gambar 2.2.



Gambar 2.2 *Bluetooth HC – 06* [3]

Bluetooth HC-06 memiliki sprsifikasi dalam penggunaannya antara lain :

a. Sensitivitas =80 dBm (*Typical*).
b. Daya transmit RF sampai dengan +4dBm.
c. Operasi daya rendah 1,8 V- 3,6 V *I/O.*
d. Kontrol *PIO.*
e. Antarmuka *UART* dengan *baudrate* yang dapat diprogram.

2.5 Modul Sensor PZEM-004T

Modul sensor PZEM-004T adalah modul sensor yang dapat mengukur tegangan, arus dan daya dengan menggunakan *Arduino* / ESP8266 / Raspberry Pi seperti *platform opensource*. Modul sensor ini melayani semua persyaratan dasar pengukuran modul sensor PZEM-004T sebagai papan terpisah. Dimensi fisik dari papan modul sensor PZEM-004T adalah 3,1 × 7,4 cm. Modul sensor PZEM-004T ini dibundel dengan kumparan trafo arus yang memiliki diameter 33mm. Bagian utama dari modul sensor PZEM-004T ini adalah chip SD3004 serta SDIC Microelectronics Co., Ltd. Modul sensor PZEM-004T ini juga dilengkapi dengan antarmuka komunikasi data serial TTL yang melalui *port serial* dapat dibaca serta bisa mengatur parameter yang relevan. Bentuk fisik dari modul PZEM-004T ditunjukan pada gambar 2.3.

Gambar 2.3 Modul sensor PZEM-004T [4]

2.5.1 Spesifikasi sensor PZEM-004T

Sensor PZEM-004T ini memiliki beberapa fitur atau spesifikasi, fitur atau spesifikasi tersebut, antara lain :

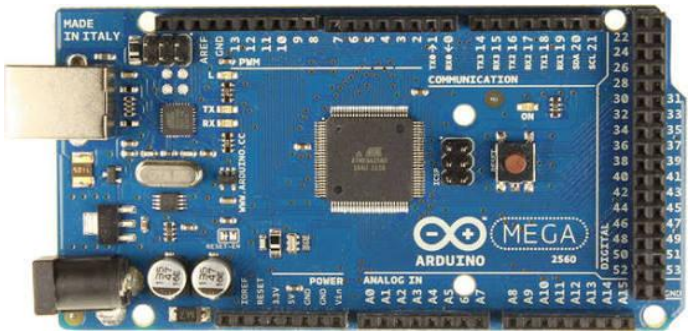
Tabel 2.1 Tabel spesifikasi sensor PZEM-004T [4]

No	Karakteristik	Nilai
1	Tegangan kerja	80 – 260 VAC
2	Pengukuran arus	0 – 100 A
3	Nilai daya	22kW
4	Frekuensi operasi	45 – 65 Hz
5	Akurasi pengukuran	10 grade
6	Antarmuka *serial* UART	UART 9600
7	Tegangan Supply	5V

2.6 *Arduino Mega 2560*

Arduino Mega 2560 adalah papan mikrokontroler berbasis *ATmega2560. Arduino Mega 2560* memiliki 54 *pin digital input/output*, dimana 15 *pin* dapat digunakan sebagai *output PWM, 16 pin* sebagai *input analog*, dan 4 *pin* sebagai *UART (port serial hardware)*, 16 MHz kristal *osilator*, koneksi *USB, jack power, header ICSP*, dan tombol *reset*. Ini semua yang diperlukan untuk mendukung mikrokontroler. Cukup dengan menghubungkannya ke komputer melalui kabel *USB* atau *power* yang dihubungkan dengan *adaptor AC-DC* atau baterai yang

digunakan untuk memulai mengaktifkannya. Bentuk fisik dari *arduino mega 2560* ditunjukkan pada gambar 2.4.



Gambar 2.4 *Arduino Mega 2560* [5]

2.6.1 Spesifikasi *Arduino Mega 2560*

Fitur atau spesifikasi tersebut digunakan sebagai acuan dalam pemilihan Arduino dan I/O yang digunakan dalam proyek akhir ini *Arduino Mega 2560* memiliki beberapa fitur atau spesifikasi yang akan di jelaskan dalam tabel 2.2.

Tabel 2.2 Spesifikasi *Arduino Mega 2560* [5]

No	Karateristik	Nilai
1	Mikrokontroler	ATMEGA 2560
2	Tegangan Operasi	5v
3	Tegangan *Input*	7v-12v
4	Tegangan *Input*	6v-20v
5	Pin Digital *I/O*	54 Pin
6	Input Analog	16 Pin
7	Arus DC per pin *I/O*	20 mA
8	Arus DC untuk pin 3,3 v	50 mA
9	*Memory Flash*	256 KB
10	SRAM	8 KB
11	EEPROM	4 KB
12	*Clock Speed*	16 MHz

2.7 LCD I2C 20x4 *(Liquid Crystal Display)*

LCD atau *Liquid Crystal Display* adalah suatu jenis media tampilan yang menggunakan kristal cair sebagai penampil utama, kristal cair tidak menghasilkan cahaya secara langsung. Sumber cahaya pada *LCD* dihasilkan oleh iluminasi yang disebut sebagai *backlight* yang berada pada sisi belakang *LCD*. Sebagai pembentuk karakter pada *LCD*, sistem ini menggunakan HD44780 yang telah ditanam pada *LCD*. HD44780 dapat mencetak karakter *ASCII* dan 28 simbol khusus lainnya. Antarmuka *LCD* pada sistem ini menggunakan *I2C* dengan PCF8574 sebagai kontrolernya. *I2C* membutuhkan 2 pin sebagai kontrolernya, yaitu *SDA* (*Serial Data Line*) dan *SCL* (*Serial Clock Line*). Bentuk fisik dari *LCD I2C* pada gambar 2.5.

Gambar 2.5 LCD I2C 20x4 [6]

2.8 *Relay*

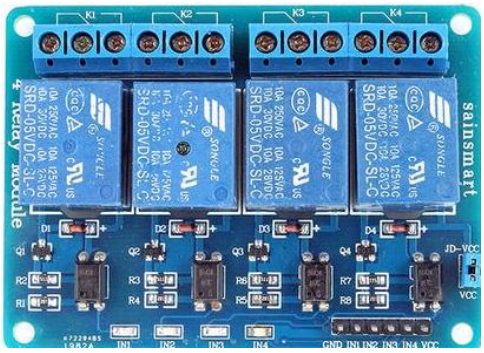
Relay adalah saklar yang dioperasikan secara listrik dan merupakan komponen *Electromechanical* yang terdiri dari 2 bagian utama yakni *electromagnet* dan mekanikal. Modul *relay* menggunakan prinsip kerja berupa elektromagnetik untuk menggerakkan kontak saklar sehingga dengan arus listrik yang kecil relay masih dapat menghantarkan listrik yang bertegangan lebih tinggi. Modul *relay* yang digunakan ini merupakan sebuah perangkat rangkaian komplit *relay* yang telah didesain khusus dilengkapi dengan *pin* untuk menyambungkan koneksi ke *arduino mega 2560*. Bentuk fisik dari *relay* tersebut ditunjukkan pada gambar 2.6.

Pada dasarnya, *relay* ini terdiri dari 4 komponen dasar yaitu :

1. *Elektromagnet (coil)*
2. *Armature*
3. *Switch Contact Point*
4. *Spring*

Kontak pin *relay* sendiri terdiri dari 2 jenis yaitu :

1. *Normally Close* yaitu kondisi awal sebelum diaktifkan akan selalu berada di posisi *CLOSE.*
2. *Normally Open* yaitu kondisi awal sebelum diaktifkan akan selalu berada di posisi *OPEN.*



Gambar 2.6 *Relay* [7]

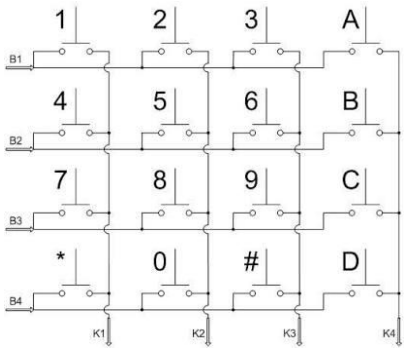
2.9 *Keypad 4x4*

Keypad matrik adalah tombol-tombol yang telah disusun secara matriks sehingga dapat mengurangi penggunaan *pin input*. *Keypad* ini adalah penggabungan dari 16 tombol yang dibentuk menjadi satu serta memiliki lapisan perekat dibagian belakangnya.

2.9.1 Konstruksi *keypad 4x4*

Konstruksi *keypad 4x4*, yaitu terdiri dari 4 baris dan 4 kolom dengan *keypad* berupa saklar *push button* yang diletakkan disetiap persilangan kolom dan barisnya. Rangkaian *keypad* dibawah terdiri dari 16 saklar *push button* dengan

konfigurasi 4 baris dan 4 kolom. 8 *line* yang terdiri dari 4 baris dan 4 kolom tersebut dihubungkan dengan *pin arduino Mega 2560*. Sisi baris dari *keypad* ditandai dengan nama Row1, Row2, Row3, dan Row4. Sedangkan sisi kolom dari *keypad* ditandai dengan nama Col1, Col2, Col3, Col4. Sisi *input* atau *output* dari *keypad 4x4* inti tidak mengikat, dapat dikonfigurasikan kolom sebagai *input* dan baris sebagai *output* atau sebaliknya tergantung dari *programmer*-nya. Gambar konstruksi *keypad 4x4*, ditunjukkan pada gambar 2.7.



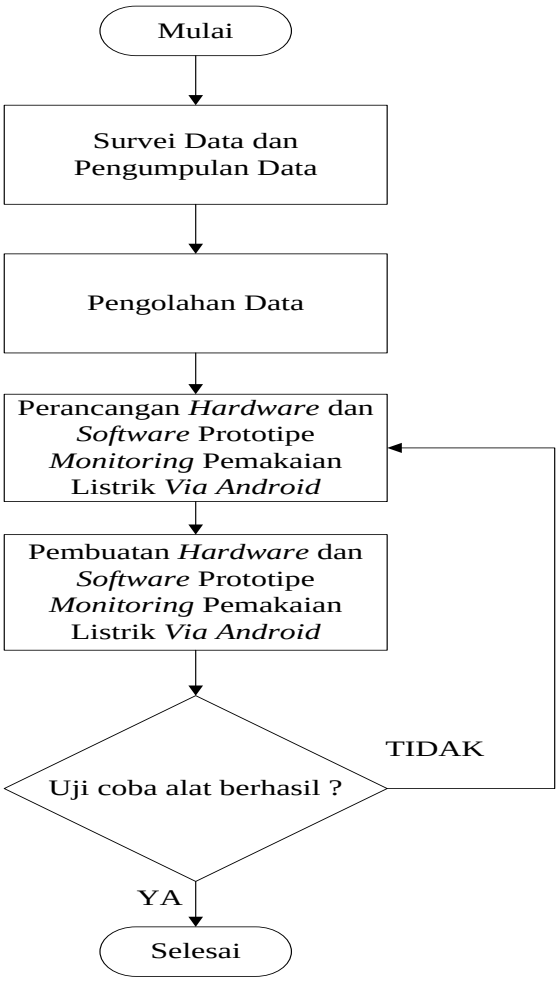
Gambar 2.7 Konstruksi *keypad 4x4* [5]

2.10 *Buzzer*

Buzzer adalah sebuah komponen elektronika yang berfungsi untuk mengubah getaran listrik menjadi getaran suara. Jenis *buzzer* yang sering ditemukan dan digunakan bisanya adalah *buzzer piezoelectric* hal ini disebabkan oleh berbagai kelebihan yang dimiliki *buzzer* contohnya saja karna b*uzzer* jenis ini mudah didapatkan, lebih ringan, harga yang murah serta memiliki keunggulan yang lebih mudah digabungkan ke dalam rangkaian elektronika lainnya. *Buzzer* terdiri dari kumparan yang terpasang pada diafragma dan kemudian kumparan tersebut dialiri arus sehingga menjadi elektromagnet, kumparan yang tadi akan tertarik ke dalam atau keluar, tergantung dari arah arus dan polaritas magnetnya, karena kumparan dipasang pada diafragma maka setiap gerakan kumparan akan menggerakkan diafragma secara bolak-balik sehigga membuat udara bergetar yang akan menghasilkan suara. *Buzzer* biasa digunakan sebagai indikator bahwa proses telah selesai atau terjadi suatu kesalahan pada sebuah alat (*alarm*). [7]

BAB III
METODE PELAKSANAAN

Dalam pelaksanaan proyek akhir ini, dilakukan tahapan-tahapan penelitian yang bertujuan untuk memudahkan penulis dalam proses pembuatan proyek akhir. Adapun tahap-tahap penelitian yang dilakukan penulis adalah seperti yang terlihat pada *flowchart* Gambar 3.1.



Gambar 3.1 *Flowchart* metode pelaksanaan

3.1 Survei Data dan Pengumpulan Data

Pengumpulan data merupakan proses pencarian data referensi-referensi yang memiliki kaitan dengan data yang dibutuhkan dalam proses pengerjaan proyek akhir maupun penyusunan makalah proyek akhir ini baik melalui *internet* maupun dari buku-buku. Dalam proses pencarian referensi dilakukan dengan mencari data yang berkaitan dengan jurnal *kWh* meter, penginstalasian beban, titik peletakan sensor, mengkoneksikan *bluetooth hc-06*. Pada tahap proses pencarian referensi ini akan dipelajari beberapa hal-hal yang berhubungan dengan proyek akhir ini diantaranya:

1. Pencarian data referensi secara primer, yaitu melakukan diskusi dengan pembimbing dan instruktur Politeknik Manufaktur Negeri Bangka Belitung yang lain untuk memperjelas informasi yang ada dan untuk mendapatkan konsep yang akan menjadi pilihan untuk digunakan.
2. Pencarian data referensi secara sekunder, yaitu sebagai berikut:
    a. Dari referensi-referensi buku

       Mencari referensi-referensi dari buku dilakukan untuk mendapatkan rumus-rumus yang berkaitan dengan proyek akhir ini.
    b. Dari internet

       Mencari referensi dari internet dilakukan untuk mendapatkan contoh model atau bentuk dari pemrograman yang akan digunakan untuk proyek akhir ini. Namun untuk mencari referensi tersebut dapat dilakukan dengan memilih dari sumber yang terpercaya.

3.2 Pengolahan Data

Setelah selesai proses pencarian data referensi kemudian dilakukan proses pengolahan data. Pengolahan data adalah tahap untuk mengolah data yang sudah dikumpulkan berupa desain, sistem control dan komponen-komponen penunjang lainnya yang akan digunakan pada pembuatan proyek akhir ini. Kemudian didiskusikan kembali bersama pembimbing untuk memperjelas informasi yang ada dan dari hasil diskusi dengan pembimbing dapat diambil kesimpulan menganai desain sistem, teknik pengerjaan, maupun metode yang digunakan untuk menyelesaikan proyek akhir ini. Setelah melakukan pengolahan data dan survey data yang dilakukan maka dilakukan lah perancangan hardware dan software pada alat yang akan dibuat baik itu perancangan power supply, sensor, relay dan komponen-komponen lain.

3.3 Perancangan *Hardware* dan *Software* Prototipe *Monitoring* Pemakaian Listrik *Via Android*

Perancangan *hardware* pada alat ini bertujuan untuk membuat suatu rancangan yang dapat dibuat dengan waktu yang efisien, adapun rancangan *hardware* terdiri dari konstruksi yang sesuai, letak komponen, dan fungsi komponen. Berikut perancangan *hardware* yang dibuat, antara lain perancangan rangkaian skematik *Power Supply*, perancangan rangkaian skematik sensor PZEM-004T, perancangan rangkaian skematik *Relay 5v*, perancangan rangkaian skematik *Keypad 4x4*, perancangan rangkaian skematik *LCD I2C 20x4*, perancangan rangkaian skematik *buzzer* dan led indicator, perancagan rangkaian skematik *Bluetooth HC-06*.

3.4 Pembuatan *Hardware* dan *Software* Prototipe *Monitoring* Pemakaian Listrik *Via Android*

Pembuatan alat ini meliputi pembuatan *hardware* dan *software* yang digunakan. Pembuatan *hardware* dilakukan stelah perancangan *hardware.* Sedangkan pembuatan *software* ini dilakukan setelah perancangan *hardware* selesai dibuat. Rancangan *software* yang telah dibuat kemudian digabungkan dengan konstruksi lainnya berdasarkan rancangan konstruksi yang diinginkan. Dan untuk pembuatan *software* proyek akhir ini penulis menggunakan *Software Integrated Development Environment (IDE).*

3.5 Uji Coba *Hardware* dan *Software*

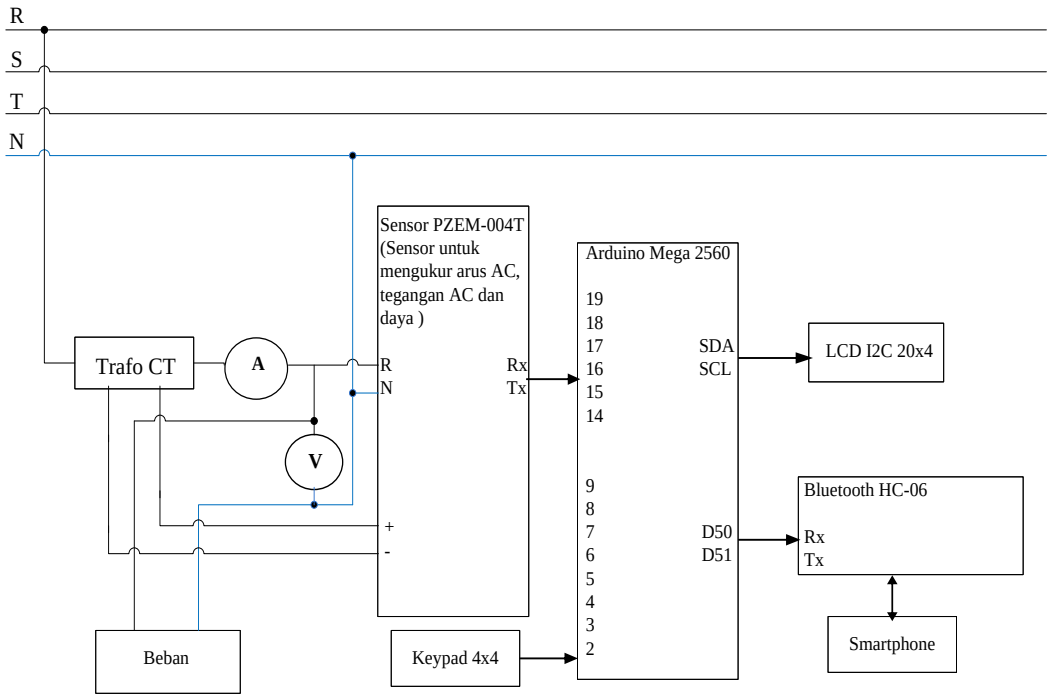
Dalam suatu uji coba alat biasanya mengalami *trial and error.* Untuk itu *hardware* dan *software* yang sudah dibuat harus diuji coba agar proses kerja yang diinginkan tercapai. Apabila dalam proses uji coba mengalami gangguan (*error*) dan tidak bekerja sesuai yang diinginkan, maka proses selanjutnya adalah perbaikan pada sistem yang mengalami gangguan tersebut. Setelah dilakukan uji coba kembali, jika berhasil dan sesuai dengan proses kerja yang diinginkan, maka pembuatan selesai.

BAB IV
PEMBAHASAN

Dalam bab ini berisi pembahasan tentang proses yang dilakukan dalam pembuatan alat, yang meliputi proses perancangan alat, proses pembuatan alat dan proses pengujian serta perbaikan alat.

4.1 Proses Perancangan Sistem Kontrol Prototipe *Monitoring* Pemakaian Listrik *Via Android*

Perancangan sistem kontrol sangat menentukan hasil yang akan didapatkan. Blok diagram sistem kontrol yang digunakan pada alat prototipe *monitoring* pemakaian listrik *via android* ditunjukkan pada gambar 4.1.



Gambar 4.1 Blok diagram sistem kontrol prototipe *monitoring* pemakaian listrik *via android*

Pada blok diagram sistem kontrol prototipe *monitoring* pemakaian listrik *via android* terdapat trafo ct dan beban yang digunakan pada alat ini. Berikut ini adalah keterangan untuk trafo ct pada sensor PZEM-004T dan beban yang digunakan :

a. Trafo CT

   Pengukuran arus : 0 – 100 A

   Akurasi pengukuran : 10 grade

b. Beban

   Beban yang digunakan : 1. Setrika listrik 300W

   2. Kipas angin 35 W

   3. Cas Laptop 18 W

Berdasarkan blok diagram di atas, berikut ini adalah tabel 4.1. yang menjelaskan tentang *pin-pin* pada *Arduino Mega 2560* yang dihubungkan ke masing-masing *input* dan *output*.

Tabel 4.1 *Pin* yang digunakan pada *Arduino Mega 2560*

Komponen	*Pin* pada *arduino mega 2560*
PZEM-004T	RX1 dan TX1, RX2 dan TX2, RX3 dan TX3
Relay 4 channel	10,11,12
Keypad 4x4	9,8,7,6,5,4,3,2
LCD I2C 20x4	SDA,SCL
Bluetooth HC-06	RX1(19),TX1(18)

Berdasarkan gambar 4.1 blok diagram sistem kontrol prototipe *monitoring* pemakaian listrik *via android* diatas, berikut penjelasan komponen-komponen yang digunakan dalam proyek ini :

1. Modul sensor PZEM-004T

   Komponen ini merupakan komponen yang berfungsi sebagai sensor pengukuran dari arus beban, tegangan beban, daya beban dan cos phi. Modul sensor PZEM-004T ini dibundel dengan kumparan trafo arus yang memiliki diameter 33mm. Bagian utama dari modul sensor PZEM-004T ini adalah chip SD3004 serta SDIC Microelectronics Co., Ltd. Modul sensor PZEM-004T ini juga dilengkapi dengan antarmuka komunikasi data serial TTL yang melalui *port serial* dapat dibaca serta bisa mengatur parameter yang relevan.

2. *Relay*

   Komponen ini merupakan komponen kontrol yang terdapat pada alat ini. Karena komponen ini berfungsi sebagai saklar untuk beban. Jadi cara kerja *relay* ini pada alat adalah sebagai *on* atau *off* pada beban dan instruksi kontrolnya dikontrol melalui *android* yang telah terdapat aplikasi *blynk*. Pengontrolan ini menggunakan fitur – fitur yang terdapat pada aplikasi *Blynk*. kontak yang digunakan untuk pengontrolan dalam alat ini adalah kontak *NC (normally close)* dari *relay*. Sedangkan *pin input relay* dikoneksikan ke *Arduino Mega 2560*.

3. *Arduino Mega 2560*

   Arduino mega 2560 ini adalah sebagai mikrokontroller yang digunakan dalam proses kontrol alat ini dan menggunakan *software Arduino IDE* sebagai pendukung untuk menulis program yang akan diperintahkan dan dijalankan untuk alat ini. *Arduino Mega 2560* memiliki 54 *pin digital input/output*, dimana 15 *pin* dapat digunakan sebagai *output PWM*, *16 pin* sebagai *input analog*, dan 4 *pin* sebagai *UART (port serial hardware)*, 16 MHz kristal *osilator*, koneksi *USB, jack power, header ICSP*, dan tombol *reset*.

4. LCD I2C 20x4

   LCD I2C Ini adalah *display* atau sebagai penampil hasil yang dapatkan dalam alat ini. Yang ditampilkan pada LCD adalah pulsa *kWh*, arus tiap beban dan tegangan yang terdapat pada beban dari alat ini. Dan juga berfungsi sebagai

penampil uang *token* untuk pulsa *kWh* meter prabayar. Terdapat 4 *pin* yang ada pada LCD ini, diantaranya adalah *pin VCC, GND, SDA* dan *SCL*. *Pin SDA* dan *SCL* inilah yang digunakan sebagai komunikasi antara *Arduino* dan LCD.

5. *Keypad 4x4*

   Sebuah piranti yang berfungsi sebagai masukan yang berupa tombol yang harus ditekan atau lebih dikenal dengan *pushbutton*. Terdapat 8 pin yang harus dihubungkan ke *Arduino*. Dan *keypad* ini, digunakan untuk memasukkan nilai *token* yang berupa uang dalam pembelian pulsa *kWh* meter prabayar. *Keypad* ini adalah penggabungan dari 16 tombol yang dibentuk menjadi satu serta memiliki lapisan perekat dibagian belakangnya. *Keypad 4x4* biasanya digunakan untuk aplikasi pengunci atau aplikasi lainnya yang membutuhkan *input keypad.*

6. Laptop

   Digunakan untuk menampilkan hasil *data base* alat monitoring suhu & detak jantung berbasis arduino.

7. Led Indikator

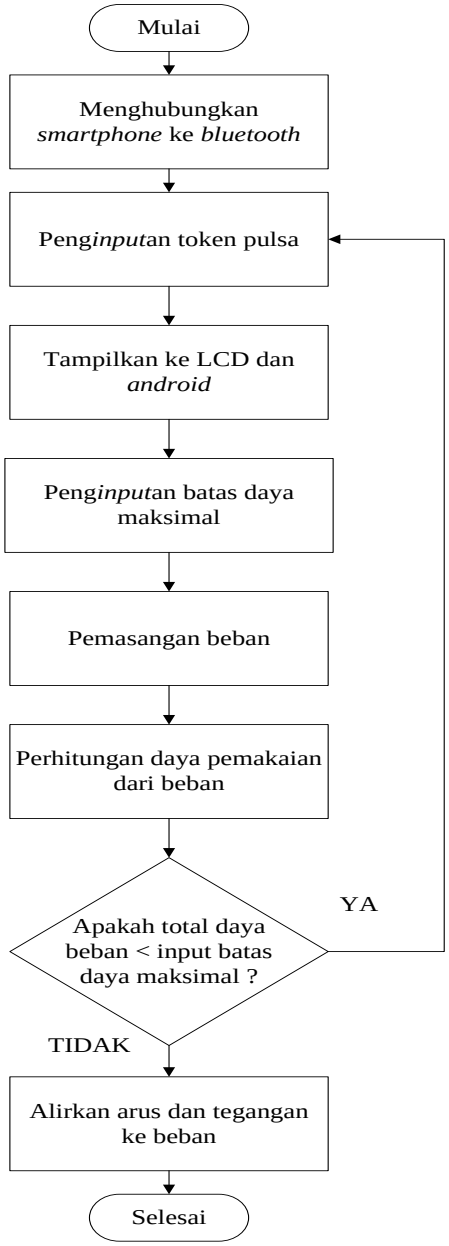
   Digunakan untuk menunjukkan ke pengguna apabila led hijau menyala maka pulsa *kWh* telah terisi. Jika led merah menyala maka pulsa *kWh* sudah mencapai batas limit pulsa atau pulsa habis.

8. *Bluetooth HC-06*

   Digunakan untuk komunikasi antara *arduino mega 2560* dengan *smartphone* dan juga digunakan sebagai media pengiriman data hasil pengukuran sensor PZEM-004T terhadap beban yang digunakan.

Flowchart dari sistem kerja dari alat ini menjelaskan tentang cara kerja dari alat. Proses dari flowchart ini terdiri dari beberapa langkah untuk menyelesaikan alat prototipe *monitor*ing pemakaian listrik *via android*, antara lain menghubungkan koneksi *bluetooth HC-06* dengan *bluetooth* smartphone, pengisian token pulsa menggunakan *keypad*, peng*input*an daya maksial beban menggunakan *keypad*,

pemasangan beban, perhitungan daya listrik pada saat beban dipasang, dan menampilkan hasil pengukuran ke *android*. *Flowchart* dari sistem kerja dari alat prototipe *monitor*ing pemakaian listrik *via android* ditunjukkan pada gambar 4.2.



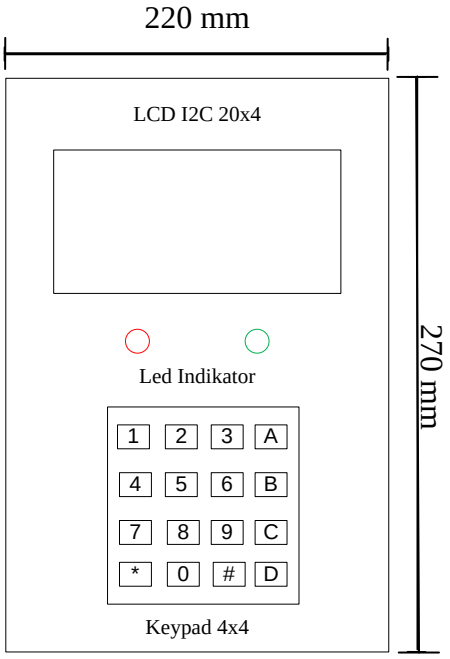
Gambar 4.2 *Flowchart* sistem kerja alat prototipe *monitor*ing pemakaian listrik *via android*

4. 2 Proses Pembuatan *Hardware*

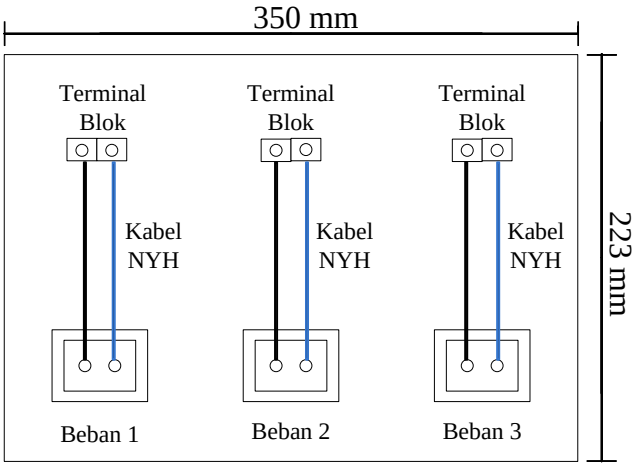
Perancangan dan Pembuatan *hardware* terdiri dari 2 bagian yaitu, perancangan dan pembuatan *hardware* secara mekanik serta perancangan dan pembuatan *hardware* secara elektrik. Adapun untuk tahap-tahap perancangan dan pembuatan *hardware* akan dibahas sebagai berikut:

4.2.1 Perancangan *Hardware* Secara Mekanikal

Dalam proses ini menggunakan bahan triplek dan *box kWh*. Triplek digunakan berfungsi sebagai landasan dari komponen-komponen yang akan di rangkaian menjadi sebuah kontrol dan sebagai landasan untuk tempat terminal beban. Sedangkan *box kWh*, digunakan sebagai penutup komponen-komponen rangkaian kontrol dan juga sebagai media untuk meletakkan LCD I2C serta *keypad 4x4*. Berikut adalah tampak atas konstruksi *kWh* meter prabayar yang ditunjukkan pada gambar 4.3 dan tampak atas konstruksi terminal beban yang ditunjukkan pada gambar 4.4.



Gambar 4.3 Tampak atas konstruksi *kWh* meter prabayar



Gambar 4.4 Tampak atas konstruksi terminal beban

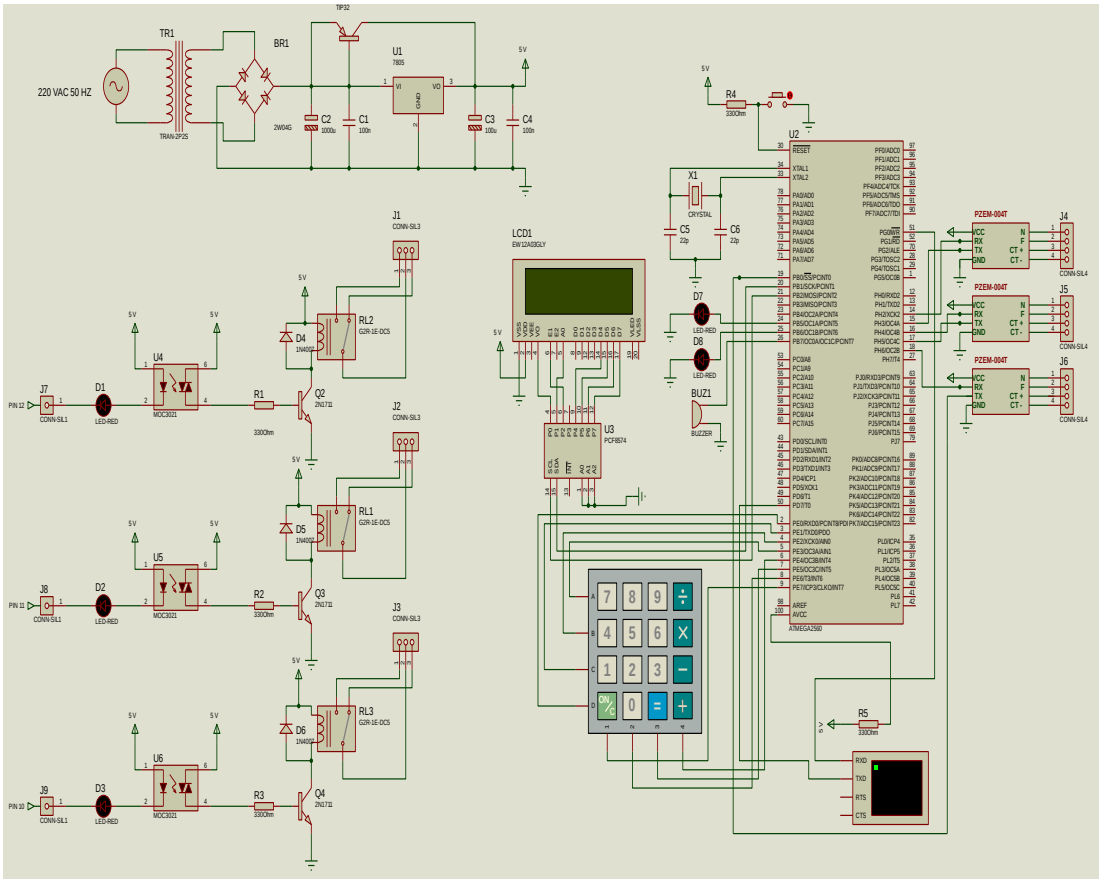
4.2.2 Pembuatan *Hardware* Secara Mekanikal

Alat ini akan berbentuk *box kWh* dengan landasan yang menggunakan triplek dengan memiliki tebal 5 mm. Agar terlihat lebih mudah untuk dibawa dan digunakan serta mempermudah dalam peletakkannya. Alat ini dibentuk dengan ukuran 22 cm x 27 cm. Ini adalah ukuran triplek untuk landasan komponen kontrol. Sedangkan ukuran triplek terminal beban adalah 35 cm x 22,3 cm. Dengan spesifikasi ukuran tersebut, tentu akan mempermudah dalam pembawaan alat tersebut dan ukuran telah disesuaikan dengan ukuran *box kWh*. Hasil pembuatan konstruksi dari prototipe *monitoring* pemakaian listrik *via android*, ditunjukkan pada gambar 4.5.

Gambar 4.5 Hasil pembuatan konstruksi perangkat keras *kWh* meter prabayar

4.2.3 Proses Perancangan dan Pembuatan Secara Elektrikal

Berikut adalah skematik lengkap perancangan *hardware* secara elektrik alat prototipe *monitoring* pemakaian listrik *via android* yang ditunjukkan pada gambar 4.6



Gambar 4.6 Skematik prototipe *monitoring* pemakaian listrik *via android*

Proses perancangan dan pembuatan *hardware* secara elektrik terdiri dari beberapa bagian yaitu:

a. Perancangan dan pembuatan *power supply*
b. Perancangan dan pembuatan sensor PZEM-004T

c. Perancangan dan pembuatan *relay*

d. Perancangan dan pembuatan keypad

e. Perancangan dan pembuatan LCD

f. Perancangan dan pembuatan *Bluetooth HC-06*

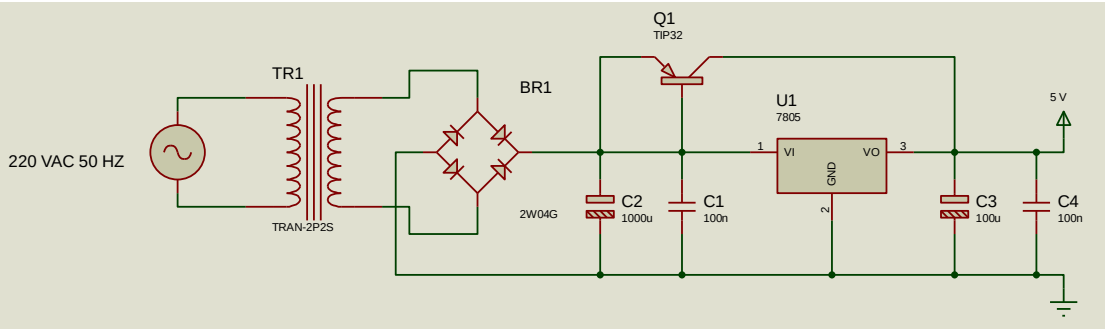
g. Perancangan dan pembuatan led indikator

h. Perancangan dan pembuatan *buzzer*

Menurut skematik *hardware* yang telah dibuat seperti gambar 4.6 diatas, berikut ini adalah penjelasan mengenai bagian-bagiannya yang dari komponen yang ada pada gambar 4.6 :

4.2.3.1 Perancangan *Power Supply* 5v 3A

Power supply atau yang dikenal juga sebagai rangkaian catu daya yang memiliki fungsi sebagai pengubah tegangan pada AC menjadi tegangan DC. Pada prototipe *monitoring* pemakaian listrik *via android* ini menggunakan catu daya 5V dengan memiliki nilai arus 3A sebagai pensuplai tegangan yang akan terhubung ke *arduino*, sensor arus, sensor tegangan, *relay* dan LCD. Penggunaan catu daya 5V 3A ini dikarenakan arduino, sensor serta relay dan alat penunjang lainnya hanya membutuhkan catu daya yang memiliki tegangan 5V. Rangkaian *hardware power supply*, ditunjukkan pada gambar 4.7.



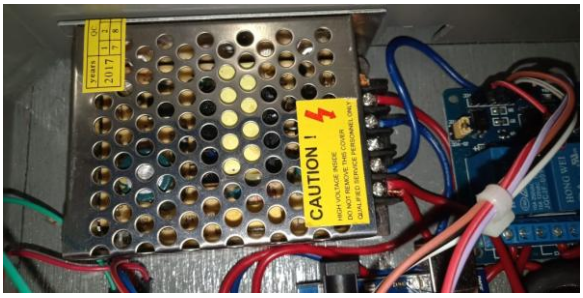
Gambar 4.7 Rangkaian *skematik power supply* 5v 3A

Berikut ini penjelasan prinsip kerja rangkaian catu daya ini adalah :

Tegangan 220 VAC 1A dialirkan ke trafo step down yang akan merubah tegangan menjadi 12 VAC 1A. Tegangan 12 VAC 1A disearahkan menggunakan dioda gelombang penuh dengan arus yang memiliki nilai sebesar 3A. Keluaran dari dioda merupakan gelombang penuh kemudian melewati kapasitor untuk di*filter*, proses ini bertujuan untuk menghilangkan *ripple* agar mendapatkan hasil tegangan DC murni. Kemudian tegangan 12 VDC 3A masuk ke dalam IC LM7805 dan akan diubah tegangannya menjadi tegangan yang bernilai 5VDC 3A. Kapasitor 100 uf disini berfungsi untuk memperhalus *ripple* yang sudah di regulasi oleh IC LM7805. Sedangkan transistor TIP32 akan berfungsi untuk memperkuat arus yang mengalir dari dioda.

4.2.3.2 Pembuatan *Power Supply* 5v 3A

Pada proyek akhir ini catu daya yang digunakan adalah *power supply switching* dengan tegangan masukan sebesar 200-240VAC dan arus sebesar 2,5A. Sedangkan untuk tegangan keluaran dari *power supply switching* ini sebesar +5VDC dengan arus 3A. Pembuatan *power supply switching* diputuskan dengan membeli modul *power supply switching* yang telah jadi dan mudah didapat dipasaran. Pemasangan *power supply switching* pada prototipe *monitoring* pemakaian listrik *via android* dapat dilihat gambar 4.8 menunjukan gambar *power supply switching* yang digunakan dalam proyek akhir.



Gambar 4.8 Pemasangan *power supply switching* 5v 3A

4.2.3.3 Pengujian *Power Supply Switching* 5v 3A

Pada saat uji coba rangkaian catu daya, dilakukan pengujian apakah rangkaian catu daya yang digunakan sesuai dengan yang diinginkan. Berikut adalah gambar dari pengujian *power supply switching* 5V 3A, ditunjukkan pada gambar 4.8.

Gambar 4.9 Hasil pengujian *Power Supply* 5v 3A

Setelah dilakukan proses pengukuran pada *Power Supply* 5VDC 3A dengan menggunakan alat ukur multitester dibuatlah tabel hasil pengujian yang dilakukan dalam pengukuran *Power Supply* 5VDC 3A yang ditunjukkan pada tabel 4.2.

Tabel 4.2. Hasil Pengujian *Power Supply*

Item	Output Tegangan Nilai Sebenarnya	*Power Supply*(V) Nilai Terbaca	*Error* (%)
Power Supply 5v 3A	5 v	5,15 v	0,1

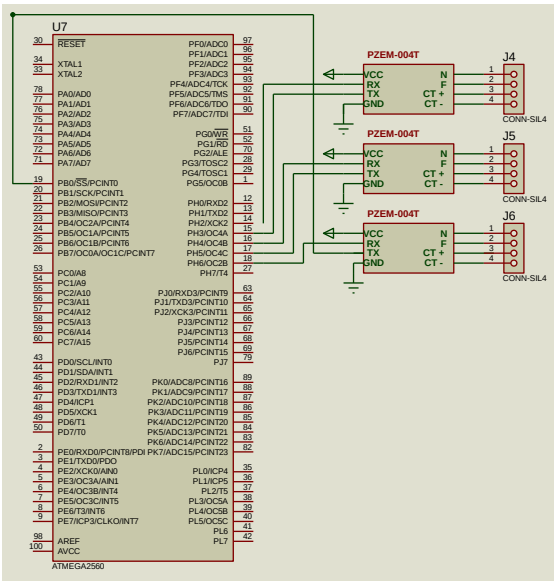
Dari data tabel diatas, penulis dapat menganalisa bahwa power supply switching 5v 3A layak digunakan untuk proyek akhir ini. Dikarenakan tegangan yang sebenarnya pada power supply switching 5v 3A dengan hasil pengukuran yang dilakukan memiliki persentase error sebesar 0,1 %.

4.2.3.4 Perancangan dan Pembuatan Sensor Arus, Sensor Tegangan dan Sensor Cos φ

Dalam proses ini terdapat dua proses yang dilakukan yaitu adalah proses perancangan sensor PZEM-004T dan proses pembuatan sensor PZEM-004T. Dalam proses perancangan sensor PZEM-004T, penulis melakukan proses merancang skematik rangkaian dan *pin-pin* yang digunakan sensor PZEM-004T.

4.2.3.4.1 Perancangan Sensor Arus, Sensor Tegangan dan Sensor Cos φ

Dalam merancang sensor ini penulis menggunakan sensor PZEM-004T yang berfungsi multiguna dapat mengukur arus, tegangan, daya dan Cos φ. Berikut ini adalah gambar skematik sensor PZEM-004T, ditunjukkan pada gambar 4.10.



Gambar 4.10 Skematik sensor PZEM-004T

Dalam proses perancangan sensor ini, penulis merancang skematik dari sensor PZEM-004T. Skematik dari sensor PZEM-004T ini dibuat menggunakan software atau aplikasi elektronika yaitu proteus. Setelah membuat skematik di aplikasi kemudian penulis akan menjelaskan *pin* yang digunakan dalam merancang

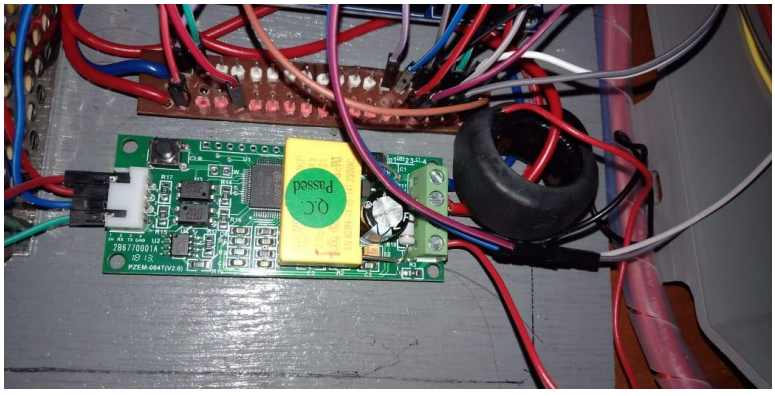
skematik antara sensor PZEM-004T dengan Arduino Mega 2560. Berikut adalah tabel penjelasan *pin* yang terhubung dari Sensor PZEM-004T ke *Arduino Mega 2560*:

Tabel 4.3 Tabel koneksi sensor PZEM-004T dengan *Arduino Mega 2560*

No	Pin Sensor PZEM-004T	Pin *Arduino Mega 2560*
1	VCC	Pin 5v
2	Rx1, Tx1	Pin 19, Pin 18
3	Rx2, Tx2	Pin 17, Pin 16
4	Rx3, Tx3	Pin 15, Pin 14
5	GND	Pin GND

4.2.3.4.2 Pembuatan Sensor Sensor Arus, Sensor Tegangan dan Sensor Cos φ

Pembuatan sensor PZEM-004T diputuskan dengan membeli modul yang telah jadi dan yang mudah didapatkan di pasaran. Hal ini dikarenakan modul yang dijual secara umum dapat langsung digunakan tanpa harus membuat modul sensor tersebut, hal ini tentu dapat meminimalisir waktu dalam proses pengerjaan proyek akhir. Dan juga meminimalisir error atau gagal dalam pembuatan sensor yang sesungguhnya. Gambar 4.11 menunjukan sensor PZEM-004T yang sudah terpasang pada hardware prototipe *monitoring* pemakaian listrik *via android*.



Gambar 4.11 Pemasangan sensor PZEM-004T

4.2.3.4.3 Pengujian Sensor Arus

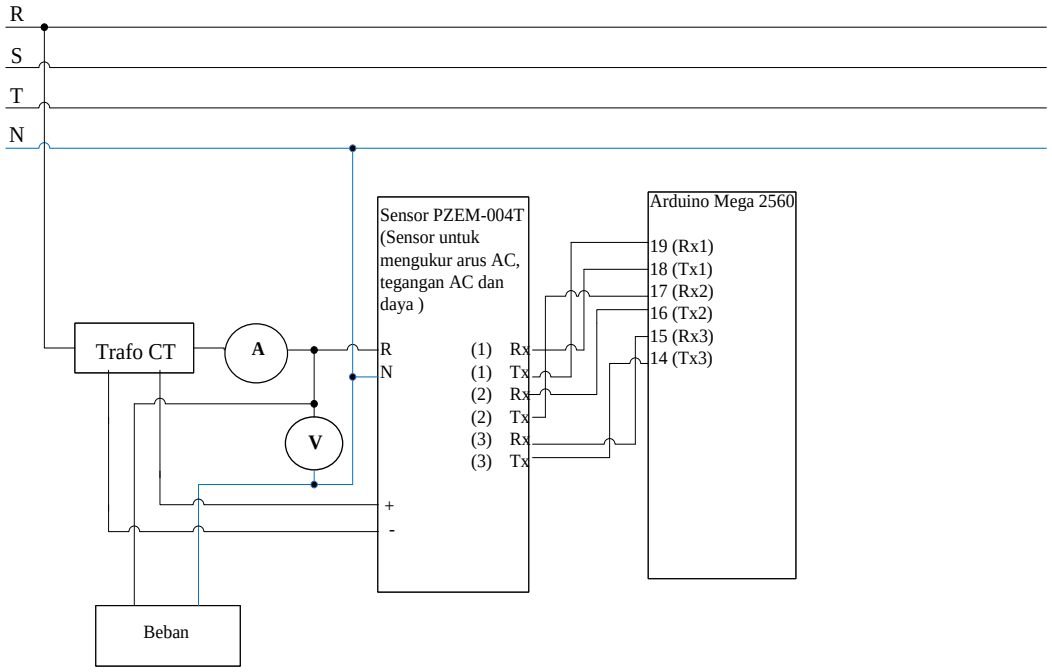
Setelah melakukan proses perancangan dan pembuatan dari sensor PZEM-004T yaitu proses merancang skematik. Maka dilakukanlah proses instalasi atau pengkoneksian antara *Arduino mega 2560* dan sensor PZEM-004T. Sebelum menghubungkan sensor PZEM-004T kepada *Arduino Mega 2560,* sensor terlebih dahulu diuji. Pengujian dilakukan dengan menampilkan nilai hasil pengukuran yang dilakukan sensor PZEM-004T yang berupa nilai arus pada beban dan nilai tegangan pada beban yang ditampilkan di LCD I2C 20x4 ditunjukkan pada gambar 4.13, gambar 4.14, gambar 4.15. Sedangkan nilai tegangan pada setiap terminal yang ditampilkan di LCD I2C 20x4 ditunjukkan gambar 4.17, gambar 4.18, gambar 4.19. Berikut ini program sensor PZEM-004T untuk pengujian :

```
#include <SoftwareSerial.h>
#include <PZEM004T.h>
PZEM004T pzem1(&Serial1);
PZEM004T pzem2(&Serial2);
PZEM004T pzem3(&Serial3);
IPAddress ip(192,168,1,1);
float sensorteg = 0;
float sensorarus1 = 0;
float sensorarus2 = 0;
float sensorarus3 = 0;
float sensordaya1 = 0;
float sensordaya2 = 0;
float sensordaya3 = 0;
float cospi1 =1;
float cospi2 =1;
float cospi3 =1;
void setup (){
```

```
        pzem1.setAddress(ip);
        pzem2.setAddress(ip);
        pzem3.setAddress(ip);}
    void tampilarusteg()
    { //input_keypad();
      dtostrf(sensorarus1,2,2,lcdbuff2); sprintf(lcdbuff,"%sA ",lcdbuff2);
      lcd.setCursor (1,2); lcd.print (lcdbuff);
      dtostrf(sensorarus2,2,2,lcdbuff2); sprintf(lcdbuff,"%sA ",lcdbuff2);
      lcd.setCursor (7,2); lcd.print (lcdbuff);
      dtostrf(sensorarus3,2,2,lcdbuff2);   sprintf(lcdbuff,"%sA ",lcdbuff2);
      lcd.setCursor (13,2); lcd.print (lcdbuff); dtostrf(sensorteg,2,2,lcdbuff2);
      sprintf(lcdbuff,"%sVAC ",lcdbuff2);
      lcd.setCursor (1,1); lcd.print (lcdbuff);} Void hitungkWh (){
      sensordaya1= pzem2.power(ip); sensordaya2= pzem3.power(ip);
      sensordaya3= pzem1.power(ip);   sensorteg = pzem2.voltage(ip);
      sensorarus1 = pzem2.current(ip);
      if (sensorarus1<0){sensorarus1=0;cospi1=1;}
      sensorarus2 = pzem3.current(ip);
      if (sensorarus2<0){sensorarus2=0;cospi2=1;}
      sensorarus3 = pzem1.current(ip);
      if (sensorarus3<0){sensorarus3=0;cospi3=1;}
      if (sensorarus1>0){cospi1=sensordaya1/(sensorteg*sensorarus1);}
      if (sensorarus2>0){cospi2=sensordaya2/(sensorteg*sensorarus2);}
      if (sensorarus3>0){cospi3=sensordaya3/(sensorteg*sensorarus3);}  }
     Void loop (){ tampilkwh();tampilarusteg();}
```

Gambar 4.12 rangkaian elektrikal pengujian yang dilakukan untuk mendapatkan hasil pengukuran sensor PZEM-004T terhadap arus, tegangan dan daya beban yang digunakan pada alat prototipe *monitoring* pemakaian listrik *via android*.

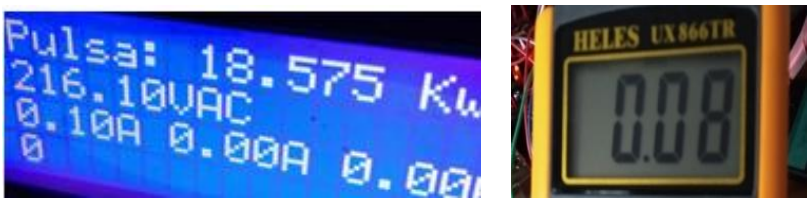
Pada gambar ini terdapat skematik dari peletakan alat ukur yang digunakan seperti peletakan alat ukur amperemeter yang digunakan untuk mengukur arus beban. Sedangkan voltmeter yang digunakan untuk mengukur tegangan beban.



Gambar 4.12 Rangkaian elektrikal pengujian sensor PZEM-004T terhadap arus

Hasil pengujian dari modul sensor PZEM-004T terhadap arus dari beban yang ditampilkan pada *LCD* dan diukur dengan alat ukur multimeter. Hasil ukur ini merupakan acuan yang akan digunakan dalam perhitungan pemakaian listrik terhadap pulsa *kWh* dan sebagai acuan untuk perhitungan persentase error. Ditunjukkan pada gambar 4.13, gambar 4.14, gambar 4.15.

Gambar 4.13 Pengujian sensor arus terhadap beban kipas angin



Gambar 4.14 Pengujian sensor arus terhadap beban cas laptop

Gambar 4.15 Pengujian sensor arus terhadap beban setrika

Untuk beban yang diukur adalah kipas angin dengan daya 55 W, cas laptop dengan daya 18 W, dan setrika dengan daya 300 W. Pengukuran ini dilakukan untuk mendapatkan data dari modul sensor PZEM-004T dan alat ukur multitester sebagai perbandingan atau acuan untuk mendapatkan nilai hasil data yang akurat serta mengurangi nilai persentase error pada perhitungan yang digunakan pada alat ini. Setelah dilakukan pengukuran pada tiap- tiap beban,maka data yang telah didapatkan dari hasil pengukuran yang telah dilaksanakan akan dicantumkan didalam tabel yang ditunjukkan pada tabel 4.4.

Tabel 4.4 Data hasil pengujian sensor PZEM-004T terhadap arus

Daya beban dan daya tampil *di LCD*		Hasil Ukur	Hasil Ukur	Persentase *error* %
Nama beban	*Android*	Sensor	Multimeter	
Kipas angin 42 watt	41 W	0.22 A	0.23 A	0
Cas laptop 18 watt	15 W	0.10 A	0.08 A	0.2
Setrika 300 watt	334 W	1.55 A	1.54 A	0.006

Dari data diatas didapatkan hasil pengukuran modul sensor PZEM-004T, baik itu hasil pengukuran dari modul sensor PZEM-004T ataupun dari alat ukur yang berupa multimeter. Untuk menentukan persentase *error* dari modul sensor PZEM-004T ini dapat dilakukan dengan cara menggunakan perhitungan yang memasukan angka ke dalam rumus sebagai berikut :

$$\textit{error}\text{ sensor PZEM-004T} = \left|\frac{nilai\ \ ukur\ \ sensor - nilai\ \ ukur\ \ multimeter}{nilai\ \ ukur\ \ sensor}\right| X100 \ldots(6)$$

Dibawah ini merupakan perhitungan persentase *error* modul Sensor PZEM-004T terhadap arus beban. Hasil dari perhitungan error ini akan digunakan sebagai perhitungan pemakaian listrik dan perhitungan pemakaian pulsa kWh pada alat ini:

- Hasil ukur multimeter pada beban kipas angin

  $\textit{error}$ sensor PZEM-004T = $\left|\frac{0.22\ A - 0.23A}{0.23\ A}\right| X100\% = 0\ \%$

- Hasil ukur multimeter pada beban cas laptop

  $\textit{error}$ sensor PZEM-004T = $\left|\frac{0.10A - 0.08A}{0.10A}\right| X100\% = \ 0.2\%$

- Hasil ukur multimeter pada beban setrika

  $\textit{error}$ sensor PZEM-004T = $\left|\frac{1.55A - 1.54A}{1.55A}\right| X100\% = \ 0.006\ \%$

Dari data tabel pengujian dan dari hasil perhitungan persentase error yang dilakukan penulis dapat menganalisadan menyimpulkan bahwa saat menggunakan modul sensor PZEM-004T tidak perlu dilakukan pengkalibrasian dalam penggunaan modul sensor PZEM-004T ini, serta dapat disimpulkan dari data diatas bahwa modul sensor PZEM-004T layak digunakan untuk proyek akhir ini karena persentase error yang didapatkan dari modul sensor PZEM-004T mencapai 0%.

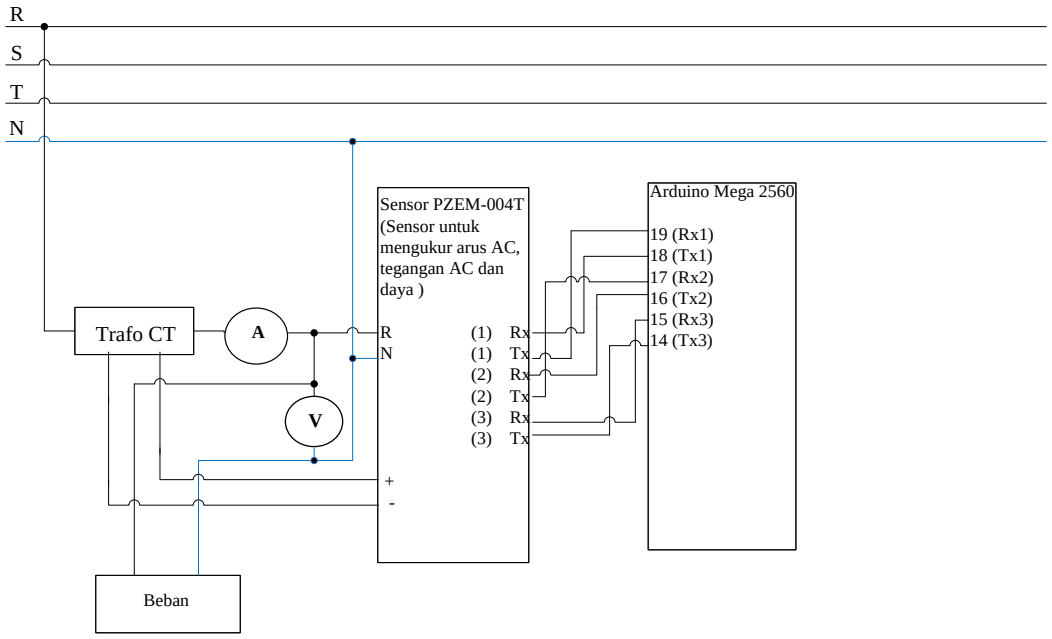
4.2.3.4.4 Pengujian Sensor Tegangan

Setelah melakukan proses perancangan dan pembuatan dari sensor PZEM-004T yaitu proses merancang skematik. Maka dilakukanlah proses instalasi atau pengkoneksian antara *Arduino mega 2560* dan sensor PZEM-004T. Sebelum menghubungkan sensor PZEM-004T kepada *Arduino Mega 2560,* sensor terlebih dahulu diuji. Pengujian dilakukan dengan menampilkan nilai hasil pengukuran yang dilakukan sensor PZEM-004T yang berupa nilai arus pada beban dan nilai tegangan pada beban yang ditampilkan di LCD I2C 20x4 ditunjukkan pada gambar 4.13, gambar 4.14, gambar 4.15. Sedangkan nilai tegangan pada setiap terminal yang ditampilkan di LCD I2C 20x4 ditunjukkan gambar 4.17, gambar 4.18, gambar 4.19. Berikut ini program sensor PZEM-004T untuk pengujian :

```
#include <SoftwareSerial.h>
#include <PZEM004T.h>
PZEM004T pzem1(&Serial1);
PZEM004T pzem2(&Serial2);
PZEM004T pzem3(&Serial3);
IPAddress ip(192,168,1,1);
float sensorteg = 0;
float sensorarus1 = 0;
float sensorarus2 = 0;
float sensorarus3 = 0;
float sensordaya1 = 0;
float sensordaya2 = 0;
float sensordaya3 = 0;
float cospi1 =1;
float cospi2 =1;
float cospi3 =1;
void setup (){
```

```
        pzem1.setAddress(ip);
        pzem2.setAddress(ip);
        pzem3.setAddress(ip);}
void tampilarusteg()
{ //input_keypad();
  dtostrf(sensorarus1,2,2,lcdbuff2); sprintf(lcdbuff,"%sA ",lcdbuff2);
  lcd.setCursor (1,2); lcd.print (lcdbuff);
  dtostrf(sensorarus2,2,2,lcdbuff2); sprintf(lcdbuff,"%sA ",lcdbuff2);
  lcd.setCursor (7,2); lcd.print (lcdbuff);
  dtostrf(sensorarus3,2,2,lcdbuff2);
  sprintf(lcdbuff,"%sA ",lcdbuff2);
  lcd.setCursor (13,2); lcd.print (lcdbuff);
  dtostrf(sensorteg,2,2,lcdbuff2);
  sprintf(lcdbuff,"%sVAC ",lcdbuff2);
  lcd.setCursor (1,1); lcd.print (lcdbuff);}
 Void hitungkWh (){
 sensordaya1= pzem2.power(ip); sensordaya2= pzem3.power(ip);
 sensordaya3= pzem1.power(ip);  sensorteg = pzem2.voltage(ip);
  sensorarus1 = pzem2.current(ip);
  if (sensorarus1<0){sensorarus1=0;cospi1=1;}
sensorarus2 = pzem3.current(ip);
  if (sensorarus2<0){sensorarus2=0;cospi2=1;}
  sensorarus3 = pzem1.current(ip);
  if (sensorarus3<0){sensorarus3=0;cospi3=1;}
  if (sensorarus1>0){cospi1=sensordaya1/(sensorteg*sensorarus1);}
  if (sensorarus2>0){cospi2=sensordaya2/(sensorteg*sensorarus2);}
  if (sensorarus3>0){cospi3=sensordaya3/(sensorteg*sensorarus3);}  }
 Void loop (){ tampilkwh();tampilarusteg();}
```

Dalam pengujian sensor PZEM-004T terhadap tegangan pada alat ini yang fungsinya untuk menampilkan nilai tegangan pada beban yang diukur dan di*monitor*ing. Gambar pengujian sensor PZEM-004T terhadap tegangan ditunjukkan pada gambar 4.16.



Gambar 4.16 Rangkaian elektrikal pengujian sensor PZEM-004T terhadap tegangan

Hasil pengujian dari modul sensor PZEM-004T terhadap tegagan dari beban yang ditampilkan pada *LCD*, diukur dengan alat ukur multimeter da terukur dari *voltmeter digital*. Ditunjukkan pada gambar 4.17, gambar 4.18, gambar 4.19.

Gambar 4.17 Pengukuran mutimeter pada tegangan *input* sensor PZEM-004T

Gambar 4.18 Pengukuran sensor PZEM-004T terhadap tegangan beban tampilan di *LCD*

Gambar 4.19 Pengukuran sensor PZEM-004T terhadap tegangan beban tampilan di *Wattmeter Digital*

Pengukuran ini dilakukan untuk mendapatkan data dari modul sensor PZEM-004T, sebagai perbandingan atau acuan untuk mendapatkan nilai hasil data yang akurat serta mengurangi nilai persentase error pada perhitungan yang digunakan pada alat ini. Setelah dilakukan pengukuran tegangan pada tiap- tiap beban, maka data yang telah didapatkan dari hasil pengukuran yang telah dilaksanakan akan dicantumkan didalam tabel yang ditunjukkan pada tabel 4.5.

Tabel 4.5 Data hasil pengujian sensor PZEM-004T terhadap tegangan

Tegangan *Input* Sensor PZEM-004T	Tegangan Beban	Tegangan VCC Sensor PZEM-004T	Pengukuran Sensor PZEM-004T	*Error* %
227 V	224 V	5,15 V	224 V	0
227 V	224 V	5,15 V	225 V	0.4
227 V	225 V	5,15 V	225 V	0

Dari data diatas didapat hasil pengukuran sensor PZEM-004T terhadap tegangan, baik itu hasil pengukuran dari sensor ataupun dari multimeter. Untuk menentukan

persentase *error* modul sensor PZEM-004T terhadap tegangan dapat dilakukan dengan menggunakan perhitungan sebagai berikut :

$$\textit{error}\text{ sensor PZEM-004T } = \left|\frac{nilai\ \ tegangan\ \ beban - nilai\ \ ukur\ \ sensor}{nilai\ \ tegangan\ \ beban}\right| X100\% \ldots(7)$$

Dibawah ini merupakan perhitungan persentase *error* modul sensor PZEM-004T terhadap tegangan :

- Hasil ukur multimeter terhadap tegangan beban

$$\textit{error}\text{ sensor terhadap tegangan } = \left|\frac{224V - 224V}{224V}\right| X100\% = 0\%$$

$$\textit{error}\text{ sensor terhadap tegangan } = \left|\frac{224V - 225V}{224V}\right| X100\% = 0.4\%$$

$$\textit{error}\text{ sensor terhadap tegangan } = \left|\frac{225V - 225V}{224V}\right| X100\% = 0\%$$

Dari data tabel pengujian dan dari hasil perhitungan persentase error yang dilakukan dapat di analisa bahwa modul sensor PZEM-004T memiliki sensitivitas yang tinggi dan memiliki persentase error yang sangat kecil sehingga sensor PZEM-004T ini layak digunakan pada proyek akhir ini untuk melakukan proses monitoring pengukuran arus dan tegangan beban yang akan digunakan untuk acuan dalam pengukuran dan perhitungan daya pemakaian listrik.

4.2.3.4.5 Pengujian Sensor Cos φ

Setelah melakukan proses perancangan dan pembuatan dari sensor PZEM-004T yaitu proses merancang skematik. Maka dilakukanlah proses instalasi atau pengkoneksian antara *Arduino mega 2560* dan sensor PZEM-004T. Sebelum

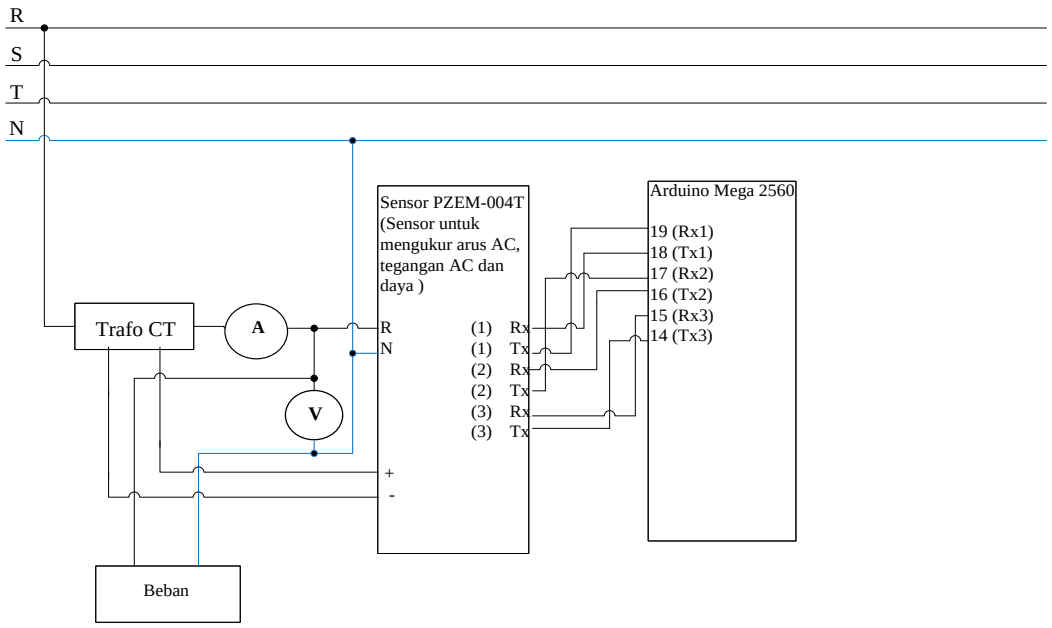
menghubungkan sensor PZEM-004T kepada *Arduino Mega 2560*, sensor terlebih dahulu diuji. Pengujian ini dilakukan untuk menentukan nilai dari Cos φ yang terdapat pada masing – masing beban. Pengujian ini berisi perhitungan yang digunakan utuk menentukan nilai Cos φ dengan menggunakan rumus yang akan dijelaskan pada sub bab ini. Berikut ini program sensor PZEM-004T untuk pengujian:

```
#include <SoftwareSerial.h>
#include <PZEM004T.h>
PZEM004T pzem1(&Serial1);
PZEM004T pzem2(&Serial2);
PZEM004T pzem3(&Serial3);
IPAddress ip(192,168,1,1);
float sensorteg = 0;
float sensorarus1 = 0;
float sensorarus2 = 0;
float sensorarus3 = 0;
float sensordaya1 = 0;
float sensordaya2 = 0;
float sensordaya3 = 0;
float cospi1 =1;
float cospi2 =1;
float cospi3 =1;
void setup (){
      pzem1.setAddress(ip);
      pzem2.setAddress(ip);
      pzem3.setAddress(ip);}
void tampilarusteg()
{ //input_keypad();
  dtostrf(sensorarus1,2,2,lcdbuff2); sprintf(lcdbuff,"%sA ",lcdbuff2);
```

```
 lcd.setCursor (1,2);
 lcd.print (lcdbuff);
 dtostrf(sensorarus2,2,2,lcdbuff2); sprintf(lcdbuff,"%sA ",lcdbuff2);
 lcd.setCursor (7,2);
 lcd.print (lcdbuff);
 dtostrf(sensorarus3,2,2,lcdbuff2);
 sprintf(lcdbuff,"%sA ",lcdbuff2);
 lcd.setCursor (13,2);
lcd.print (lcdbuff);
 dtostrf(sensorteg,2,2,lcdbuff2);
 sprintf(lcdbuff,"%sVAC ",lcdbuff2);
 lcd.setCursor (1,1);
 lcd.print (lcdbuff);}
Void hitungkWh (){
sensordaya1= pzem2.power(ip); sensordaya2= pzem3.power(ip);
sensordaya3= pzem1.power(ip);  sensorteg = pzem2.voltage(ip);
 sensorarus1 = pzem2.current(ip);
 if (sensorarus1<0){sensorarus1=0;cospi1=1;}
sensorarus2 = pzem3.current(ip);
 if (sensorarus2<0){sensorarus2=0;cospi2=1;}
 sensorarus3 = pzem1.current(ip);
 if (sensorarus3<0){sensorarus3=0;cospi3=1;}
 if (sensorarus1>0){cospi1=sensordaya1/(sensorteg*sensorarus1);}
 if (sensorarus2>0){cospi2=sensordaya2/(sensorteg*sensorarus2);}
 if (sensorarus3>0){cospi3=sensordaya3/(sensorteg*sensorarus3);}  }
Void loop (){ tampilkwh();tampilarusteg();}
```

Dalam pengujian sensor PZEM-004T terhadap Cos φ pada alat ini yang fungsinya untuk menghitung nilai Cos φ pada beban yang diukur dan Cos φ ini akan

digunakan untuk perhitungan pemakaian listrik sesuai dengan daya yang digunakan. Gambar pengujian sensor PZEM-004T terhadap Cos φ ditunjukkan pada gambar 4.20.



Gambar 4.20 Rangkaian elektrikal pengujian sensor PZEM-004T terhadap Cos φ

Data tegangan yang digunakan untuk perhitungan rata-rata adalah tabel hasil pengukuran sensor PZEM-004T dapat dilihat pada tabel 4.5. Dan rumus perhitungan rata-rata dinyatakan pada persamaan (8).

$$\text{Rata-rata} = \frac{Jumla\ h\ Data}{Banyak\ \ data} \quad \text{.....(8)}$$

Berikut ini adalah perhitungan yang dilakukan untuk mendapatkan tegangan rata-rata yang digunakan dalam perhitungan Cos φ ditunjukkan pada tabel 4.5.

$$\text{Rata-rata} = \frac{Jumla\ h\ Data}{Banyak\ \ Data} = \frac{224\ V+225\ V+225\ V}{3} = 224{,}6\ \text{V}$$

Sedangkan untuk data arus yang digunakan dalam tabel hasil pengujian keseluruhan adalah tabel hasil pengukuran sensor PZEM-004T terhadap arus yang dapat dilihat pada tabel 4.4.

Untuk menghitung daya dalam satuan *Watt*, maka diperlukan faktor daya ataupun yang dikenal dengan Cos φ, dalam perhitungan daya harus menggunakan Cos φ karena jika hanya menggunakan rumus daya semu daya belum dikatakan satuan *Watt*. Rumus yang digunakan sebagai berikut :

$$\text{Cos } \varphi = P/S = P/V*I \quad \text{.....(10)}$$

Keterangan : Cos φ = Faktor Daya
P = Daya pada beban (W)
S = Daya Semu (VA)
V = Tegangan Beban (V)
I = Arus Beban (A)

Dibawah ini merupakan perhitungan serta rumus Cos φ pengukuran pada pengujian sensor :

- Cos phi untuk beban kipas angin
  Cos phi = P/S = P/V*I = 42 W/224,6V*0.22A = 0.85

- Cos phi untuk beban cas laptop
  Cos phi = P/S = P/V*I = 18 W/ 224.6V*0.10 A= 0,80

- Cos phi untuk beban setrika
  Cos phi = P/S = P/V*I = 300 W/ 224.6V*1.55 A= 0,86

Tabel 4.6 Hasil pengujian sensor Cos φ

Beban Yang Digunakan	Hasil Pengukuran Sensor PZEM-004T		Aplikasi (Daya dan *kWhP*pulsa)			*Error* %	Cos φ
	V(V)	I(A)	P(W)	awal	akhir		
Kipas angin (42W)	224,6	0,22	41	13,6	13,58	2	0,85
Cas laptop (18W)	224,6	0,10	17	13,6	13,58	5,5	0,80
Setrika (300W)	224,6	1,55	322	13,6	13,58	7	0,86

Dari data tabel 4.6 diatas, penulis dapat mengaalisa dan dapat menyimpulkan bahwa sensor Cos φ yang dibuat dalam pembuatan proyek akhir ini layak digunakan. Karena hasil sensor Cos φ ini dalam perhitungannya sangat baik untuk digunakan pada beban dan Cos φ yang dihasilkan tidak kurang dari batas minimal Cos φ yang digunakan pada beban.

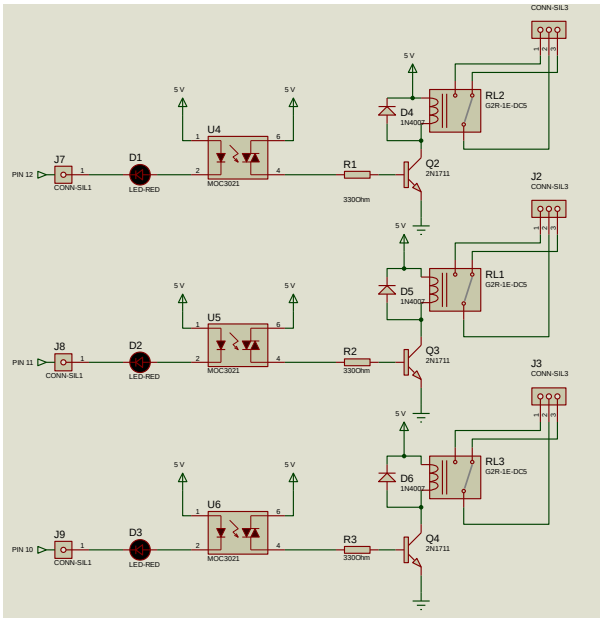
4.2.3.5 Perancangan dan Pembuatan *Relay 5v*

Dalam proses ini terdapat tiga proses yang dilakukan yaitu proses yang pertama adalah proses perancangan relay 5v dan proses yang kedua adalah proses pembuatan relay 5v. Dalam proses perancangan relay 5v, penulis melakukan proses merancang skematik rangkaian dan *pin-pin* yang digunakan relay 5v.

Dalam proses pembuatan relay 5v, akan dilakukan jika proses pertama telah selesai. Penulis melakukan instalasi atau menghubungkan kabel *jumper* dari *pin-pin* yang digunakan relay 5v ke *Arduino Mega 2560*. Dan setelah melakukan proses instalasi maka penulis akan menguji dengan melakukan pemograman yang akan digunakan dalam alat ini.

4.2.3.5.1 Perancangan *Relay 5V*

Berikut adalah gambar skematik pada perancangan *relay 5v* alat prototipe *monitoring* pemakaian listrik *via android* ditunjukkan pada gambar 4.16.



Gambar 4.21 Skematik *relay 5v*

Gambar 4.20 Skematik dari relay 5v ini menjelaskan *pin* yang digunakan. Seperti pada gambar skematik diatas berikut adalah tabel penjelasan *pin* yang terhubung dari relay 5v ke *Arduino Mega 2560* ditunjukkan pada tabel 4.6.

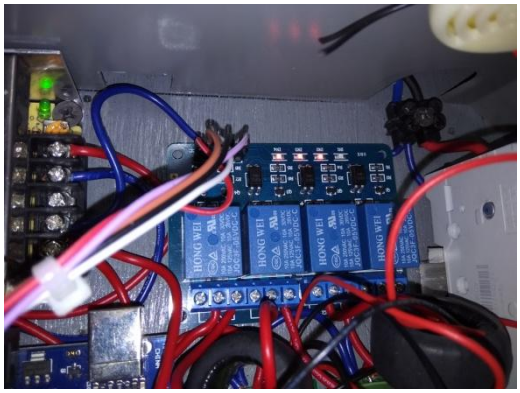
Tabel 4.7 Tabel koneksi *relay 5v* ke *Arduino Mega 2560*

No	Pin *relay 5v*	Pin *Arduino Mega 2560*
1	VCC	Pin 5v
2	IN1	Pin 12
3	IN2	Pin 11
4	IN3	Pin 10
5	GND	Pin GND

4.2.3.5.2 Pembuatan *Relay 5v*

Pembuatan Modul relay 5v diputuskan dengan membeli modul yang telah jadi dan yang mudah didapatkan di pasaran. Hal ini dikarenakan modul yang dijual

secara umum dapat langsung digunakan tanpa harus membuat modul relaytersebut, hal ini tentu dapat meminimalisir waktu dalam proses pengerjaan proyek akhir. Gambar 4.22 menunjukan modul relay 5v yang sudah terpasang pada prototipe *monitoring* pemakaian listrik *via android*.



Gambar 4.22 Pemasagan *relay 5v*

4.2.3.5.3 Pengujian *Relay 5v*

Setelah melakukan proses perancangan dan pembuatan dari relay 5v yaitu proses merancang skematik. Maka dilakukanlah proses pengujian antara *Arduino mega 2560* dan relay 5v. Pembuatan *hardware* relay 5v dilakukan dengan cara menghubungkan relay 5v ke *Arduino Mega 2560.* Sebelum dilakukan pembuatan relay 5v terlebih dahulu dilakukan pengujian relay tersebut. Berikut adalah program pengujian relay 5v*:*

```
// menamai pin relay
#define relay1pin 12 // relay1
#define relay2pin 11 // relay2
#define relay3pin 10 // relay3
void setup (){pinMode (relay1pin,OUTPUT);
pinMode (relay2pin,OUTPUT);pinMode (relay3pin,OUTPUT);
```

pinMode (buzzer,OUTPUT);digitalWrite (relay1pin,HIGH);
digitalWrite (relay2pin,HIGH);digitalWrite (relay3pin,HIGH);}
Void loop (){
If(kwhsimpan<=0.0){digitalWrite(relay1pin,LOW);digitalWrite(relay2pin,LOW);digitalWrite(relay3pin,LOW);digitalWrite(ledpin1,HIGH);delay(100);
digitalWrite(ledpin1,LOW);};

Perintah untuk program diatas adalah jika pulsa *kWh* = 0, maka relay akan mematikan semua beban dan relay akan berlogika *HIGH* atau led indikator relay akan menyala. Berikut adalah hasil pengujian relay 5v terhadap pulsa *kWh* yang ditunjukkan pada tabel 4.8 :

Tabel 4.8 Tabel pengujian *relay 5v*

No	Kondisi	Relay 1	Relay 2	Relay 3
1	0 *kWh*	1	1	1
2	>0 *kWh*	0	0	0

Dari data tabel 4.7 diatas, penulis dapat menganalisa dan menyimpulkan bahwa relay 5v yang digunakan dalam proyek akhir ini layak untuk digunakan dalam pembuatan proyek akhir ini. Karena penulis melakukan pengujian terhadap relay 5v ini dengan memberikan perintah atau instruksi ke Arduino Mega 2560 yang akan diproses relay 5v tersebut. Perintah tersebut adalah ketika kondisi alat tidak ada pulsa atau 0 *kWh* maka relay 5v akan aktif *HIGH.* Ketika kondisi alat terdapat pulsa *kWh* atau pulsa *kWh* > 0 *kWh* maka relay 5v akan berlogika *LOW.*

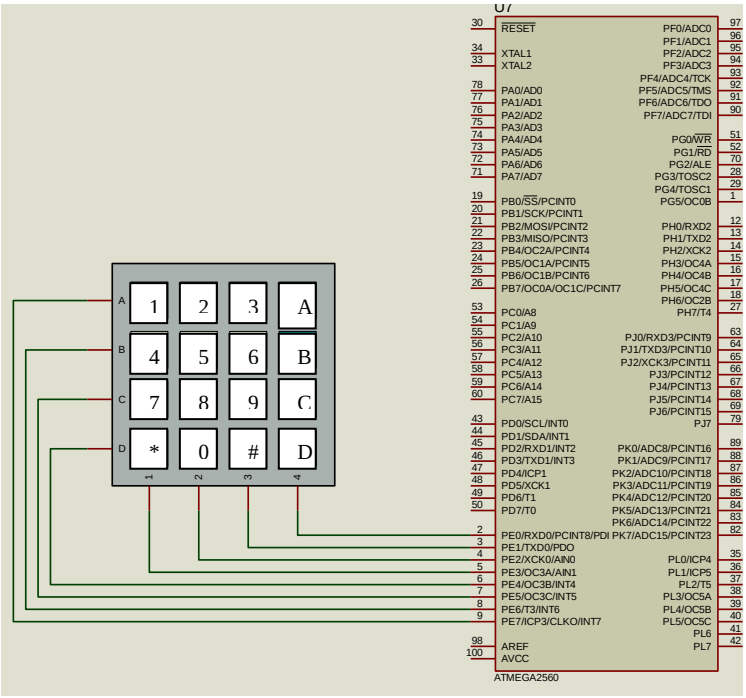
4.2.3.6 Perancangan dan Pembuatan *Keypad 4x4*

Dalam proses ini terdapat tiga proses yang dilakukan yaitu Proses perancangan *keypad 4x4* meliputi proses merancang dari pin-pin yang digunakan

untuk *keypad 4x4.* Sedangkan proses pembuatan meliputi proses instalasi atau pemasangan kabel jumper dari *keypad 4x4* ke *arduino mega 2560.* Dan proses terakhir adalah proses pengujian dari *keypad 4x4.*

4.2.3.6.1 Perancangan *Keypad 4x4*

Berikut adalah gambar skematik pada perancangan *keypad 4x4* alat prototipe *monitoring* pemakaian listrik *via android*. Skematik ini akan mendukung dari proses pengujian yang akan dilakukan yang ditunjukkan pada gambar 4.23.



Gambar 4.23 Skematik *keypad 4x4*

Dalam proses perancangan *keypad 4x4* ini, penulis merancang skematik dari *keypad 4x4.* Skematik dari *keypad 4x4* ini menjelaskan *pin* yang digunakan dalam pengkoneksian untuk diuji pada saat proses pengujian. Seperti pada gambar 4.23 skematik *keypad 4x4* diatas berikut adalah tabel penjelasan *pin* yang terhubung dari keypad 4x4 ke *Arduino Mega 2560* ditunjukkan pada tabel 4.9.

Tabel 4.9 Tabel koneksi pin *keypad 4x4* ke *Arduino Mega 2560*

No	Pin *keypad 4x4*	Pin *Arduino Mega 2560*
1	A (baris 1)	Pin 9
2	B (baris 2)	Pin 8
3	C (baris 3)	Pin 7
4	D (baris 4)	Pin 6
5	1 (kolom 1)	Pin 5
6	2 (kolom 2)	Pin 4
7	3 (kolom 3)	Pin 3
8	4 (kolom 4)	Pin 2

4.2.3.6.2 Pembuatan *Keypad 4x4*

Pembuatan *keypad 4x4* diputuskan dengan membeli modul yang telah jadi dan mudah didapat dipasaran. Adapun *keypad 4x4* yang digunakan pada proyek akhir dapat dilihat pada gambar 4.24 berikut.

Gambar 4.24 Pemasangan *keypad 4x4*

4.2.3.6.3 Pengujian *Keypad 4x4*

Setelah melakukan proses perancangan dari *keypad 4x4* yaitu proses merancang skematik. Pembuatan *hardware keypad 4x4* dilakukan dengan cara

menghubungkan *keypad 4x4* ke *Arduino Mega 2560*. Sebelum dilakukan pembuatan *keypad 4x4* terlebih dahulu dilakukan pengujian bluetooth hc-06 tersebut. Berikut adalah program pengujian *keypad 4x4*:

```
#include <Keypad.h>
char array[10],i=0;////inisisalisasi variabel berupa array untuk jumlah digit input pulsa pada keypad
const byte ROWS = 4; //four rows
const byte COLS = 4; //four columns
//define the cymbols on the buttons of the keypads
char hexaKeys[ROWS][COLS] = {
 {'1','4','7','*'}, {'2','5','8','0'}, {'3','6','9','#'},{'A','B','C','D'} };
byte rowPins[ROWS] = {9, 8, 7, 6}; //connect to the row pinouts of the keypad
byte colPins[COLS] = {5, 4, 2, 3}; //connect to the column pinouts of the keypad
Keypad customKeypad = Keypad( makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS);
void simpan_nilai(){ if(i==1) {nilai=array[i];}
if(i>=2 && i<=11){nilai=(nilai*10)+array[i];}}
//fungsi untuk membaca semua data yg di input pada keypad
void input_keypad() { if (i<1) {i=0;nilai=0;}
if (i>6){nilai=(nilai-array[i])/10;i--;} if (i>0&&nilai==0){i=0;}
char customKey = customKeypad.getKey();
switch (customKey)
 { //case NO_KEY://break; case '1':
 i++;array[i]=0;
simpan_nilai();
digitalWrite (buzzer,HIGH);
```

delay(100);
digitalWrite (buzzer,LOW);break;};
Void loop (){ simpan_nilai(); input_keypad();}

Setelah dihubungkan, *keypad 4x4* akan melakukan perintah sesuai dengan program yang diatas. Berikut adalah hasil pengujian *keypad 4x4* yang ditampilkan ke *LCD* ditunjukkan pada tabel 4.10.

Gambar 4.10 Tabel pengujian *keypad 4x4*

No	*Input Keypad*	*Output* pada LCD
1	0	0
2	1	1
3	2	2
4	3	3
5	4	4
6	5	5
7	6	6
8	7	7
9	8	8
10	9	9

Dari data tabel 4.9 diatas, penulis dapat menganalisa dan dapat menyimpulkan data yang didapatkan dari hasil pengujian ini bahwa keypad 4x4 yang digunakan pada proyek akhir ini layak untuk digunakan sebagai media penginputan token pulsa *kWh* dan sebagai media untuk penginputan batas daya maksimal. Karena pada pengujian ini penulis melakukan 10 kali pengujian dengan menginputkan angka pada keypad 4x4 yang terdiri dari angka 0 sampai angka 9 yang akan ditampilkan di LCD dan hasil yang ditampilkan di LCD sesuai dengan penginputan.

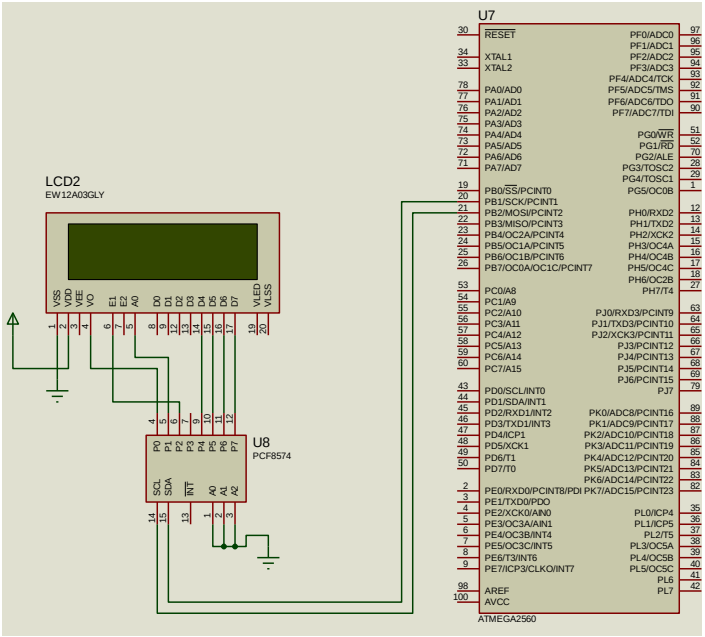
4.2.3.7 Perancangan dan Pembuatan LCD I2C 20x4

Dalam proses ini terdapat dua proses yang dilakukan yaitu proses yang pertama adalah proses perancangan LCD I2C 20x4 dan proses yang kedua adalah proses pembuatan LCD I2C 20x4. Dalam proses perancangan LCD I2C 20x4, penulis melakukan proses merancang skematik rangkaian dan *pin-pin* yang digunakan LCD I2C 20x4.

Dalam proses pembuatan LCD I2C 20x4, akan dilakukan jika proses pertama telah selesai. Penulis melakukan instalasi atau menghubungkan kabel *jumper* dari *pin-pin* yang digunakan LCD I2C 20x4 ke *Arduino Mega 2560*. Dan setelah melakukan proses instalasi maka penulis akan menguji dengan melakukan pemograman yang akan digunakan dalam alat ini.

4.2.3.7.1 Perancangan LCD I2C 20x4

Berikut adalah gambar skematik pada perancangan LCD I2C 20x4 alat prototipe *monitoring* pemakaian listrik *via android* ditunjukkan pada gambar 4.25.



Gambar 4.25 Skematik LCD I2C 20x4

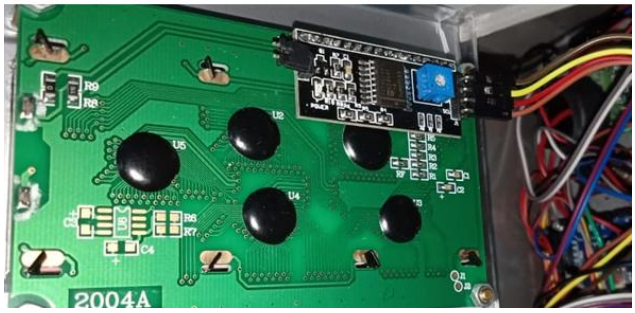
Dalam proses perancangan LCD I2C 20x4 ini, penulis merancang skematik dari LCD I2C 20x4. Gambar 4.25 Skematik LCD I2C 20x4 ini menjelaskan *pin* yang digunakan. Seperti pada gambar skematik diatas berikut adalah tabel penjelasan *pin* yang terhubung dari LCD I2C 20x4 ke *Arduino Mega 2560* yag ditunjukkan pada tabel 4.11.

Tabel 4.11 Tabel koneksi LCD I2C 20x4 ke *Arduino Mega 2560*

No	Pin LCD I2C 20x4	Pin *Arduino Mega 2560*
1	VCC	Pin 5v
2	Pin SDA	Pin 20
3	Pin SCL	Pin 21
4	GND	Pin GND

4.2.3.7.2 Pembuatan LCD I2C 20x4

Pembuatan LCD I2C 20x4 diputuskan dengan membeli modul yang telah jadi dan mudah didapat dipasaran. Hal ini berguna untuk meminimalisir waktu dalam pembutan tugas akhir ini. Dan juga hal ini berguna untuk meminimalisir error atau gagal dalam pembuatan alat ini dan jika hanya membeli di pasaran, LCD ini hanya dihubungkan langsung ke Arduino Mega 2560 dengan kabel jumper yang dapat dibeli di pasaran. Adapun LCD I2C 20x4 yang digunakan pada proyek akhir dapat dilihat pada gambar 4.26 berikut.



Gambar 4.26 Pemasangan LCD I2C 20x4

4.2.3.7.3 Pengujian LCD I2C 20x4

Pembuatan *hardware* LCD I2C 20x4 dilakukan dengan cara menghubungkan LCD I2C 20x4 ke *Arduino Mega 2560*. Sebelum dilakukan pembuatan LCD I2C 20x4 terlebih dahulu dilakukan pengujian LCD I2C 20x4 tersebut. Berikut adalah program pengujian LCD I2C 20x4:

```
#include <Wire.h> //menambahkan library i2c
#include <LiquidCrystal_I2C.h>//menambahkan library i2c lcd
LiquidCrystal_I2C lcd ( 0x3f, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);
void setup (){Wire.begin ; lcd.begin (20,4);
lcd.setCursor (3,0);lcd.print ("MOCHAMAD RIVAI");lcd.setCursor (3,1);lcd.print ("DWI RETNO A"); lcd.setCursor (1,2);lcd.print ("TUGAS AKHIR 2018");delay(1000);lcd.clear(); }Void loop () {}
```

Berikut adalah tabel pengujian LCD ditunjukkan pada tabel 4.12.

Tabel 4.12 Tabel pengujian LCD I2C 20x4

No	Perintah	Tampialn di LCD
1	Menampilkan Nama	
2	Menampilkan Arus beban, Tegangan beban, Pulsa *kWh* dan Penginputan token pulsa *kWh*	

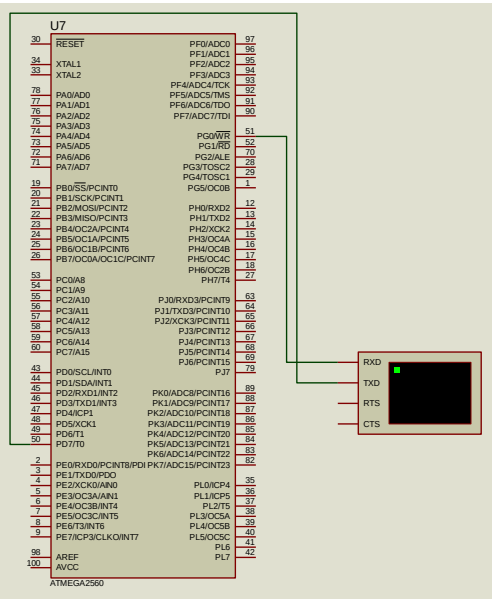
Dari data tabel 4.12 diatas, penulis dapat menganalisa dan dapat menyimpulkan bahwa LCD I2C 20x4 layak digunakan pada proyek akhir ini. Karena perintah yang diinstruksikan sesuai dengan tampilan yang ada di LCD.

4.2.3.8 Perancangan dan Pembuatan *Bluetooth HC-06*

Dalam proses ini terdapat dua proses yang dilakukan yaitu proses yang pertama adalah proses perancangan *Bluetooth HC-06* dan proses yang kedua adalah proses pembuatan *Bluetooth HC-06.*

4.2.3.8.1 Perancangan *Bluetooth HC-06*

Berikut adalah gambar 4.27 skematik pada perancangan dan pembuatan *Bluetooth HC-06* alat prototipe *monitoring* pemakaian listrik *via android* :



Gambar 4.27 Skematik *bluetooth HC-06*

Gambar 4.27 Skematik dari *Bluetooth HC-06* ini menjelaskan *pin* yang digunakan. Seperti pada gambar skematik diatas berikut adalah tabel penjelasan *pin*

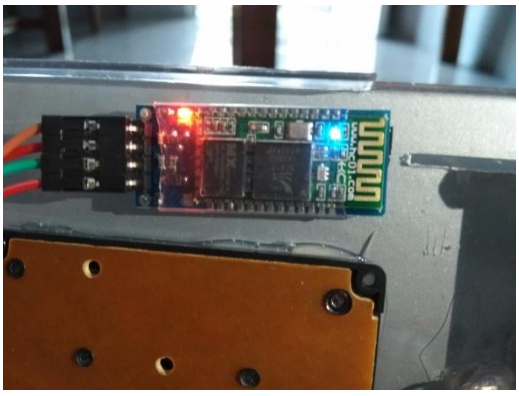
yang terhubung dari *Bluetooth hc-06* ke *Arduino Mega 2560* yang ditunjukkan pada tabel 4.13.

Tabel 4.13 Koneksi pin *Bluetooth HC-06* ke *Arduino Mega 2560*

No	Pin *Bluetooth HC-06*	Pin *Arduino Mega 2560*
1	VCC	Pin 5v
2	Pin Rx	Pin 51
3	Pin Tx	Pin 50
4	GND	Pin GND

4.2.3.8.2 Pembuatan *Bluetooth HC-06*

Pembuatan *Bluetooth HC-06* diputuskan dengan membeli modul *Bluetooth HC-06* yang telah jadi dan mudah didapat dipasaran. Pemasangan *Bluetooth HC-06* pada Prototipe *Monitoring* Pemakaian Listrik *Via Android* dapat dilihat pada gambar 4.28 berikut.



Gambar 4.28 Pemasangan *Bluetooth HC-06*

4.2.3.8.3 Pengujian *Bluetooth HC-06*

Pembuatan *hardware Bluetooth HC-06* dilakukan dengan cara menghubungkan *Bluetooth HC-06* ke *Arduino Mega 2560*. Sebelum dilakukan

pembuatan *Bluetooth HC-06* terlebih dahulu dilakukan pengujian *Bluetooth HC-06* tersebut. Berikut adalah program pengujian *Bluetooth HC-06*:

```
#include <SoftwareSerial.h>
SoftwareSerial btSerial(50, 51); // RX, TX
void setup ()
{btSerial.begin(9600);}
Void loop (){}
```

Sedangkan hasil pengujian jarak penerimaan data oleh *Bluetooth* HC-06 yang telah terpasang pada alat dapat dilihat pada tabel 4.14 berikut.

.

4.14 Tabel pengujian jarak *Bluetooth HC-06*

No	Jarak	Data Diterima
1	1 Meter	Ya
2	3 Meter	Ya
3	5 Meter	Ya
4	7 Meter	Ya
5	9 Meter	Ya
6	11 Meter	Ya
7	13 Meter	Ya
8	15 Meter	Ya
9	17 Meter	Ya
10	19 Meter	Ya
11	21 Meter	Tidak

Dari data tabel 4.13 diatas, pengujian yang dilakukan dapat dianalisa bahwa *Bluetooth* HC-06 yang digunakan dapat menerima data yang dikirim oleh aplikasi

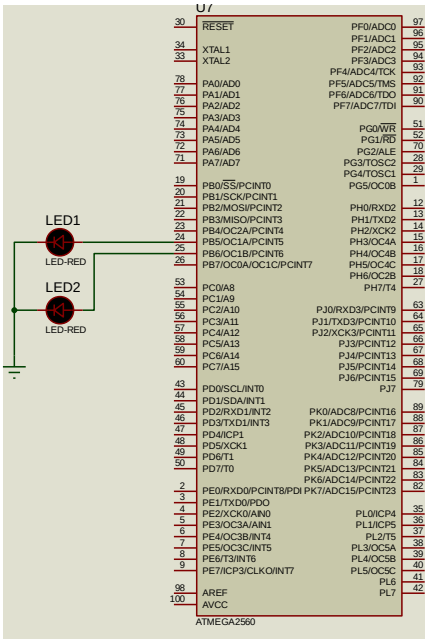
pada Android hingga jarak 20 meter. Pada pengujian jarak 21 meter *Bluetooth* HC-06 tidak lagi dapat menerima data yang dikirim oleh aplikasi pada Android. Dari hasil pengujian dan analisa dapat disimpulkan bahwa *Bluetooth* yang digunakan sesuai dengan perancangan yang diinginkan.

4.2.3.9 Perancangan dan Pembuatan Led Indikator

Dalam proses ini terdapat dua proses yang dilakukan yaitu proses yang pertama adalah proses perancangan led indikator, proses yang kedua adalah proses pembuatan led indikator dan proses yang ketiga adalah proses pengujian dari led indikator. Pengujian ini terdiri dari ketika pulsa *kWh* dan ketika pulsa *kWh* habis.

4.2.3.9.1 Perancangan Led Indikator

Berikut adalah gambar skematik pada perancangan led indikator alat prototipe *monitoring* pemakaian listrik *via android* yang ditunjukkan pada gambar 4.29.



Gambar 4.29 Skematik led indikator

Gambar 4.29 Skematik dari led indikator ini menjelaskan *pin* yang digunakan. Led indikator ini terdiri dari dua warna led indikator Seperti pada gambar 4.29 skematik diatas berikut adalah tabel penjelasan *pin* yang terhubung dari led indikator ke *Arduino Mega 2560* yang ditunjukkan pada tabel 4.15.

Tabel 4.15 Tabel pin koneksi led indikator dengan *Arduino Mega 2560*

No	Kaki pada led	Pin *Arduino Mega 2560*
1	Anoda (Led Merah)	Pin 24
2	Anoda (Led Hijau)	Pin 25
3	Katoda (Led Merah dan Led Hijau)	Pin GND

4.2.3.9.2 Pembuatan Led Indikator

Dalam proses ini penulis langsung melakukan proses *soldering* pada led dan kabel *jumper* untuk terhubung ke *Arduino mega 2560*. Berikut ini tabel daftar komponen yang digunakan yang ditunjukkan pada tabel 4.16. Dan gambar pemasangan led indikator ditunjukkan pada gambar 4.30.

Gambar 4.30 Pemasangan led indikator

Tabel 4.16 Tabel daftar komponen dan alat pembuatan led yang digunakan

No	Komponen dan Alat yang digunakan	Jumlah
1	Led merah dan Led hijau	Masing – masing 1 buah
2	Resistor 560 Ohm	2 buah
3	Solder 40 W	1 buah

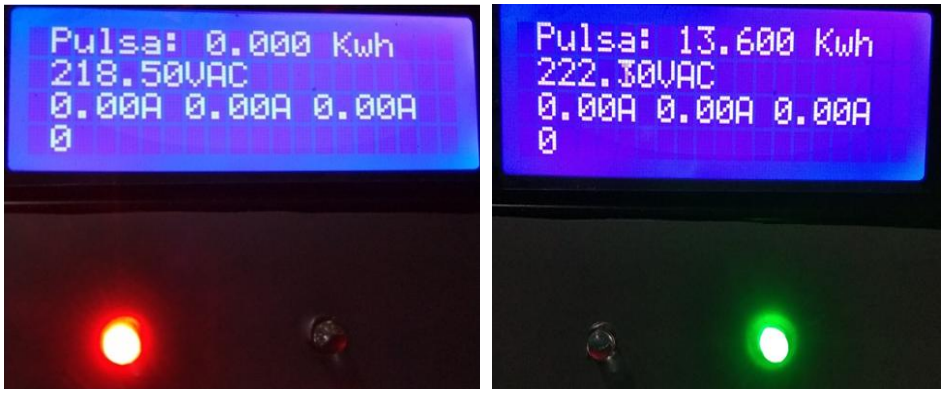
4	Timah solder	Secukupnya
5	Kabel *Shrink Tube*	Secukupnya
6	Kabel *Jumper*	Secukupnya

4.2.3.9.3 Pengujian Led Indikator

Pembuatan *hardware* led indikator dilakukan dengan cara menghubungkan led indikator ke *Arduino Mega 2560.* Sebelum dilakukan pembuatan led indikator terlebih dahulu dilakukan pengujian led indikator tersebut. Berikut adalah program pengujian led indikator:

```
int ledpin1 = 25;
int ledpin2 = 24;
void setup ()
{ pinMode (ledpin1,OUTPUT);pinMode (ledpin2,OUTPUT);}
Void loop ()
 {if (kwhsimpan<=0.0){ digitalWrite(ledpin1,HIGH);}}
```

Berikut adalah hasil pengujian led indikator ketika pulsa *kWh* = 0 dan pulsa *kWh* >0 *kWh* yang ditunjukkan pada gambar 4.31.



Gambar 4.31 Hasil pengujian led indikator

Berikut ini adalah tabel hasil pengujian terhadap led indikator pada alat prototipe monitoring pemakaian listrik via android yang ditunjukkan pada tabel 4.17.

Tabel 4.17 Tabel pengujian Led Indikator

No	Pulsa *kWh*	Led Merah	Led Hijau
1	0 *kWh*	1	0
2	>0 *kWh*	0	1

Dari data tabel diatas, penulis dapat menganalisa dan menyimpulkan bahwa led indikator yang kami buat sesuai dengan fungsinya. Karena ketika pulsa *kWh* tersebut bernilai 0 *kWh* maka led indikator yang berwarna merah akan menyala. Dan ketika pulsa *kWh* tersebut bernilai >0 *kWh* maka led indikator yang berwarna hijau akan menyala dan led indikator berwarna merah akan padam.

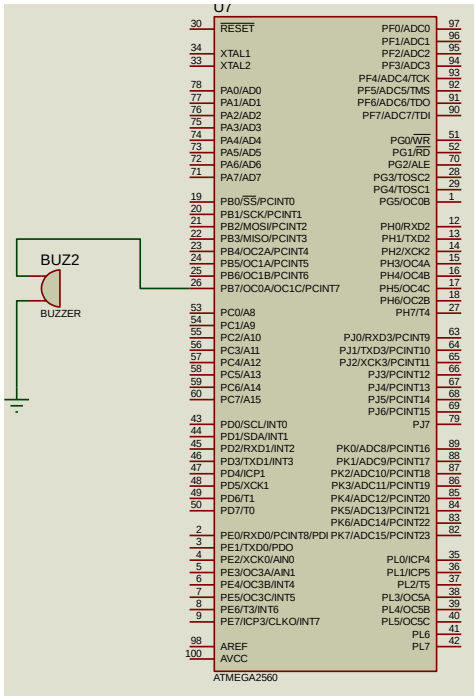
4.2.3.8 Perancangan dan Pembuatan *Buzzer*

Dalam proses ini terdapat dua proses yang dilakukan yaitu proses yang pertama adalah proses perancangan led indikator dan proses yang kedua adalah proses pembuatan led indikator.

Dalam proses pembuatan *buzzer*, akan dilakukan jika proses pertama telah selesai. Penulis melakukan instalasi atau menghubungkan kabel *jumper* dari *pin-pin* yang digunakan *buzzer* ke *Arduino Mega 2560*. Dan setelah melakukan proses instalasi maka penulis akan menguji dengan melakukan pemograman yang akan digunakan dalam alat ini.

4.2.3.8.1 Perancangan *Buzzer*

Berikut adalah gambar skematik pada perancangan dan pembuatan buzzer alat prototipe *monitoring* pemakaian listrik *via android* yang ditunjukkan pada gambar 4.32.



Gambar 4.32 Skematik *buzzer*

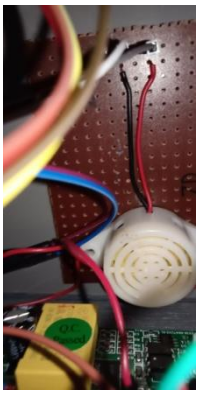
Seperti pada gambar 4.32 skematik *buzzer* diatas berikut adalah tabel penjelasan *pin* yang terhubung dari buzzer ke *Arduino Mega 2560* yang ditunjukkan pada tabel 4.18.

Tabel 4.18 Tabel koneksi pin *Buzzer* ke *Arduino Mega 2560*

No	Kaki pada *buzzer*	Pin *Arduino Mega 2560*
1	Positif	Pin 26
2	Negatif	Pin GND

4.2.3.8.2 Pembuatan *Buzzer*

Dalam proses ini penulis langsung melakukan proses *soldering* pada buzzer di pcb berlubang untuk terhubung ke *Arduino mega 2560*. Berikut ini tabel daftar komponen yang digunakan yang ditunjukkan pada tabel 4.19. Dan gambar pemasangan buzzer ditunjukkan pada gambar 4.33.



Gambar 4.33 Pemasangan *buzzer*

Tabel 4.19 Tabel daftar komponen dan alat *buzzer yang* digunakan

No	Komponen dan alat yang digunakan	Jumlah
1	Buzzer 5v	1 buah
2	Solder 40 W	1 buah
3	Timah solder	Secukupnya
4	PCB berlubang	Secukupnya
5	Kabel jumper	Secukupnya

4.2.3.8.3 Pengujian *Buzzer*

Pembuatan *hardware buzzer* dilakukan dengan cara menghubungkan *buzzer* ke *Arduino Mega 2560.* Sebelum dilakukan pembuatan *buzzer* terlebih dahulu dilakukan pengujian *buzzer* tersebut. Berikut adalah program pengujian *buzzer*:

```
int buzzer = 26;void setup () {pinMode (buzzer,OUTPUT);}
Void input_keypad ()
{  if (i<1) {i=0;nilai=0;} if (i>6)  {nilai=(nilai-array[i])/10;i--; }
if (i>0&&nilai==0) {i=0;}
char customKey = customKeypad.getKey(); switch (customKey)
{  //case NO_KEY: //break; case '0':   i++;array[i]=0;simpan_nilai();
```

digitalWrite(buzzer,HIGH);delay(100);digitalWrite(buzzer,LOW);break;}
Void loop (){ input_keypad();}

Berikut ini adalah tabel hasil pengujian terhadap led indikator pada alat prototipe *monitoring* pemakaian listrik *via android* yang ditunjukkan pada tabel 4.20.

Tabel 4.20 Hasil pengujian *buzzer*

No	*Input Keypad*	Logika	Keterangan
1	1	1	Berbunyi
2	2	1	Berbunyi
3	3	1	Berbunyi
4	4	1	Berbunyi
5	5	1	Berbunyi
6	6	1	Berbunyi
7	7	1	Berbunyi
8	8	1	Berbunyi
9	9	1	Berbunyi
10	0	1	Berbunyi
11	A	1	Berbunyi
12	B	1	Berbunyi
13	C	1	Berbunyi
14	D	1	Berbunyi
15	*	1	Berbunyi
16	#	1	Berbunyi

Dari data tabel diatas, penulis dapat menganalisa dan dapat menyimpulkan bahwa *buzzer* yang dibuat bekerja sesuai dengan fungsinya. Karena fungsi *buzzer* pada alat ini sebagai indikator dalam penginputan keypad untuk pengisian pulsa *kWh*.

4.3 Proses Pembuatan *Software*

Proses ini terdiri dari dua tahapan, tahap pertama yang harus dilakukan adalah merancang bentuk *software* yang akan ditampilkan ke *smartphone*. Lalu tahap kedua yang harus dilakukan dalam pembuatan yaitu adalah pembuatan *software* yang akan ditampilakan ke *smartphone*.

4.3.1 Perancangan *Software* Aplikasi *Blynk*

Dalam proses ini, perancangan *software* menggunakan *smartphone* dengan *BLYNK apps*. Aplikasi ini sudah menyiapkan fitur-fitur yang digunakan dalam pengontrolan *kWh* meter prabayar. Serta keunggulan lain dari aplikasi ini juga adalah pengguna lebih mudah menggunakan aplikasinya karena langsung bisa terkoneksi dengan *arduino*. Dengan cara pada pengaturan fitur-fiturnya *pin*nya diganti dengan *pin* yang ada di *arduino*. Bahasa yang digunakan untuk pemograman pada aplikasi ini mudah dimengerti dan mudah digunakan oleh manusia. Selain bahasa pemograman aplikasi ini dapat mengontrol beberapa perangkat keras yang dapat berkomunikasi secara nirkabel. Perangkat keras yang dapat berkomunikasi secara nirkabel itu antara lain semua macam *arduino*, *ESP8266*, *NodeMCU*, *Bluetooth HC-06* / *HC-05* dan semua macam *raspberry pi*. Berikut penjelasan tentang fitur dalam merancang *Software* untuk *Smartphone* antara lain :

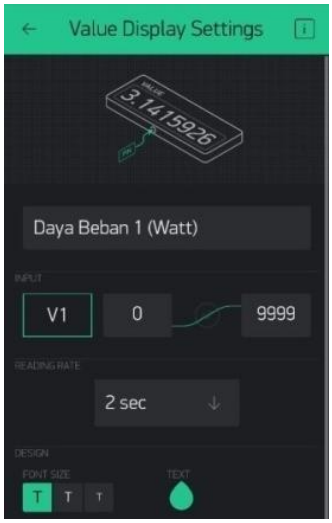
a. Perancangan virtual tampilan pulsa *kWh*
b. Perancangan virtual tampilan daya pada beban 1,
c. Perancangan virtual tampilan daya pada beban 2
d. Perancangan virtual tampilan daya pada beban 3
e. Perancangan virtual tombol *on/off* untuk beban 1,
f. Perancangan virtual tombol *on/off* untuk beban 2
g. Perancangan virtual tombol *on/off* untuk beban 3
h. Perancangan virtual tombol mode *manual/auto*
i. Perancangan virtual koneksi *bluetooth hc-06*

Gambar 4.34 menunjukkan hasil perancangan *software* aplikasi *blynk* yang dibuat dalam pengontrolan prototipe *monitoring* pemakaian listrik *via android*, hasil tersebut antara lain virtual tampilan pulsa, fitur koneksi *bluetooth*, fitur tombol *on/off*, fitur tombol mode otomatis dan manual. Dan virtual penampil daya beban. Gambar 4.35 menunjukkan pengaturan penamaan dan pengaturan *pin* yang digunakan dalam komunikasi dan pengaturan seberapa banyak data yang dikirim. Dalam merancang *virtual* 0 yang berfungsi sebagai penampil dari hasil pengiriman data ke *smartphone* berupa data sisa pulsa *kWh*.

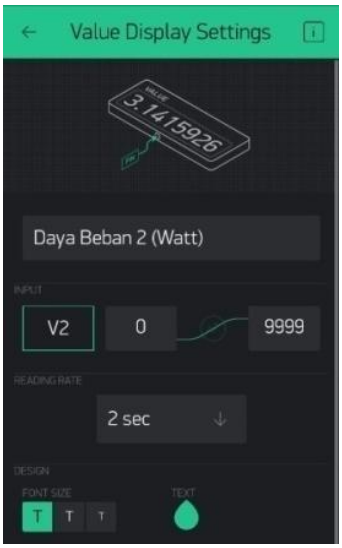
Gambar 4.34 Perancangan *software* untuk *smartphone*

Gambar 4.35 Perancangan *virtual* tampilan pulsa *kWh* di *smartphone*

Dalam merancang *virtual* 1 yang ditunjukkan pada gambar 4.36 berfungsi sebagai penampil dari hasil pengiriman data ke *smartphone* berupa data hasil pengukuran daya beban pada beban 1 dalam satuan *Watt*. Pada gambar 4.37 berfungsi sebagai penampil dari hasil pengiriman data ke *smartphone* berupa data hasil pengukuran daya beban pada beban 2 dalam satuan *Watt*.

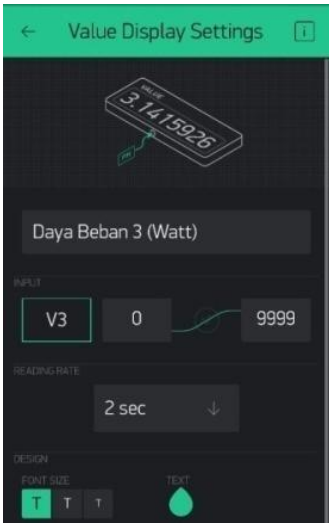


Gambar 4.36 Perancangan *virtual* tampilan daya beban1 (*Watt*) di *smartphone*



Gambar 4.37 Perancangan *virtual* tampilan daya beban2 (*Watt*) di *smartphone*

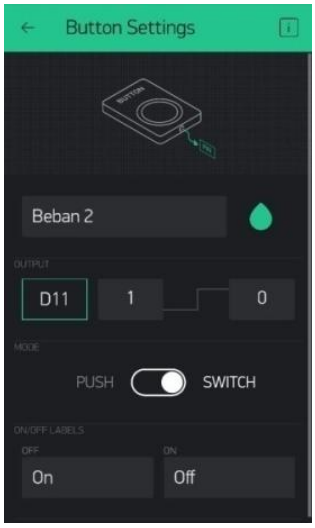
Virtual 3 yang ditunjukkan pada gambar 4.38 berfungsi sebagai penampil dari hasil pengiriman data ke *smartphone* berupa data hasil pengukuran daya beban pada beban 3 dalam satuan *Watt*. Dalam merancang tombol 1 yang ditunjukkan pada gambar 4.39 berfungsi sebagai tombol manual yang digunakan untuk mematikan beban sesuai dengan keinginan pengguna.



Gambar 4.38 Perancangan *virtual* tampilan daya beban3 (*Watt*) di *smartphone*

Gambar 4.39 Perancangan *virtual* tombol beban1 di *smartphone*

Dalam merancang tombol 2 yang ditunjukkan pada gambar 4.40 sama seperti tombol 1 berfungsi sebagai tombol manual yang digunakan untuk mematikan beban sesuai dengan keinginan pengguna. Dalam merancang tombol 3 yang ditunjukkan pada gambar 4.41 sama seperti tombol 1 dan tombol 2 berfungsi sebagai tombol manual yang digunakan untuk mematikan beban sesuai dengan keinginan pengguna.



Gambar 4.40 Perancangan *virtual* tombol beban2 di *smartphone*

Gambar 4.41 Perancangan *virtual* tombol beban3 di *smartphone*

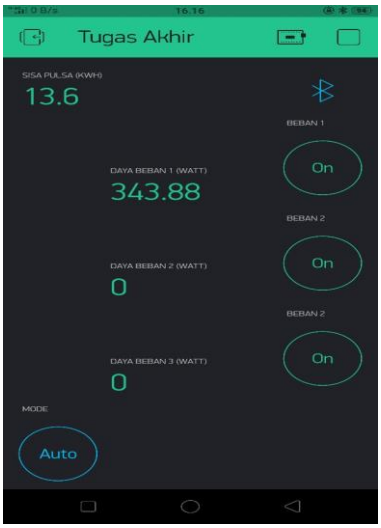
Virtual 7 ditunjukkan pada gambar 4.42 berfungsi sebagai tombol mode manual dan otomatis, sistem manual ini digunakan untuk mematikan beban sesuai dengan keinginan pengguna. Sedangkan sistem otomatis digunakan untuk mematikan beban dengan daya besar sesuai dengan *input*an maksimal daya. Pada gambar 4.43 berfungsi sebagai tombol atau fitur untuk mengkoneksikan *bluetooth* dengan *smartphone.*

Gambar 4.42 Perancangan *virtual* tombol otomatis dan manual di *smartphone*

Gambar 4.43 Perancangan *virtual* untuk koneksi *bluetooth HC-06* ke *smartphone*

4.3.2 Hasil Pembuatan *Software* dari Aplikasi *Blynk*

Setelah proses perancangan dilakukan maka didapatkanlah hasil dari perancangan *software* tersebut. *Software smartphone* ini, menggunakan *BLYNK apps* yang merupakan *IoT web server*. *Software* ini digunakan untuk mengirimkan data dari modul *Bluetooth HC-06* ke *smartphone. Software* inilah juga yang digunakan untuk mengontrol serta memonitoring pemakaian listrik. Hasil dari perancangan *software* ini ditunjukkan pada gambar 4.44.



Gambar 4.44 Hasil perancangan *software* pada aplikasi *blynk*

4.4 *Software* Pemograman

Dalam proses pemograman pada *kWh* meter prabayar ini, menggunakan *software Arduino IDE*. Sebelum melakukan pemograman terhadap *kWh* meter ini, maka diperlukan diagram alir atau disebut *flowchart*. *Flowchart* ini berfungsi untuk membantu para *programmer* membuat langkah-langkah dalam menyelesaikan program. Berikut ini adalah tahapan dalam pembuatan *flowchart*, yaitu :

1. Pendefinisian masalah
2. Pepresentasi kerja program
3. Penemuan instruksi-instruksi yang benar

4. Penulisan program

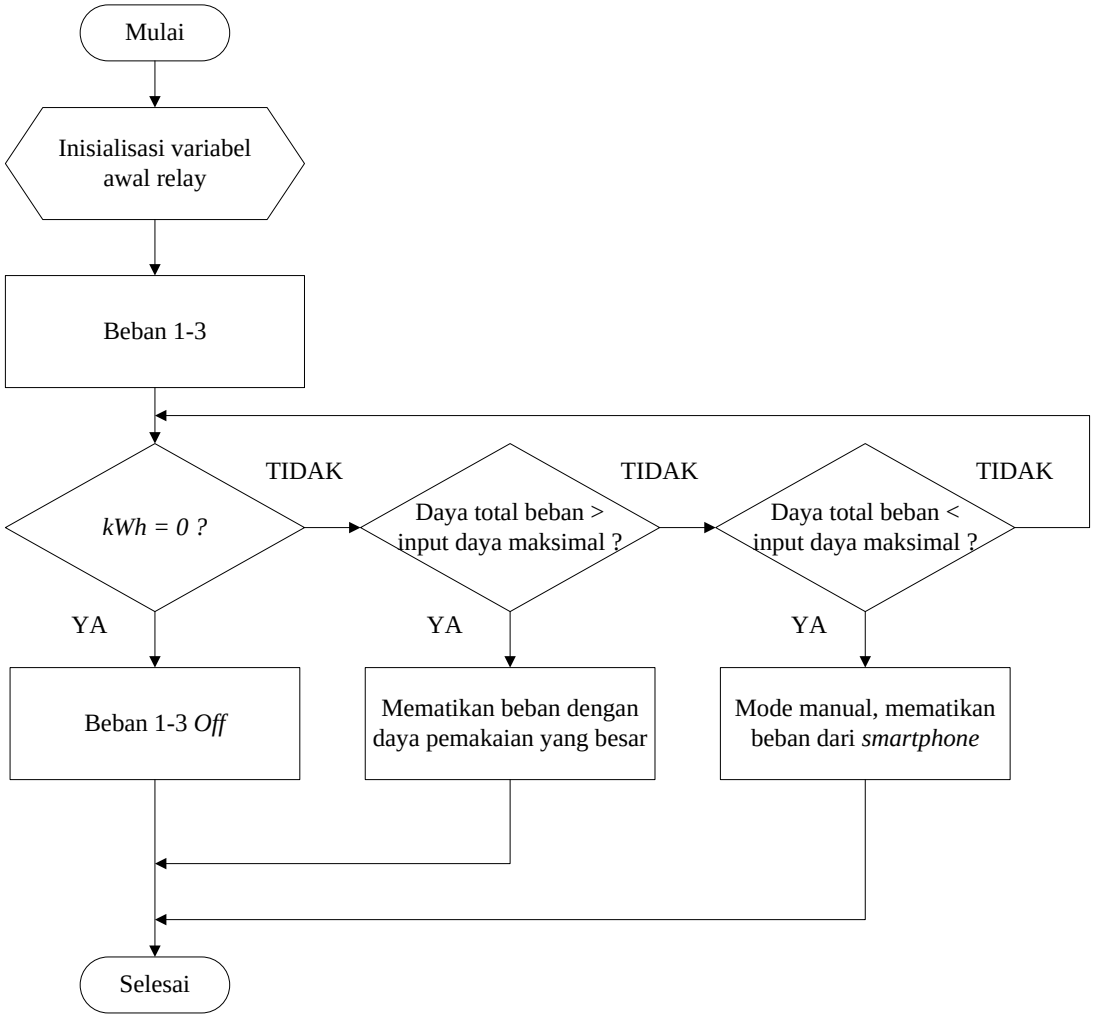
Gambar *flowchart* yang disusun berfungsi untuk membuat program keseluruhan pada alat prototipe *monitoring* pemakaian listrik *via android*, *flowchart* ini berfungsi sebagai penunjukkan alur atau tahapan-tahapan dari pemograman pada prototipe *monitoring* pemakaian listrik *via android*. Terdapat 3 *flowchart* yang dijelaskan dalam makalah ini, antara lain :

1. *Flowchart* program relay, relay ini digunakan sebagai pengontrolan beban dalam sistem otomatis maupun sistem manual.
2. *Flowchart* program sensor PZEM-004T, digunakan untuk membaca pengukuran arus beban, tegangan beban yang ditampilkan pada *LCD* dan digunakkan sebagai pembaca pengukuran daya beban yang ditampilkan pada *android*.
3. *Flowchart* program keseluruhan, digunakan untuk pembuatan program keseluruhan dari prototipe *monitoring* pemakaian listrik *via android*, program keseluruhan ini meliputi program relay, program PZEM-004T, program EEPROM, program buetooth dan program *android*.

4.4.1 *Flowchart* Program *Relay*

Flowchart ini menjelaskan sistem kerja dari pengontrolan beban yang menggunakan *relay*. Pengontrolan pada *relay* ini dibagi menjadi dua sistem. Sistem pertama adalah sistem otomatis yang dimana akan bekerja berdasarkan *input*an daya maksimal terhadap daya total dari beban yang aktif. Apabila daya total yang terpakai melebihi *input*an daya maksimal yang ada maka secara otomatis relay akan memutuskan aliran listrik ke stop kontak yang memiliki beban terbesar diantara beban lainya. Sistem kedua adalah sistem manual yang dimana akan bekerja dengan cara dikontrol menggunakan *smartphone*. Pada sistem ini *input*an daya maksimal harus melebihi daya total agar sistem otomatis tidak aktif, serta penggua harus menggati *mode* yang ada pada android dari *mode* otomatis ke *mode* manual lalu

sistem ini akan aktif dan pengguna hanya perlu menekan tombol *on/off* pada *android*. Gambar 4.45 ini akan menunjukkan *flowchart* pemograman untuk *relay*.

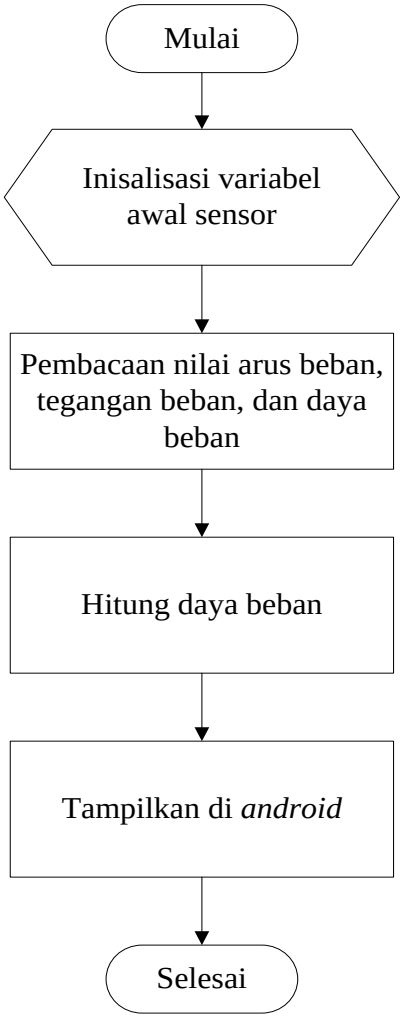


Gambar 4.45 *Flowchart* program *relay*

4.4.2 *Flowchart* Program Sensor PZEM-004T

Flowchart ini menjelaskan sistem kerja dari pembacaan sensor. Pembacaan sensor ini terdiri dari pembacaan nilai arus dari beban, nilai tegangan dari beban yang ditampilkan pada *LCD* sedangkan nilai daya dari beban ditampilkan pada *android*.

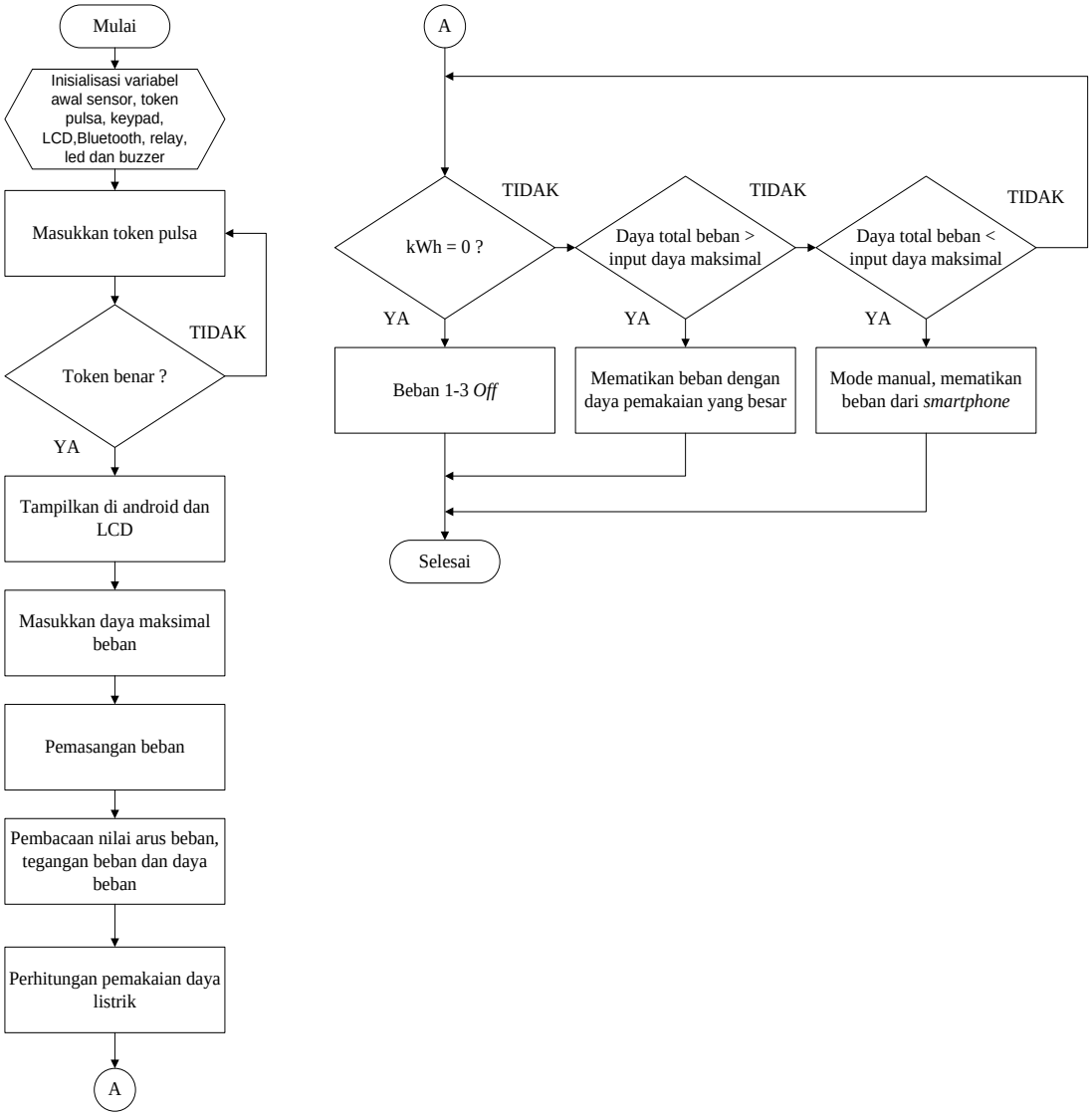
Nilai pembacaan arus dari beban, nilai pembacaan tegangan dari beban, dan nilai pembacaan daya dari beban yang terukur dari sensor PZEM-004T berupa nilai desimal dengan tipe data *float*. Sedangkan untuk mendapatkan nilai dari cos phi, maka dibuatlah rumus dari cos phi= P/S. Untuk P adalah berupa daya yang ada di beban. Sedangkan S merupakan daya semu yang memiliki rumus yaitu, S=V*I. Untuk V ini merupakan tegangan dari beban yang terukur. sedangkan I merupakan arus dari beban yang terukur. Gambar 4.46 menunjukkan flowchart pemograman sensor PZEM-004T.



Gambar 4.46 *Flowchart* program sensor PZEM-004T

4.4.3 *Flowchart* Program Keseluruhan Prototipe *Monitoring* Pemakaian Listrik *Via Android*

Flowchart ini akan menjelaskan sistem kerja dari keseluruhan prototipe *monitoring* pemakaian listrik *via android* meliputi pengontrolan beban, pengukuran arus, tegangan beserta daya beban, pengiriman dataditunjukkan pada gambar 4.45.



Gambar 4.47 *Flowchart* program keseluruhan prototipe *monitoring* pemakaian listrik *via android.*

Gambar 4.47 merupakan *flowchart* yang menjelaskan tentang program pembacaan pengukuran arus, tegangan, daya serta program peng*input*an token pulsa. Program perhitungan penggunaan daya listrik, program otomatis dan manual untuk pengontrolan beban serta program pengiriman data ke *smartphone*.

4.5 Pengujian Keseluruhan

Dalam proses pengujian dilakukan beberapa tahap untuk mencari nilai yang sesuai dengan hasil pengukuran yang dilakukan. Dibawah ini tahapan yang dilakukan dalam pengujian pada komponen yang digunakan, antara lain :

4.5.1 Proses Pengujian Keseluruhan Pada Prototipe *Monitoring* Pemakaian Listrik *Via Android*

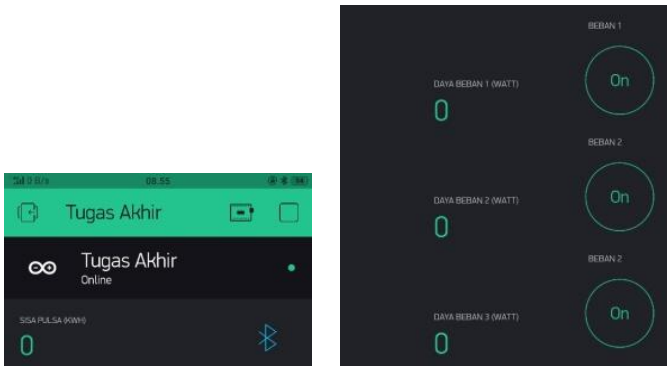
Dalam pengujian alat ini terdapat beberapa tahapan yang dilakukan penulis dalam menguji alat ini. Yang dilakukan dalam menguji alat ini adalah menguji dari pengiriman data dari *arduino* ke *smartphone* yang berupa sisa pulsa *kWh*, daya beban. Penulis menguji pengontrolan sistem otomatis dengan memasukkan nilai daya maksimal dalam penggunaan daya total beban. Dan penulis menguji pengontrolan sistem manual dengan melakukan menekan tombol *on/off* untuk mematikan atau menghidupkan beban.

Tahapan dalam pengujian alat ini antara lain :

a. Menghubungkan aplikasi *BLYNK* dengan *bluetooth HC-06*
b. Mengisi pulsa *kWh* yang di*input* pada *keypad*
c. *Input* daya maksimal yang diinginkan sebagai pembatas pemakaian daya beban
d. Melakukan percobaan dengan memasang beban cas laptop pada beban 1
e. Melakukan percobaan dengan memasang beban kipas angin pada beban 2
f. Melakukan percobaan dengan memasang beban setrika pada beban 3

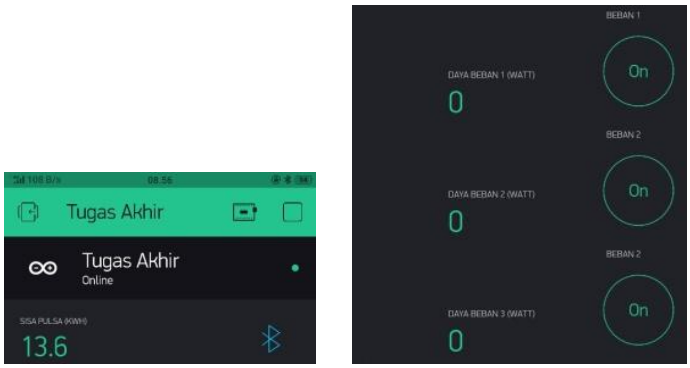
g. Alat akan menjalankan sistem otomatis apabila *input* daya maksimal lebih kecil dari total daya beban
h. Alat akan menjalankan sistem manual apabila *input* daya aksimal lebih besar dari total daya beban

a. Berikut ini tahapan dalam pengujian alat ini, yaitu tahap pertama menghubungkan aplikasi *BLYNK* dengan *bluetooth HC-06*. Gambar 4.48 menunjukkan ketika *bluetooth HC-06* dan aplikasi *BLYNK* terhubung.



Gambar 4.48 Tampilan awal di *smartphone*

b. Setelah *kWh* meter telah aktif maka pengguna harus mengisi pulsa *kWh* yang di*input* pada *keypad*. Tampilan *kWh* saat memiliki pulsa pada gambar 4.49.

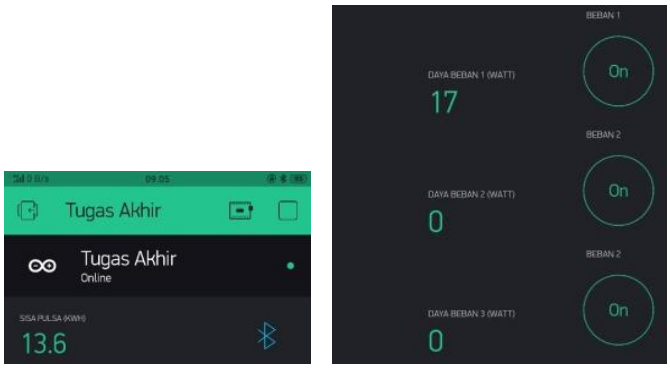


Gambar 4.49 Tampilan setelah pulsa *kWh* di*input*kan

c. Ketika *kWh* meter telah terisi pulsa maka pengguna harus mengisi *input* daya maksimal yang diinginkan sebagai pembatas pemakaian daya beban yang ditunjukkan pada gambar 4.50. Menunjukkan bahwa pengguna mengisi pulsa pada *kWh* meter sebesar 13.600 *kWh* dengan nominal uang yang di*input* pada *keypad* sebesar 20.000 dan memasukan *input* daya maksimal sebesar 300 *watt.*

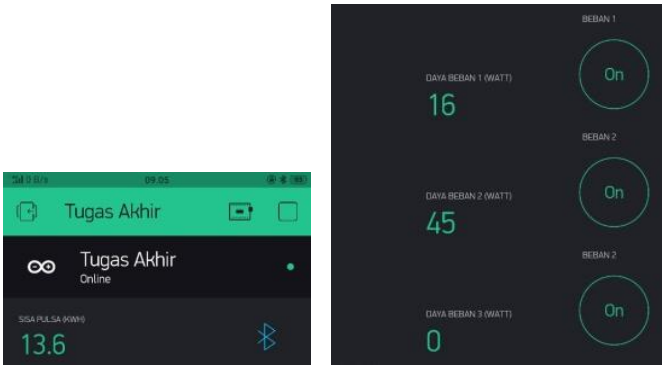
Gambar 4.50 Tampilan setelah daya maksimal di*input*kan

d. Dalam pengujian alat ini adalah setelah melakukan pengisian pulsa pada *kWh* meter dan peng*input*an daya maksimal sesuai kebutuhan dari pengguna sebagai pembatas pemakaian daya beban maka *kWh* meter sudah siap digunakan oleh pengguna. Pada proses pertama dalam pengujian pengguna melakukaan percobaan dengan memasang beban cas laptop pada beban 1. Kemudian hasil pengukuran beban akan ditampilkan ke *android* dengan daya yang satuannya berupa *Watt*. Tampilan beban cas laptop pada *android* dengan daya sebesar 17 *watt* ditunjukkan pada gambar 4.51.



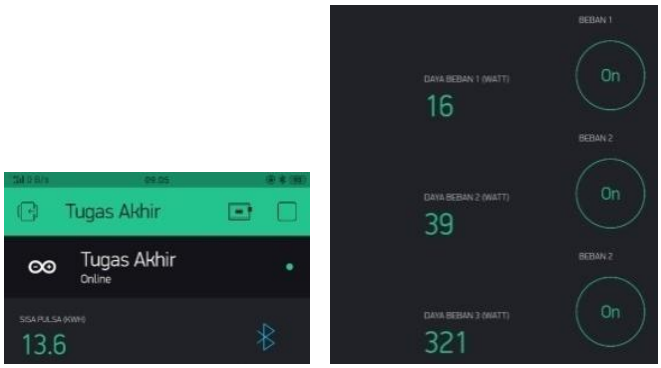
Gambar 4.51 Tampilan pengukuran dengan beban cas laptop

e. Dilakukan proses kedua dalam pengujian pengguna melakukaan percobaan dengan memasang beban kipas angin pada beban 2. Kemudian hasil pengukuran beban akan ditampilkan ke *android.* Tampilan beban kipas angin pada *android* dengan daya 45 *watt* ditunjukkan pada gambar 4.52.



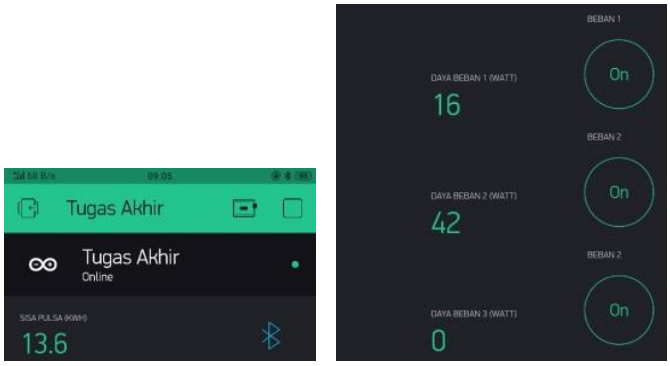
Gambar 4.52 Tampilan pengukuran dengan beban kipas angin

f. Dalam pengujian alat prototipe *monitoring* pemakaian listrik *via android* sama seperti tahap keempat dan tahap kelima pengguna memasang beban setrika pada beban 3. Dan pada tahap ini alat akan memproses atau menjalankan proses otomatis atau manual. Jika input daya maksimal lebih kecil dari total daya beban maka beban dengan daya yang besar akan dimatikan. Tampilan beban setrika pada *android* dengan daya 321 *watt* ditunjukkan pada gambar 4.53.



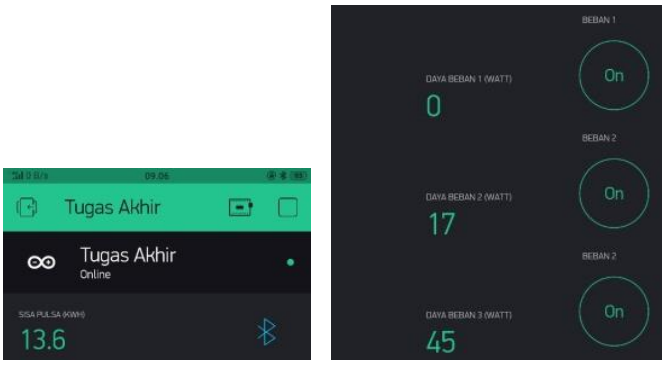
Gambar 4.53 Tampilan pengukuran dengan beban setrika

g. Sistem otomatis ini akan bekerja apabila terdapat pulsa *kWh* dan daya maksimal yang dimasukkan kurang dari jumlah daya beban dari 3 beban yang terukur maka alat ini akan otomatis memilih dan mematikan beban dengan daya yang besar. Tampilan sistem otomatis ini ditunjukkan pada gambar 4.54.



Gambar 4.54 Tampilan sistem otomatis aktif (beban 3 *off*)

h. Pengguna melakukan percobaan dan pembuktian pada stopkontak beban yang lainnya. Sistem otomatis ini akan bekerja jika beban dengan daya yang besar telah dimatikan atau dinonaktifkan pada stopkontak pertama dan pengguna memasang beban pada stopkontak yang lain maka stopkontak yang baru dipasang dengan daya yang besar akan dimatikan. Dan stopkontak pertama aktif kembali. Tampilan sistem otomatis ini ditunjukkan pada gambar 4.55.

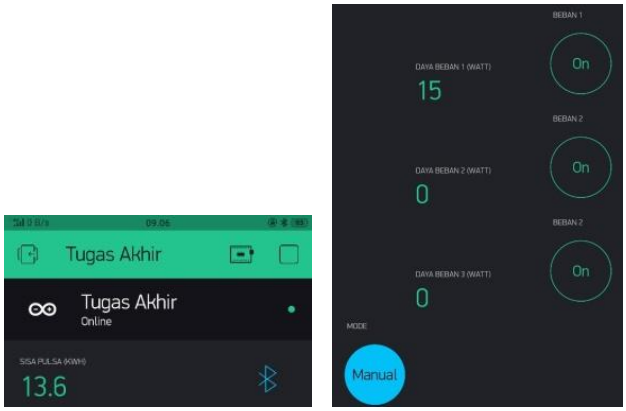


Gambar 4.55 Tampilan sistem otomatis aktif (beban 1 *off*)

i. Pada percobaan ini pengguna akan menjalankan sistem manual pada alat ini. Jadi sistem manual ini, akan aktif apabila diaktifkan oleh pengguna, terdapat pulsa *kWh*, dan memasukkan daya maksimal. Pada percobaan ini pengguna mengganti daya maksimal sebesar 400 *Watt* Tampilan daya maksimal diinputkan ditunjukkan pada gambar 4.56.

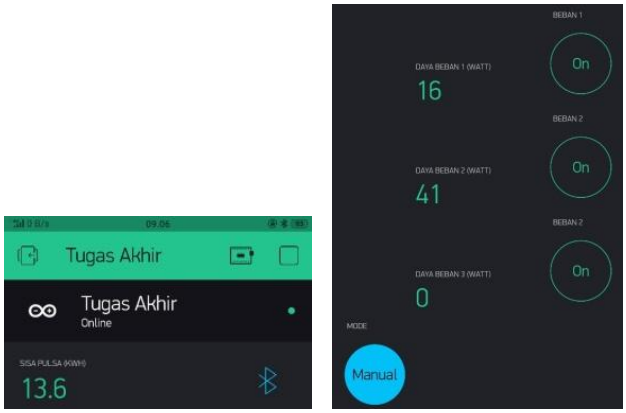
Gambar 4.56 Tampilan setelah daya maksimal di*input*kan

j. Setelah melakukan pengisian pulsa pada *kWh* meter dan peng*input*an daya maksimal sesuai kebutuhan dari pengguna sebagai pembatas pemakaian daya beban maka *kWh* meter sudah siap digunakan oleh pengguna. Proses pertama pengguna harus menekan tombol manual utuk mengaktifkan mode manual. Kemudian melakukaan percobaan dengan memasang beban cas laptop pada beban 1. Tampilan beban cas laptop yang muncul pada *android* dengan daya yang terukur sebesar 15 *watt* ditunjukkan pada gambar 4.57.



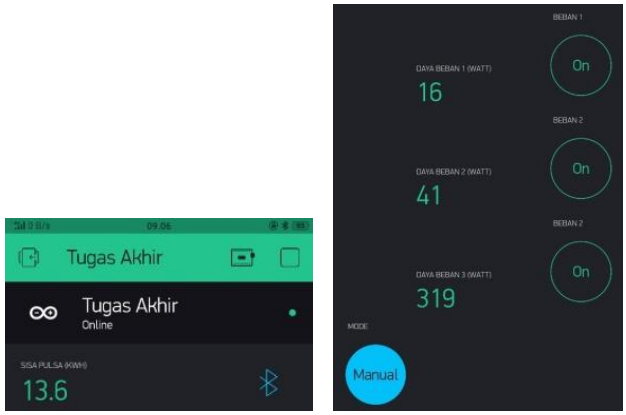
Gambar 4.57 Tampilan pengukuran dengan beban cas laptop

k. Dilakukan proses kedua dalam pengujian pengguna melakukaan percobaan dengan memasang beban kipas angin pada beban 2. Kemudian hasil pengukuran beban akan ditampilkan ke *android.* Tampilan beban kipas angin pada *android* dengan daya 41 *watt* ditunjukkan pada gambar 4.58.



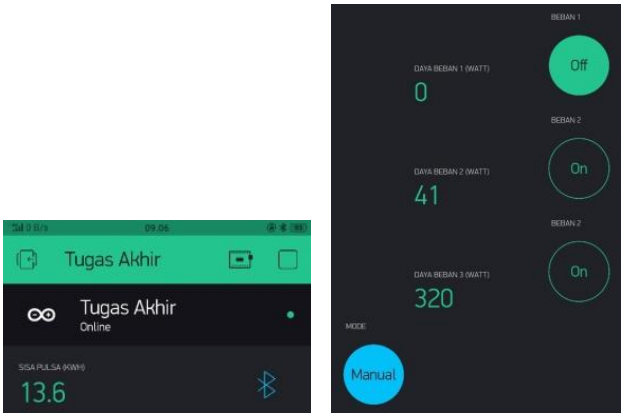
Gambar 4.58 Tampilan pengukuran dengan beban kipas angin

l. Pada alat prototipe *monitoring* pemakaian listrik *via android* sama seperti tahap kesebelas dan tahap kesepuluh pengguna memasang beban setrika pada beban 3. Tampilan beban setrika pada *android* dengan daya 319 *watt* ditunjukkan pada gambar 4.59.



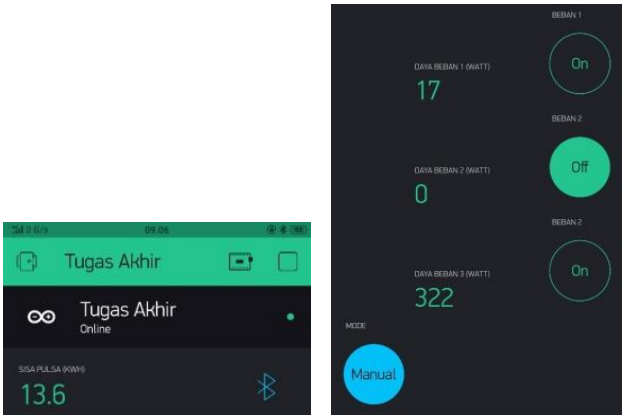
Gambar 4.59 Tampilan pengukuran dengan beban setrika

m. Sistem manual akan bekerja jika *input* daya maksimal melebihi jumlah daya beban dari 3 beban yang terukur. pengguna dapat mematikan beban sesuai keinginan pengguna. Percobaan pertama pengguna mematikan beban cas laptop pada beban 1. Tampilan sistem manual ditunjukkan pada gambar 4.60.



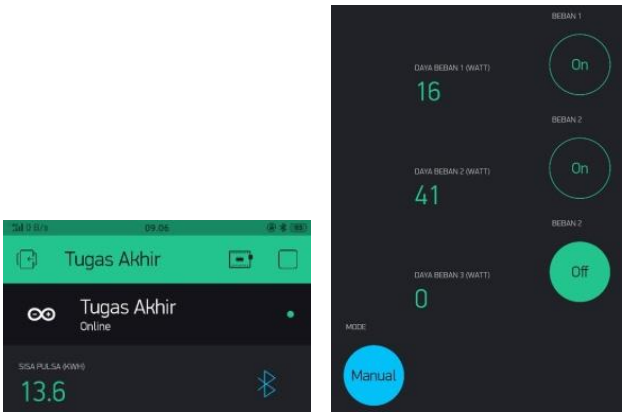
Gambar 4.60 Tampilan manual aktif (beban 1 *off*)

n. Pada alat prototipe *monitoring* pemakaian listrik *via android* pengguna melakukan percobaan dengan mematikan beban kipas angin pada beban 2. Dalam sistem manual ini input daya maksimal yang dimasukkan harus lebih besar dari daya total beban. Tampilan sistem manual ditunjukkan pada gambar 4.61.



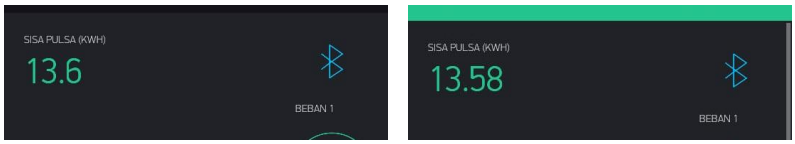
Gambar 4.61 Tampilan manual aktif (beban 2 *off*)

o. Pengguna melakukan percobaan dengan mematikan beban kipas angin pada beban 3. Tampilan sistem manual ditunjukkan pada gambar 4.62.



Gambar 4.62 Tampilan manual aktif (beban 3 *off*)

Setelah melakukan proses pengujian pulsa *kWh* selama 5 menit maka dapat dihasilkan data yang terdapat pada gambar 4.63.



Gambar 4.63 Data hasil pengujian pulsa *kWh* selama 5 menit

Setelah melakukan pengujian keseluruhan terhadap beban yang digunakan maka untuk hasil pengukuran sensor PZEM-004T terhadap tegangan harus menggunakan rumus rata-rata. Data tegangan yang digunakan untuk perhitungan rata-rata adalah tabel hasil pengukuran sensor PZEM-004T dapat dilihat pada tabel 4.5. Dan rumus perhitungan rata-rata dinyatakan pada persamaan (8).

$$\text{Rata-rata} = \frac{Jumla\ h\ Data}{Banyak\ \ data} \quad \text{.....(8)}$$

Berikut ini adalah perhitungan yang dilakukan untuk mendapatkan tegangan rata-rata yang akan digunakan dalam tabel data hasil pengujian keseluruhan yang ditunjukkan pada tabel 4.5.

$$\text{Rata-rata} = \frac{Jumla\ h\ Data}{Banyak\ \ Data} = \frac{224\ V+225\ V+225\ V}{3} = 224{,}6\ \text{V}$$

Sedangkan untuk data arus yang digunakan dalam tabel hasil pengujian keseluruhan adalah tabel hasil pengukuran sensor PZEM-004T terhadap arus yang dapat dilihat pada tabel 4.4.

Tabel hasil pengujian keseluruhan dari alat prototipe *monitor*ing pemakaian listrik *via android* , ditunjukkan pada tabel 4.21.

Tabel 4.21 Data hasil pengujian keseluruhan

Beban Yang Digunakan	Hasil Pengukuran Sensor PZEM-004T		Aplikasi (Daya dan *kWhP*pulsa)			*Error* %	Cos φ
	V(V)	I(A)	P(W)	awal	akhir		
Kipas angin (42W)	224,6	0,22	41	13,6	13,58	2	0,85
Cas laptop (18W)	224,6	0,10	17	13,6	13,58	5,5	0,80
Setrika (300W)	224,6	1,55	322	13,6	13,58	7	0,86

Dari data diatas dapat dianalisa dan disimpulkan bahwa uji coba alat yang dilakukan dengan pengujian *software* sesuai dengan keinginan dan sesuai dengan fungsinya. Karena hasil didapatkan daya beban dari hasil pengukuran sensor PZEM-004T mendekati dari nilai daya beban yang sesungguhnya. Dan didapatkan hasil persentase error yang bisa dikatagorikan kecil. Sehingga alat ini layak untuk digunakan dalam pengukuran daya beban. Untuk menentukan persentase *error* daya

beban dengan daya yang ditampilkan pada aplikasi, dapat dilakukan dengan perhitungan yang menggunakan rumus *error* sebagai berikut :

$$\textit{error} \text{ daya pada aplikasi} = \left|\frac{nilai\ \ daya\ \ awal - nilai\ \ daya\ \ ukur}{nilai\ \ daya\ \ awal}\right| X100\% \text{...(9)}$$

Dibawah ini merupakan perhitungan serta rumus persentase *error* pengukuran pada aplikasi :

- Hasil ukur daya kipas angin pada aplikasi

  error ukur daya setrika $= \left|\frac{42W - 41\ W}{42W}\right| X100\% = 2\ \%$

- Hasil ukur daya cas laptop pada aplikasi

  error ukur kipas angin $= \left|\frac{18W - 17W}{18W}\right| X100\% = 5{,}5\ \%$

- Hasil ukur daya setrika pada aplikasi

  error ukur kipas angin $= \left|\frac{300W - 322W}{300W}\right| X100\% = 7{,}3\ \%$

Untuk menghitung daya dalam satuan *Watt*, maka diperlukan faktor daya ataupun yang dikenal dengan cos phi, dalam perhitungan daya harus menggunakan cos phi karena jika hanya menggunakan rumus daya semu daya belum dikatakan satuan *Watt*. Rumus yang digunakan sebagai berikut :

$$\text{Cos phi} = P/S = P/V*I \text{.....(10)}$$

Dibawah ini merupakan perhitungan serta rumus cos phi pengukuran pada aplikasi :

- Cos phi untuk beban kipas angin

  Cos phi = P/S = P/V*I = 42 W/224,6V*0.22A = 0.85

- Cos phi untuk beban cas laptop

  Cos phi = P/S = P/V*I = 18 W/ 224.6V*0.10 A= 0,80

- Cos phi untuk beban setrika

  Cos phi = P/S = P/V*I = 300 W/ 224.6V*1.55 A= 0,86

Setelah proses pengambilan data dengan hasil dari percobaan sensor PZEM-004T terhadap daya yang terukur di aplikasi, dapat dianalisa dari data tabel 4.20 diatas bahwa modul PZEM-004T memiliki sensitivitas yang cukup tinggi, karena hasil yang didapatkan dari tampilan *android* mendekati dengan beban sesungguhnya. Sehingga alat ini layak digunakan untuk pengukuran daya beban. Untuk perhitungan cos phi, rumus yang digunakan adalah cos phi = P/S (10). Untuk P merupakan daya beban sesungguhnya. Sedangkan S adalah hasil kali dari V dan I yang terukur oleh sensor PZEM-004T.

4.6 Analisa Data

Pada proses ini, analisa data akan dilakukan jika alat tidak berhasil. Alat akan dianalisa dalam masalah pada alat tersebut meliputi *software, hardware,* rangkaian kontrol dan program. Kemudian perbaikan dilakukan berdasarkan analisa yang dialkukan. Dalam proses ini, terdapat 2 tahapan perbaikan yaitu :

1. Perbaikan kecil

   Perbaikan kecil adalah suatu tahapan pertama pada analisa data yang dimana alat tersebut berfungsi namun alat membutuhkan sedikit perbaikan.
2. Perbaikan besar adalah suatu tahapan kedua pada analisa data dimana alat tersebut tidak berfungsi atau gagal.

BAB V
PENUTUP

5.1 Kesimpulan

Setelah melakukan tahapan perancangan alat, pembuatan alat dan dilanjutkan dengan tahapan pengujian alat maka dapat disimpulkan sebagai berikut :

1. Alat prototipe *monitoring* pemakaian listrik *via android* ini dapat me*monitoring* arus, tegangan, daya dan cos φ dengan *error* pembacaan sensor sebesar 1%.
2. Alat prototipe *monitoring* pemakaian listrik *via android* ini dapat mengirim data ke *android* yang berupa sisa pulsa *kWh* dan daya pada setiap beban dengan maksimal jarak 20 meter yang memiliki persentase *error* 5%.

5.2 Saran

Dari proyek akhir yang telah dilakukan ini masih terdapat beberapa kekurangan dan dimungkinkan untuk melanjutkan perkembangan lebih lanjut. Adapun saran yang dapat penulis sampaikan sebagai berikut :

1. Alat ini menggunakan *bluetooth* HC-06 dalam pengiriman data ke *android*. *Bluetooth* memiliki maksimal jarak yaitu 20 meter, maka dari itu disarankan untuk mengembangkan pengiriman data ke *android* menggunakan internet yang memiliki jangkauan lebih luas.
2. Aplikasi yang digunakan pada alat ini adalah *blynk*. Aplikasi ini hanya dapat mengontrol satu papan *arduino*, untuk pengembangan selanjutnya pengontrolan disarankan menggunakan aplikasi *virtuino* karna dapat melakukan pengkontrolan lebih dari satu papan *arduino* dalam satu waktu yang bersamaan.

DAFTAR PUSTAKA

[1] Deonugraha Pratama dan Riski Novriandy, " Alat Penghitung Pemakaian Listrik *Via Sms*", Laporan Tugas Akhir, Politeknik Manufaktur Negeri Bangka Belitung, Sungailiat, 2014. Diakses pada tanggal 22 April 2018.

[2] Teori daya http://jurnal.itats.ac.id/wp-content/uploads/2013/06/4.-RINY-FINAL-hal-24-32.pdf. Diakses pada tanggal 23 Juli 2018.

[3] Barry Alkahfi dan Gerian Dwiki Sakti SP, "*Monitoring Control* Motor Induksi 1 *Phase* Menggunakan *Smartphone*", Laporan Tugas Akhir, Politeknik Manufaktur Negeri Bangka Belitung, Sungailiat, 2017. Diakses pada tanggal 18 April 2018.

[4] Sensor PZEM-004T http://innovatorsguru.com/ac-digital multifunction-meter-using-pzem-004t/. Diakses pada tanggal 17 Juli 2018.

[5] Ihda Aulia dan Larasati, "Rancang Bangun Kandang Pintar Untuk Hewan Peliharaan (Kucing)", Laporan Tugas Akhir, Politeknik Manufaktur Negeri Bangka Belitung, Sungailiat, 2017. Diakses pada tanggal 22 Maret 2018.

[6] Syaiful Effendi, "Sistem Pengkuran Dan Pengiriman Data Arus Listrik Menggunakan *Power Line Carrier* (PLC)", Proposal Tugas Akhir, Politeknik Negeri Padang, Padang, 2016. Diakses pada tanggal 25 April 2018.

[7] Muhammad Lutfi Budiansyah, " Rancang Bangun Sistem Penyiram Tanaman Otomatis Berbasis Arduino Di Rumah Kaca Pusat Penelitian Bioteknelogi-Lipi", Laporan Praktek Kerja Lapangan, Institut Pertanian Bogor, Bogor, 2014. Diakses pada tanggal 29 Juni 2018.

[8] Sulihati dan Andriyani, "Aplikasi Akademik *Online* Berbasis *Mobile Android*", Universitas Tama Jagakarsa, Jakarta Selatan. Diakses pada tanggal 10 Agustus 2018.

[9] Hilma HR Jufri, "Rancang Bangun Alat Ukur Daya Arus Bolak Balik Berbasis Mikrokontroler ATMega8535", Universitas Sumatera Utara, Medan. Diakses pada tanggal 10 Agustus 2018.

[10] Teori pengertian bluetooth *HC-06* dan teori pengertian bluetooth HC-05 https://www.nyebarilmu.com/tutorial-arduino-module-bluetooth-hc-05/. Diakses pada tanggal 11 Agustus 2018.

DAFTAR RIWAYAT HIDUP

A. DATA PRIBADI

Nama	: Dwi Retno Angraini
Tempat tanggal lahir	: Curup, 12 Agustus 1997
Jenis kelamin	: Perempuan
Agama	: Islam
Status	: Belum kawin
Alamat	: Perumahan Cempaka Mas No. 12 Jalan Binjai
No.handphone	: 0823-6162-3513
E-mail	: dwipolman@gmail.com

B. DATA PENDIDIKAN

1. Pendidikan Formal
   a. Tahun 2003-2009 : SDN 33 Pangkalpinang
   b. Tahun 2009-2012 : SMPN 1 Pangkalpinang
   c. Tahun 2012-2015 : SMAN 2 Pangkalpinang
   d. Tahun 2015-2018 : D III Teknik Elektronika dan Informatika, Polman Negeri Bangka Belitung

C. PENGALAMAN KERJA

PT PADMA SOODE INDONESIA : PRODUCT DEVELOPMENT

Demikian daftar riwayat hidup ini saya dengan sebenar-benarnya dan dapat dipertanggungjawabkan.

Pangkalpinang, 22 Juli 2018

Dwi Retno Angraini

DAFTAR RIWAYAT HIDUP

A. DATA PRIBADI

Nama	: Mochamad Rivai
Tempat tanggal lahir	: Pangkalpinang, 9 September 1997
Jenis kelamin	: Laki-laki
Agama	: Islam
Status	: Belum kawin
Alamat	: Jl. Sumedang air ledeng Kota Pangkalpinang
No.handphone	: 0812-8231-6218
E-mail	: mochamadrivai09@gmail.com

B. DATA PENDIDIKAN

1. Pendidikan Formal
   a. Tahun 2003-2009 : SDN 5 Pangkalpinang
   b. Tahun 2009-2012 : SMPN 2 Pangkalpinang
   c. Tahun 2012-2015 : SMKN 2 Pangkalpinang
   d. Tahun 2015-2018 : D III Teknik Elektronika dan Informatika, Polman Negeri Bangka Belitung

C. PENGALAMAN KERJA

PT SHIBA HIDROLIK PRATAMA : Engineer

Demikian daftar riwayat hidup ini saya dengan sebenar-benarnya dan dapat dipertanggungjawabkan.

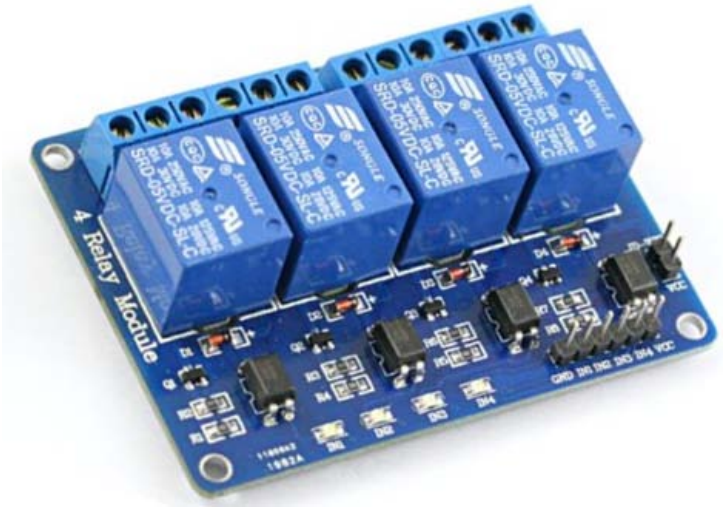
Pangkalpinang, 22 Juli 2018

Mochamad Rivai

User Guide

4 Channel 5V Optical Isolated Relay Module

This is a LOW Level 5V 4-channel relay interface board, and each channel needs a 15-20mA driver current. It can be used to control various appliances and equipment with large current. It is equipped with high-current relays that work under AC250V 10A or DC30V 10A. It has a standard interface that can be controlled directly by microcontroller. This module is optically isolated from high voltage side for safety requirement and also prevent ground loop when interface to microcontroller.



Brief Data:

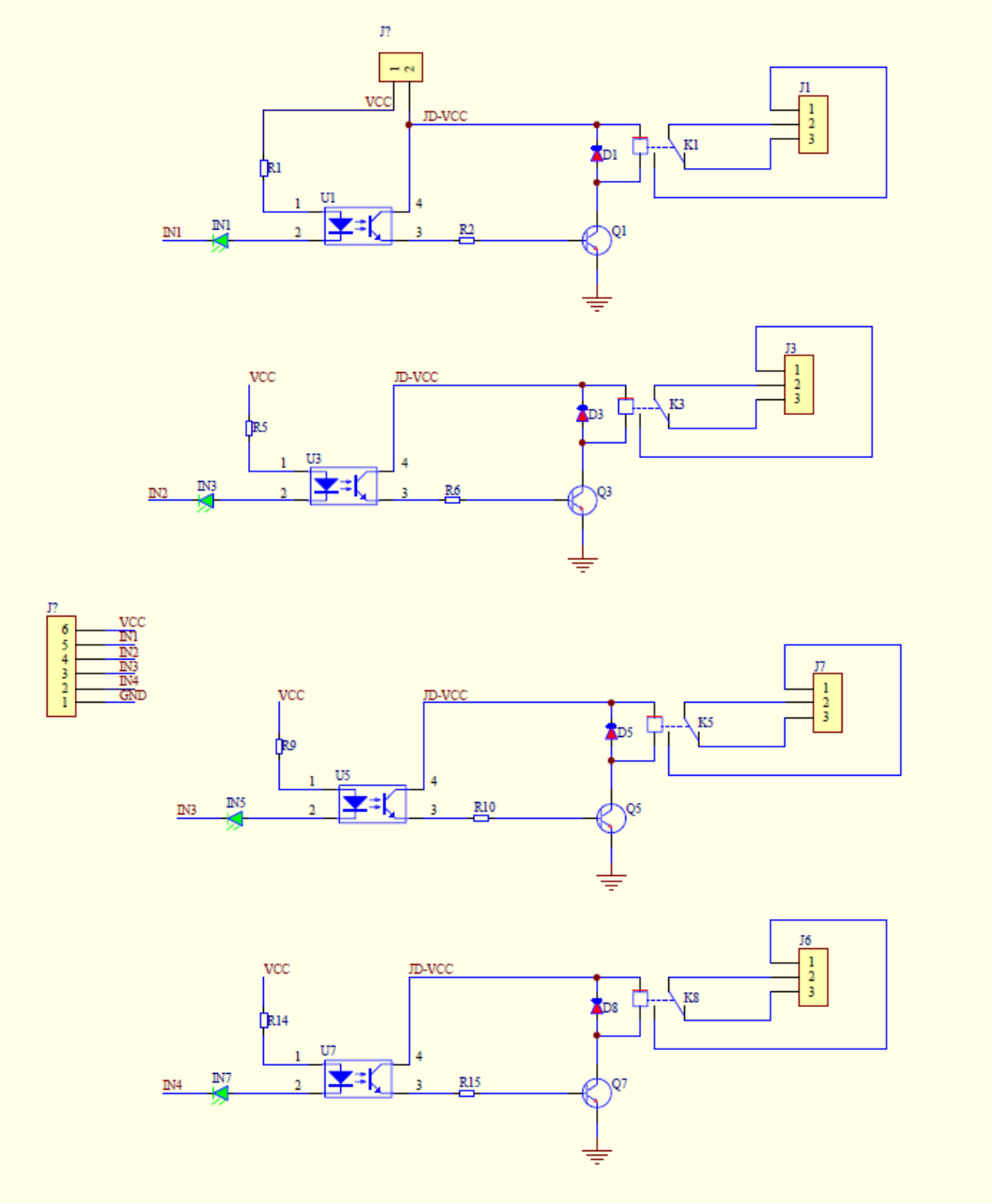
- Relay Maximum output: DC 30V/10A, AC 250V/10A.
- 4 Channel Relay Module with Opto-coupler. LOW Level Trigger expansion board, which is compatible with Arduino control board.
- Standard interface that can be controlled directly by microcontroller (8051, AVR, *PIC, DSP, ARM, ARM, MSP430, TTL logic).
- Relay of high quality low noise relays SPDT. A common terminal, a normally open, one normally closed terminal.
- Opto-Coupler isolation, for high voltage safety and prevent ground loop with microcontroller.

Schematic:

VCC and RY-VCC are also the power supply of the relay module. When you need to drive a large power load, you can take the jumper cap off and connect an extra power to RY-VCC to supply the relay; connect VCC to 5V of the MCU board to supply input signals.

NOTES: If you want complete optical isolation, connect "Vcc" to Arduino +5 volts but do NOT connect Arduino Ground. Remove the Vcc to JD-Vcc jumper. Connect a separate +5 supply to "JD-Vcc" and board Gnd. This will supply power to the transistor drivers and relay coils.

If relay isolation is enough for your application, connect Arduino +5 and Gnd, and leave Vcc to JD-Vcc jumper in place.



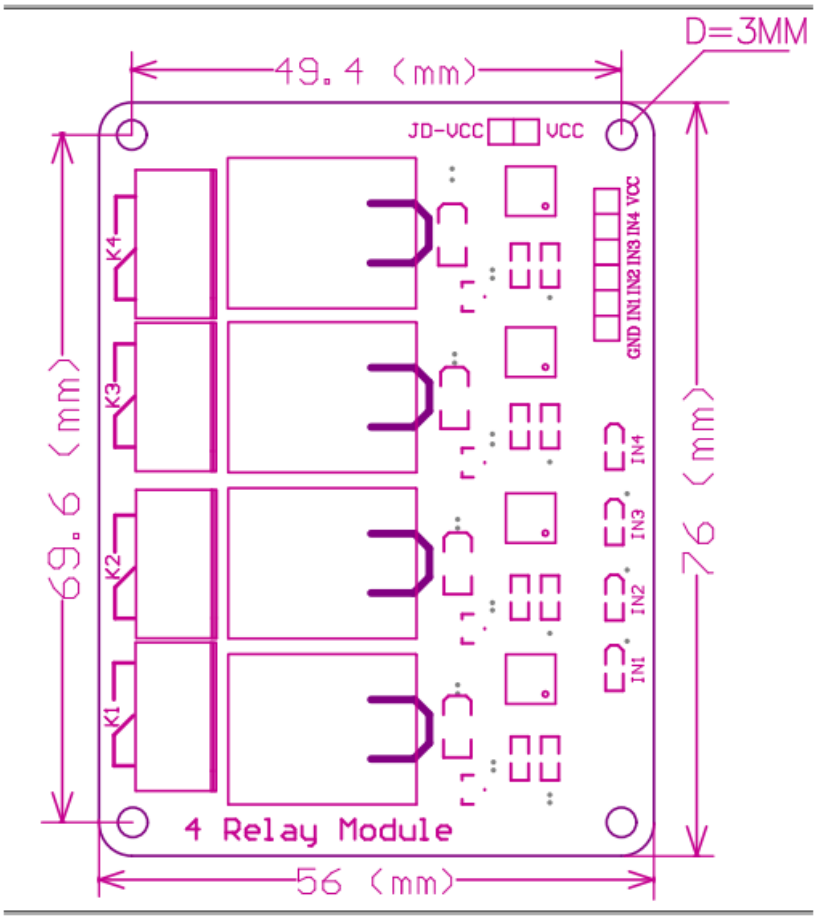
4 Channel Relay Module Schematic

It is sometimes possible to use this relay boards with 3.3V signals, if the JD-VCC (Relay Power) is provided from a +5V supply and the VCC to JD-VCC jumper is removed. That 5V relay supply could be totally isolated from the 3.3V device, or have a common ground if opto-isolation is not needed. If used with isolated 3.3V signals, VCC (To the input of the opto-isolator, next to the IN pins) should be connected to the 3.3V device's +3.3V supply.

NOTE: Some Raspberry-Pi users have found that some relays are reliable and others do not actuate sometimes. It may be necessary to change the value of R1 from 1000 ohms to something like 220 ohms, or supply +5V to the VCC connection.

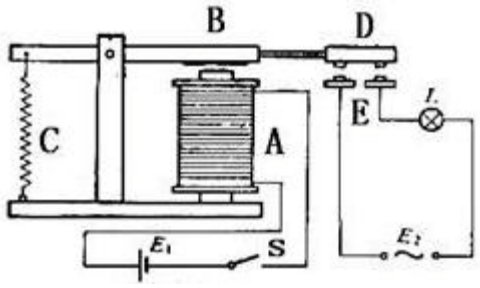
NOTE: The digital inputs from Arduino are Active LOW: The relay actuates and LED lights when the input pin is LOW, and turns off on HIGH.

Module Layout:



Operating Principle:

See the picture below: A is an electromagnet, B armature, C spring, D moving contact, and E fixed contacts. There are two fixed contacts, a normally closed one and a normally open one. When the coil is not energized, the normally open contact is the one that is off, while the normally closed one is the other that is on.



Supply voltage to the coil and some currents will pass through the coil thus generating the electromagnetic effect. So the armature overcomes the tension of the spring and is attracted to the core, thus closing the moving contact of the armature and the normally open (NO) contact or you may say releasing the former and the normally closed (NC) contact. After the coil is de-energized, the electromagnetic force disappears and the armature moves back to the original position, releasing the moving contact and normally closed contact. The closing and releasing of the contacts results in power on and off of the circuit.

Input:

VCC : Connected to positive supply voltage (supply power according to relay voltage)

GND : Connected to supply ground.

IN1: Signal triggering terminal 1 of relay module

IN2: Signal triggering terminal 2 of relay module

IN3: Signal triggering terminal 3 of relay module

IN4: Signal triggering terminal 4 of relay module

Output:

Each module of the relay has one NC (normally close), one NO (normally open) and one COM (Common) terminal. So there are 4 NC, 4 NO and 4 COM of the channel relay in total. NC stands for the normal close port contact and the state without power. NO stands for the normal open port contact and the state with power. COM means the common port. You can choose NC port or NO port according to whether power or not.

Testing Setup:

When a low level is supplied to signal terminal of the 4-channel relay, the LED at the output terminal will light up. Otherwise, it will turn off. If a periodic high and low level is supplied to the signal terminal, you can see the LED will cycle between on and off.

For Arduino：

Step 1：

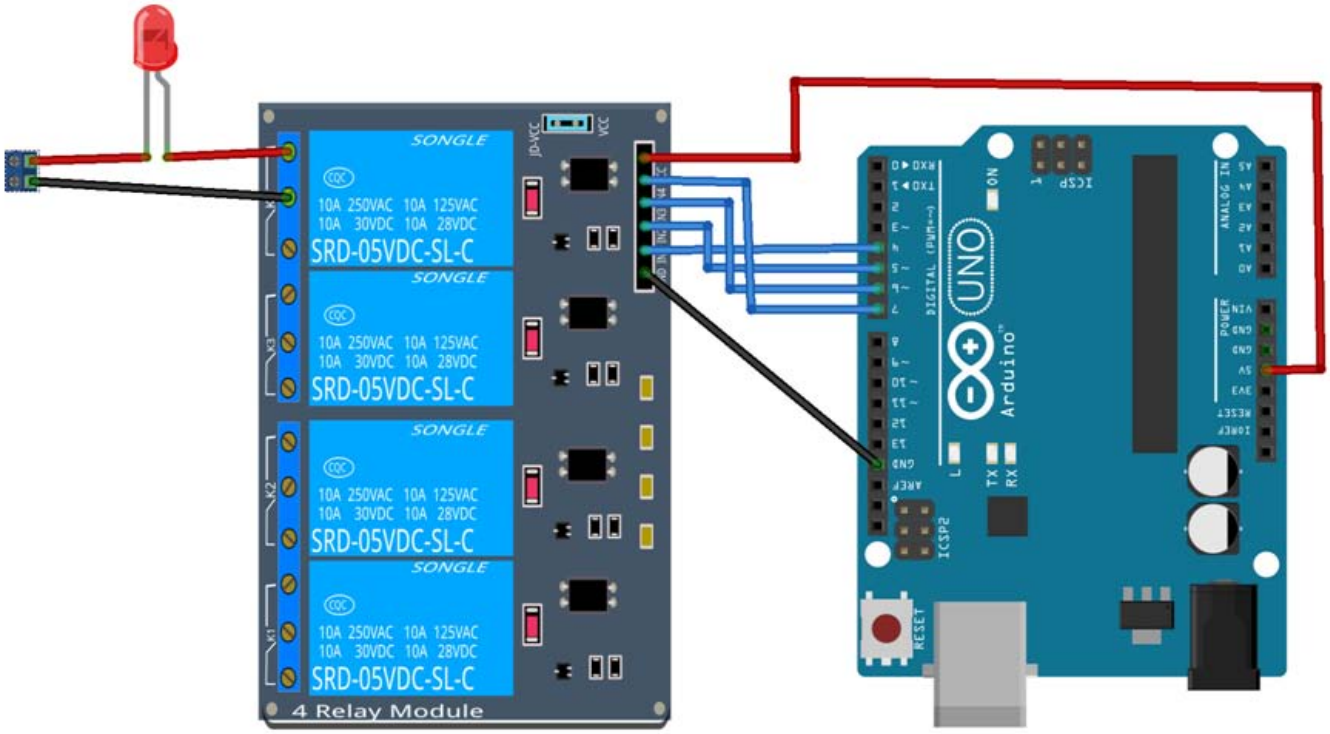
Connect the signal terminal IN1、IN2, IN3 & IN4 of 4-channel relay to digital pin 4, 5, 6, 7 of the Arduino Uno or ATMega2560 board, and connect an LED at the output terminal.

IN1> 4; IN2> 5; IN3>6; IN4>7

Step 2:

Upload the sketch "4 Channel Relay Demo " to the Arduino Uno or ATMega2560 board. Then you can see the LED cycle between on and off.

The actual figure is shown below:

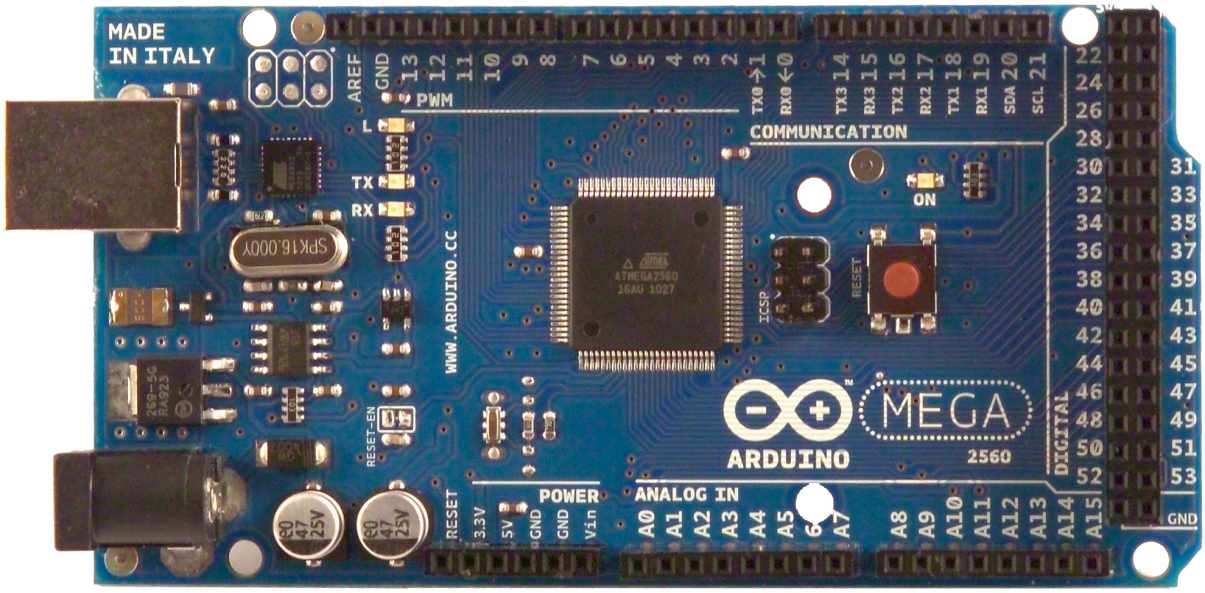


Arduino Sketch: 4 Channel Relay Demo

```
/*************************************************
   Name: 4 channel_relay
   Description: control the 4 channel relay module to ON or OFF
   Website: www.handsontec.com
   Email: techsupport@handsontec.com
**************************************************/

//the relays connect to

int RelayControl1 = 4;    // Digital Arduino Pin used to control the motor
int RelayControl2 = 5;
int RelayControl3 = 6;
int RelayControl4 = 7;


void setup()
{
  Serial.begin(9600);
  pinMode(RelayControl1, OUTPUT);
  pinMode(RelayControl2, OUTPUT);
  pinMode(RelayControl3, OUTPUT);
  pinMode(RelayControl4, OUTPUT);
}


void loop()
{

  digitalWrite(RelayControl1,HIGH);// NO1 and COM1 Connected (LED on)
  delay(1000);
```

```
  digitalWrite(RelayControl1,LOW);// NO1 and COM1 disconnected (LED off)
  delay(1000);
  digitalWrite(RelayControl2,HIGH);
  delay(1000);
  digitalWrite(RelayControl2,LOW);
  delay(1000);
  digitalWrite(RelayControl3,HIGH);
  delay(1000);
  digitalWrite(RelayControl3,LOW);
  delay(1000);
  digitalWrite(RelayControl4,HIGH);
  delay(1000);
  digitalWrite(RelayControl4,LOW);
  delay(1000);
}
```

Arduino MEGA 2560



Product Overview

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 (datasheet). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

Index

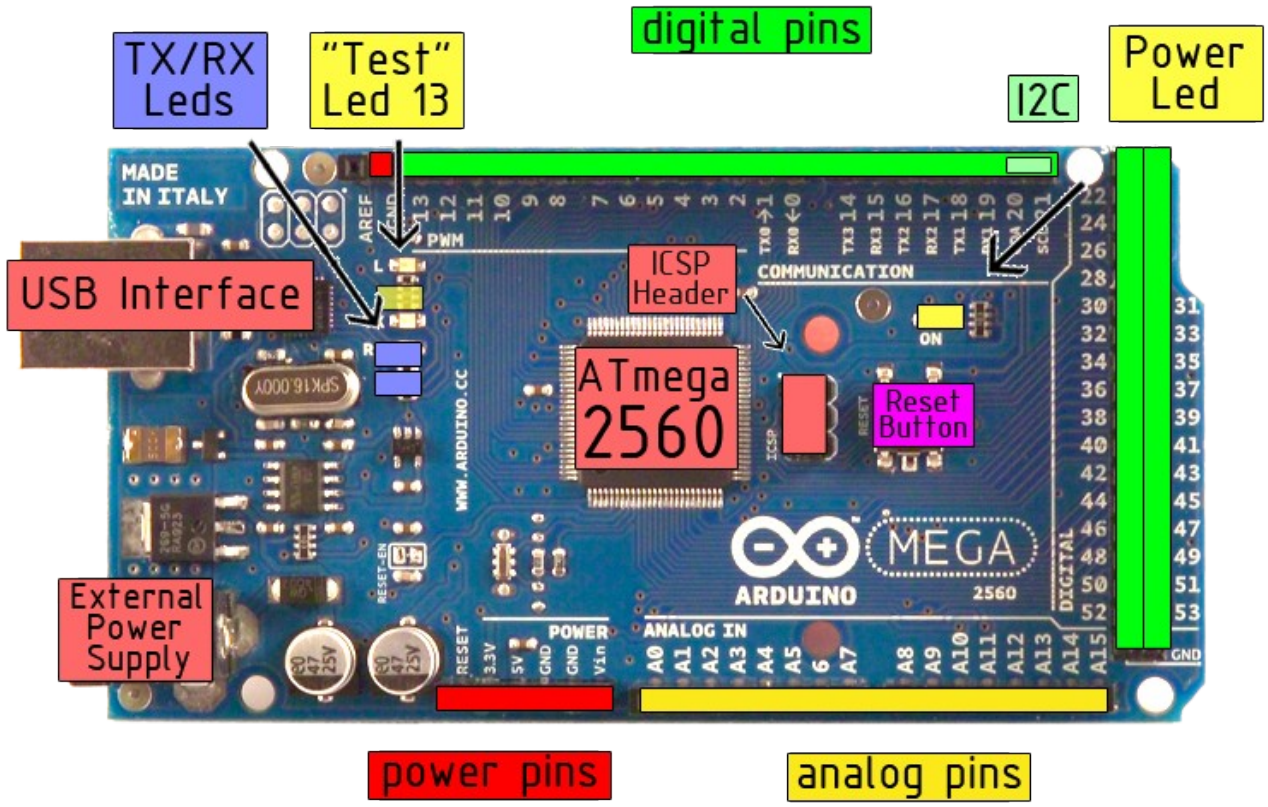
Technical Specification

EAGLE files: arduino-mega2560-reference-design.zip Schematic: arduino-mega2560-schematic.pdf

Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

the board



Power

The Arduino Mega2560 can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip .
- **External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2).** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
- **PWM: 0 to 13.** Provide 8-bit PWM output with the analogWrite() function.
- **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Duemilanove and Diecimila.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **I^2C: 20 (SDA) and 21 (SCL).** Support I^2C (TWI) communication using the Wire library (documentation on the Wiring website). Note that these pins are not in the same location as the I^2C pins on the Duemilanove.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and analogReference() function.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with analogReference().
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Communication

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A SoftwareSerial library allows for serial communication on any of the Mega's digital pins.

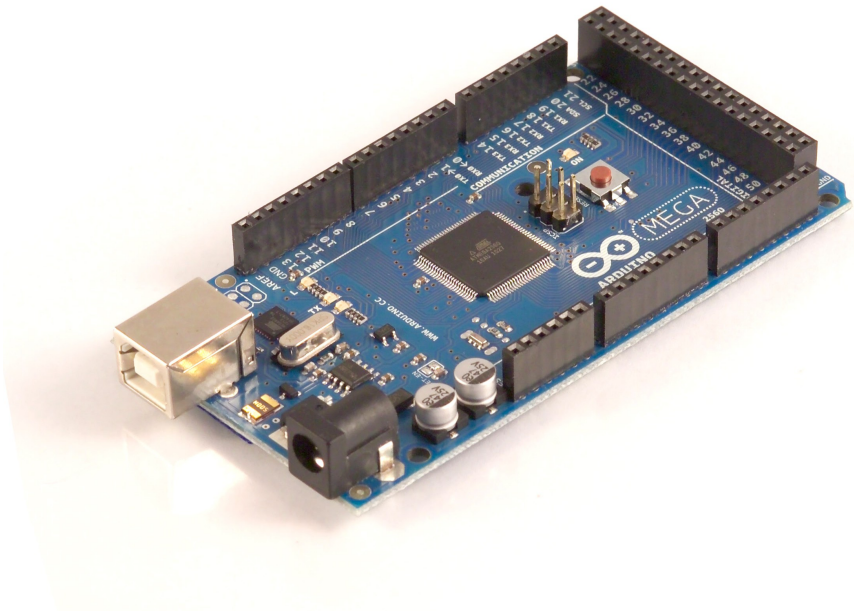
The ATmega2560 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation on the Wiring website for details. To use the SPI communication, please see the ATmega2560 datasheet.

Programming

The Arduino Mega2560 can be programmed with the Arduino software (download). For details, see the reference and tutorials.

The Atmega2560 on the Arduino Mega comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details.



Automatic (Software) Reset

Rather then requiring a physical press of the reset button before an upload, the Arduino Mega2560 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Mega2560 is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Mega2560. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Mega contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see this forum thread for details.

USB Overcurrent Protection

The Arduino Mega has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics and Shield Compatibility

The maximum length and width of the Mega PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

The Mega is designed to be compatible with most shields designed for the Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega and Duemilanove / Diecimila. **Please note that I^2C is not located on the same pins on the Mega (20 and 21) as the Duemilanove / Diecimila (analog inputs 4 and 5).**

How to use Arduino

Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the Arduino programming language (based on Wiring) and the Arduino development environment (based on Processing). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platoform program. You'll have to follow different instructions for your personal OS. Check on the Arduino site for the latest instructions. *http://arduino.cc/en/Guide/HomePage*

Linux Install

Windows Install

Mac Install

Once you have downloaded/unzipped the arduino IDE, you can Plug the Arduino to your PC via USB cable.

Blink led

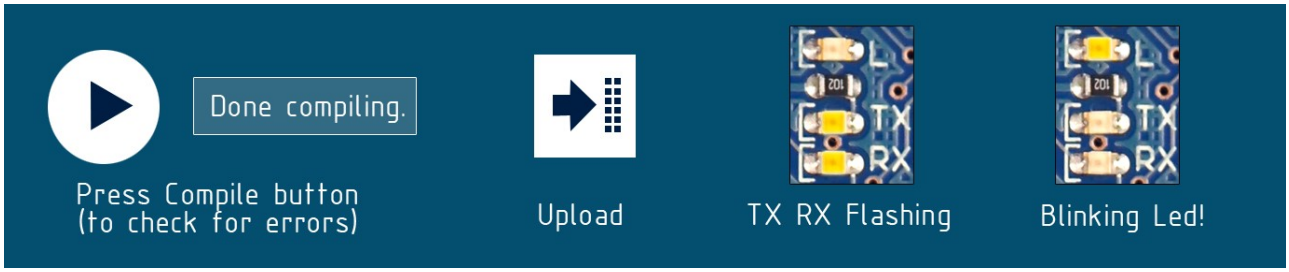
Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

File>Sketchbook>
Arduino-0017>Examples>
Digital>Blink

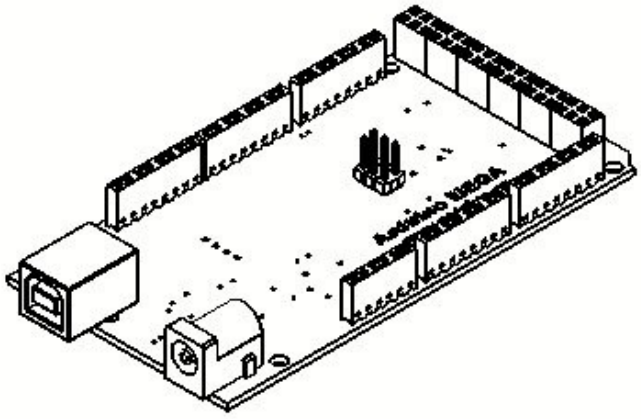
Once you have your skecth you'll see something very close to the screenshot on the right.

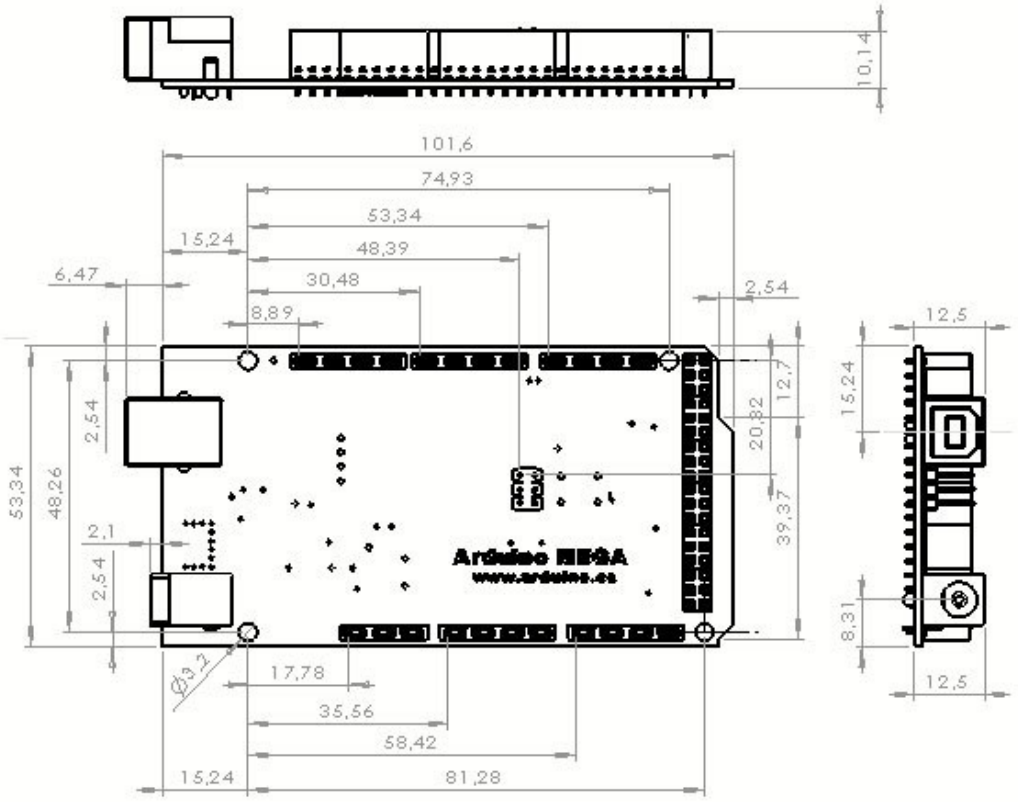
In **Tools>Board** select MEGA

Now you have to go to
Tools>SerialPort
and select the right serial port, the one arduino is attached to.



Dimensioned Drawing





Terms & Conditions

1. Warranties

1.1 The producer warrants that its products will conform to the Specifications. This warranty lasts for one (1) years from the date of the sale. The producer shall not be liable for any defects that are caused by neglect, misuse or mistreatment by the Customer, including improper installation or testing, or for any products that have been altered or modified in any way by a Customer. Moreover, The producer shall not be liable for any defects that result from Customer's design, specifications or instructions for such products. Testing and other quality control techniques are used to the extent the producer deems necessary.

1.2 If any products fail to conform to the warranty set forth above, the producer's sole liability shall be to replace such products. The producer's liability shall be limited to products that are determined by the producer not to conform to such warranty. If the producer elects to replace such products, the producer shall have a reasonable time to replacements. Replaced products shall be warranted for a new full warranty period.

1.3 EXCEPT AS SET FORTH ABOVE, PRODUCTS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS." THE PRODUCER DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE

1.4 Customer agrees that prior to using any systems that include the producer products, Customer will test such systems and the functionality of the products as used in such systems. The producer may provide technical, applications or design advice, quality characterization, reliability data or other services. Customer acknowledges and agrees that providing these services shall not expand or otherwise alter the producer's warranties, as set forth above, and no additional obligations or liabilities shall arise from the producer providing such services.

1.5 The Arduino™ products are not authorized for use in safety-critical applications where a failure of the product would reasonably be expected to cause severe personal injury or death. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Arduino™ products are neither designed nor intended for use in military or aerospace applications or environments and for automotive applications or environment. Customer acknowledges and agrees that any such use of Arduino™ products which is solely at the Customer's risk, and that Customer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

1.6 Customer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products and any use of Arduino™ products in Customer's applications, notwithstanding any applications-related information or support that may be provided by the producer.

2. Indemnification

The Customer acknowledges and agrees to defend, indemnify and hold harmless the producer from and against any and all third-party losses, damages, liabilities and expenses it incurs to the extent directly caused by: (i) an actual breach by a Customer of the representation and warranties made under this terms and conditions or (ii) the gross negligence or willful misconduct by the Customer.

3. Consequential Damages Waiver

In no event the producer shall be liable to the Customer or any third parties for any special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of the products provided hereunder, regardless of whether the producer has been advised of the possibility of such damages. This section will survive the termination of the warranty period.

4. Changes to specifications

The producer may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." The producer reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.

Enviromental Policies

The producer of Arduino™ has joined the Impatto Zero® policy of LifeGate.it. For each Arduino board produced is created / looked after half squared Km of Costa Rica's forest's.

Guangzhou HC Information Technology Co., Ltd.

Product Data Sheet

Module Data Sheet

Rev 1

1．0	2.0	2.1	2.2				
2006/6/18	2006/9/6	2010/4/22	2011/4/6				

DRAWN BY :	Ling Xin		**MODEL :** HC-06	
CHECKED BY :	Eric Huang		**Description::** BC04 has external 8M Flash and EDR module HC-06 is industrial, and compatible with civil HC-04	
APPD. BY：	Simon Mok		**REV: 2.0**	**Page :**
Former version introduction	HC-06 is the higher version of LV_BC_2.0. Linvor is the former of wavesen.			

www.wavesen.com Phone: 020-84083341 Fax: 020-84332079 QQ:1043073574
Address: Room 527, No.13, Jiangong Road, Tianhe software park, Tianhe district, Guangzhou Post: 510660
Technology consultant: support@wavesen.com Business consultant:sales@wavesen.com
Complaint and suggestion: sunbirdit@hotmail.com

Contents

1. Product's picture

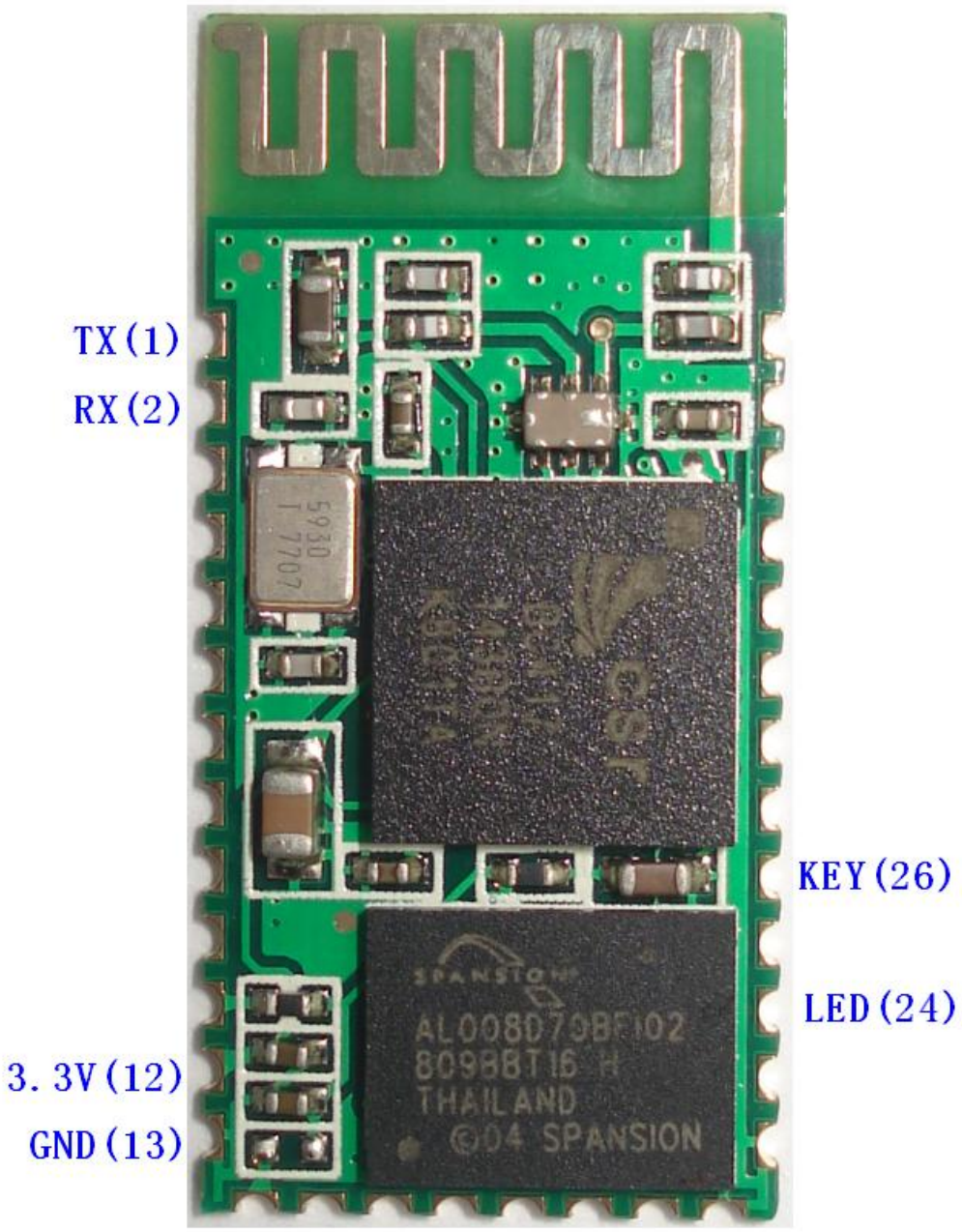


Figure 1 A Bluetooth module

www.wavesen.com Phone: 020-84083341 Fax: 020-84332079 QQ:1043073574
Address: Room 527, No.13, Jiangong Road, Tianhe software park, Tianhe district, Guangzhou Post: 510660
Technology consultant: support@wavesen.com Business consultant:sales@wavesen.com
Complaint and suggestion: sunbirdit@hotmail.com

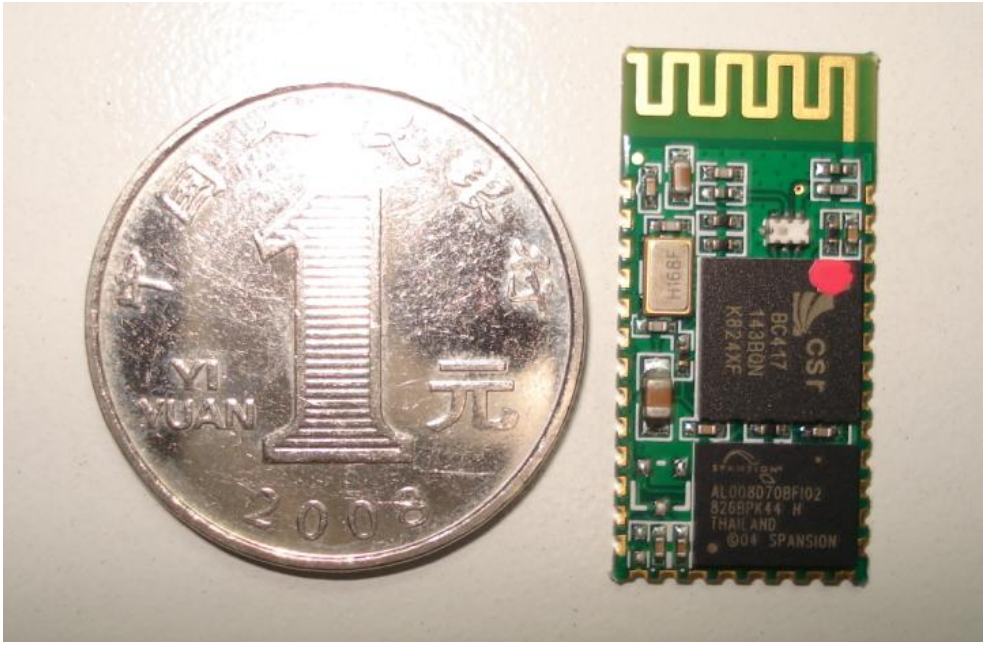


Figure 2. A Bluetooth module size

Figure 3 50 pieces chips in an anti-static blister package.

2. Feature

- Wireless transceiver
  - Sensitivity (Bit error rate) can reach -80dBm.
  - The change range of output's power: -4 - +6dBm.
- Function description (perfect Bluetooth solution)
  - Has an EDR module; and the change range of modulation depth: 2Mbps - 3Mbps.
  - Has a build-in 2.4GHz antenna; user needn't test antenna.
  - Has the external 8Mbit FLASH
  - Can work at the low voltage (3.1V~4.2V). The current in pairing is in the range of 30～40mA. The current in communication is 8mA.
  - Standard HCI Port (UART or USB)
  - USB Protocol: Full Speed USB1.1, Compliant With 2.0
  - This module can be used in the SMD.
  - It's made through RoHS process.
  - The board PIN is half hole size.
  - Has a 2.4GHz digital wireless transceiver.
  - Bases at CSR BC04 Bluetooth technology.
  - Has the function of adaptive frequency hopping.
  - Small (27mm×13mm×2mm)
  - Peripherals circuit is simple.
  - It's at the Bluetooth class 2 power level.
  - Storage temperature range: -40 ℃ - 85℃， work temperature range: -25 ℃ - +75℃
  - Any wave inter Interference: 2.4MHz， the power of emitting: 3 dBm.
  - Bit error rate: 0. Only the signal decays at the transmission link, bit error may be produced. For example, when RS232 or TTL is being processed, some signals may decay.

- Low power consumption
- Has high-performance wireless transceiver system
- Low Cost

- Application fields:
  - Bluetooth Car Handsfree Device
  - Bluetooth GPS
  - Bluetooth PCMCIA , USB Dongle
  - Bluetooth Data Transfer
- Software
  - CSR

3. PINs description

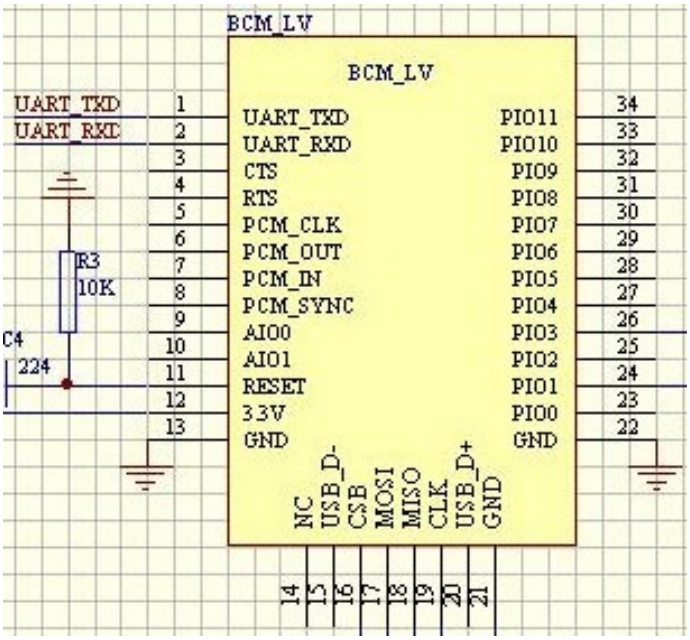


Figure 3 PIN configuration

The PINs at this block diagram is as same as the physical one.

PIN Name	PIN #	Pad type	Description	Note
GND	13 21 22	VSS	Ground pot	
1V8	14	VDD	Integrated 1.8V (+) supply with On-chip linear regulator output within 1.7-1.9V	
VCC	12	3.3V		
AIO0	9	Bi-Directional	Programmable input/output line	
AIO1	10	Bi-Directional	Programmable input/output line	

PIO0	23	Bi-Directional RX EN	Programmable input/output line, control output for LNA(if fitted)	
PIO1	24	Bi-Directional TX EN	Programmable input/output line, control output for PA(if fitted)	
PIO2	25	Bi-Directional	Programmable input/output line	
PIO3	26	Bi-Directional	Programmable input/output line	
PIO4	27	Bi-Directional	Programmable input/output line	
PIO5	28	Bi-Directional	Programmable input/output line	
PIO6	29	Bi-Directional	Programmable input/output line	CLK_REQ
PIO7	30	Bi-Directional	Programmable input/output line	CLK_OUT
PIO8	31	Bi-Directional	Programmable input/output line	
PIO9	32	Bi-Directional	Programmable input/output line	
PIO10	33	Bi-Directional	Programmable input/output line	
PIO11	34	Bi-Directional	Programmable input/output line	
RESETB	11	CMOS Input with weak intemal pull-down		
UART_RTS	4	CMOS output, tri-stable with weak internal pull-up	UART request to send, active low	
UART_CTS	3	CMOS input with weak internal pull-down	UART clear to send, active low	
UART_RX	2	CMOS input with weak internal pull-down	UART Data input	
UART_TX	1	CMOS output, Tri-stable with weak internal pull-up	UART Data output	
SPI_MOSI	17	CMOS input with weak internal pull-down	Serial peripheral interface data input	
SPI_CSB	16	CMOS input with weak internal	Chip select for serial peripheral interface, active low	

		pull-up		
SPI_CLK	19	CMOS input with weak internal pull-down	Serial peripheral interface clock	
SPI_MISO	18	CMOS input with weak internal pull-down	Serial peripheral interface data Output	
USB_-	15	Bi-Directional		
USB_+	20	Bi-Directional		
1.8V	14		1.8V external power supply input	Default : 1.8V internal power supply.
PCM_CLK	5	Bi-Directional		
PCM_OUT	6	CMOS output		
PCM_IN	7	CMOS Input		
PCM_SYNC	8	Bi-Directional		

4. The parameters and mode of product

LINVOR BLUE T

www.linvor.com

Bluetooth Module

Bluetooth

CSR,BC417143B

V 2.0

2006/09/6

蓝牙 **RF** 模块

1. 采用 CSR BC4 +8M FLASH 方案
2. 具有 PIO0-PIO11、AIO0、AIO1、USB、PCM、UART 及 SPI 接口，模块内置 8MFLASH，功能强大，用户可定制软件,适用于各种蓝牙设备，内置 RF 天线,便于调试。

蓝牙协议版本	Bluetooth Specifcation V2.0 With EDR
USB 协议 USB Protocol	Full Speed USB V1.1 Compliant With USB V2.0
频率	2.4Ghz ISM band
调制方式	GFSK(Gaussian Frequency Shift Keying)
发射功率	-4 ->4 dBm, Class 2
灵敏度	≤-80dBm at 0.1% BER
通讯速率	Asynchronous:2Mbps(Max)
供电电源	3.3V
工作温度	-20~+55 Centigrade
封装尺寸	27mmX13mmX2mm

If you want more information, please visit www.wavesen.com.

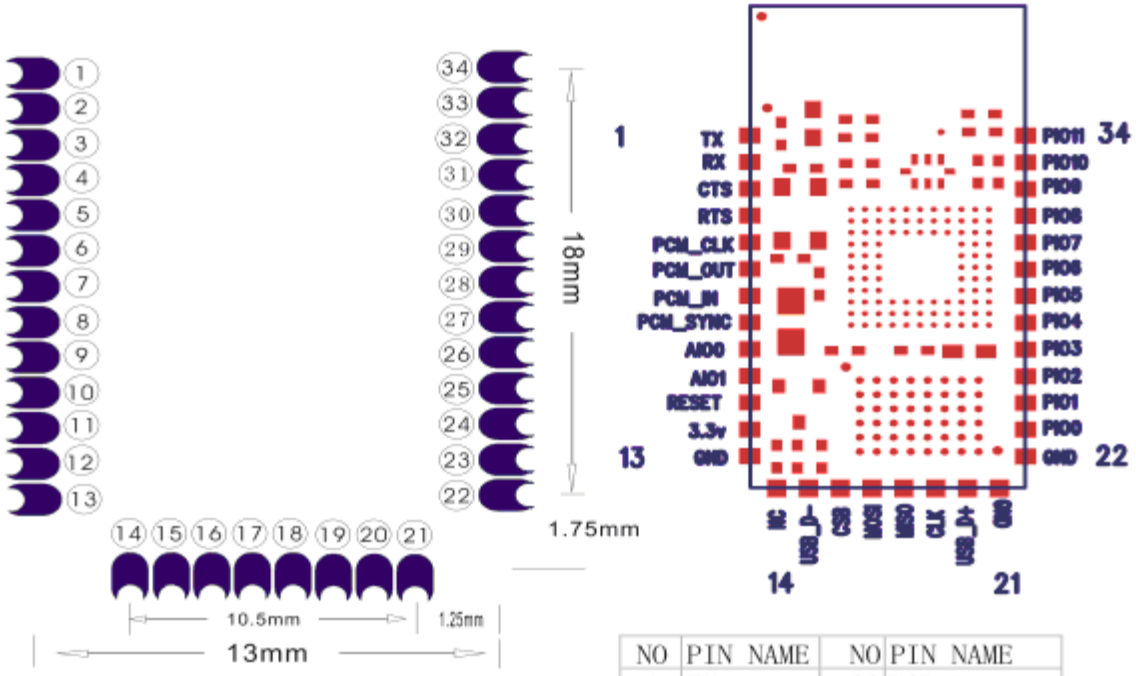
www.wavesen.com Phone: 020-84083341 Fax: 020-84332079 QQ:1043073574
Address: Room 527, No.13, Jiangong Road, Tianhe software park, Tianhe district, Guangzhou Post: 510660
Technology consultant: support@wavesen.com Business consultant:sales@wavesen.com
Complaint and suggestion: sunbirdit@hotmail.com

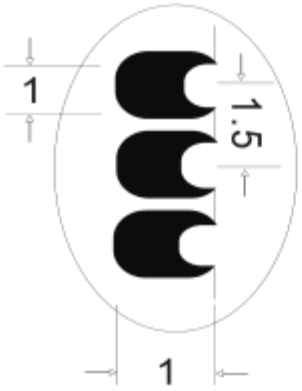
LINVOR BLUE T

www.linvor.com

LV-BC-2.0

单位： mm





PCB Layout 请参考实物

NO	PIN NAME	NO	PIN NAME
1	TX	20	USB D+
2	RX	21	GND
3	CTS	22	GND
4	RTS	23	PIO0
5	PCM CLK	24	PIO1
6	PCM OUT	25	PIO2
7	PCM IN	26	PIO3
8	PCM SYNC	27	PIO4
9	AIO0	28	PIO5
10	AIO1	29	PIO6
11	RESET	30	PIO7
12	3.3V	31	PIO8
13	GND	32	PIO9
14	NC	33	PIO10
15	USB D-	34	PIO11
16	CSB		
17	MOSI		
18	MISO		
19	CLK		

5. Block diagram

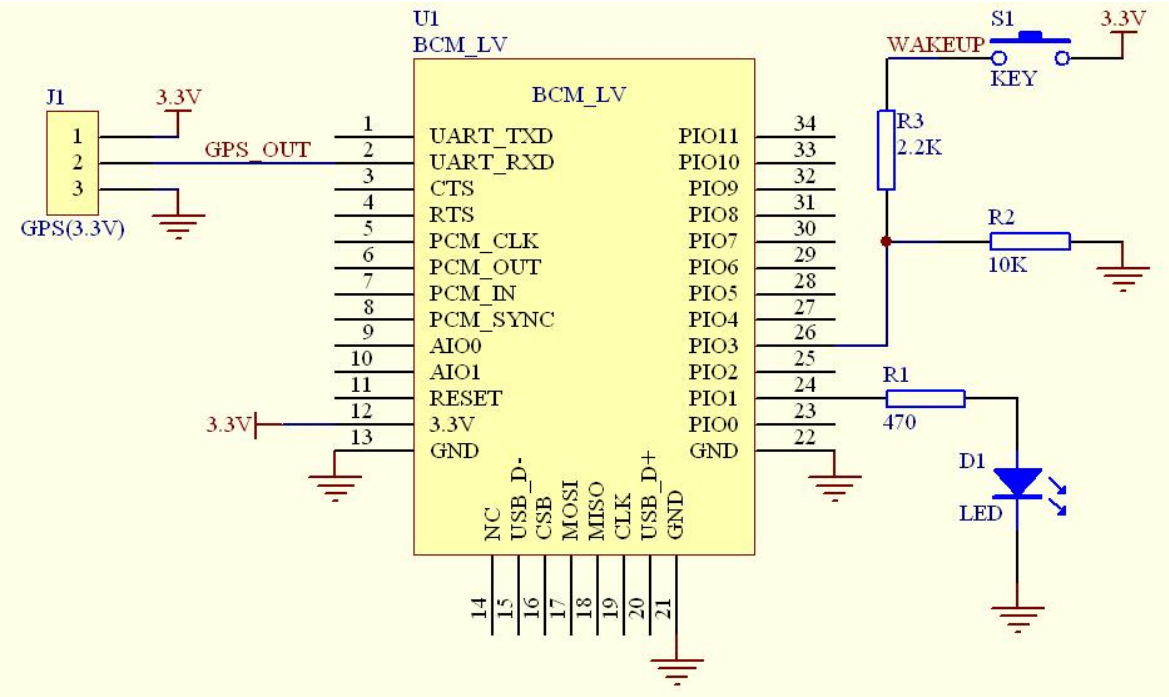


Figure 5 Block diagram 1

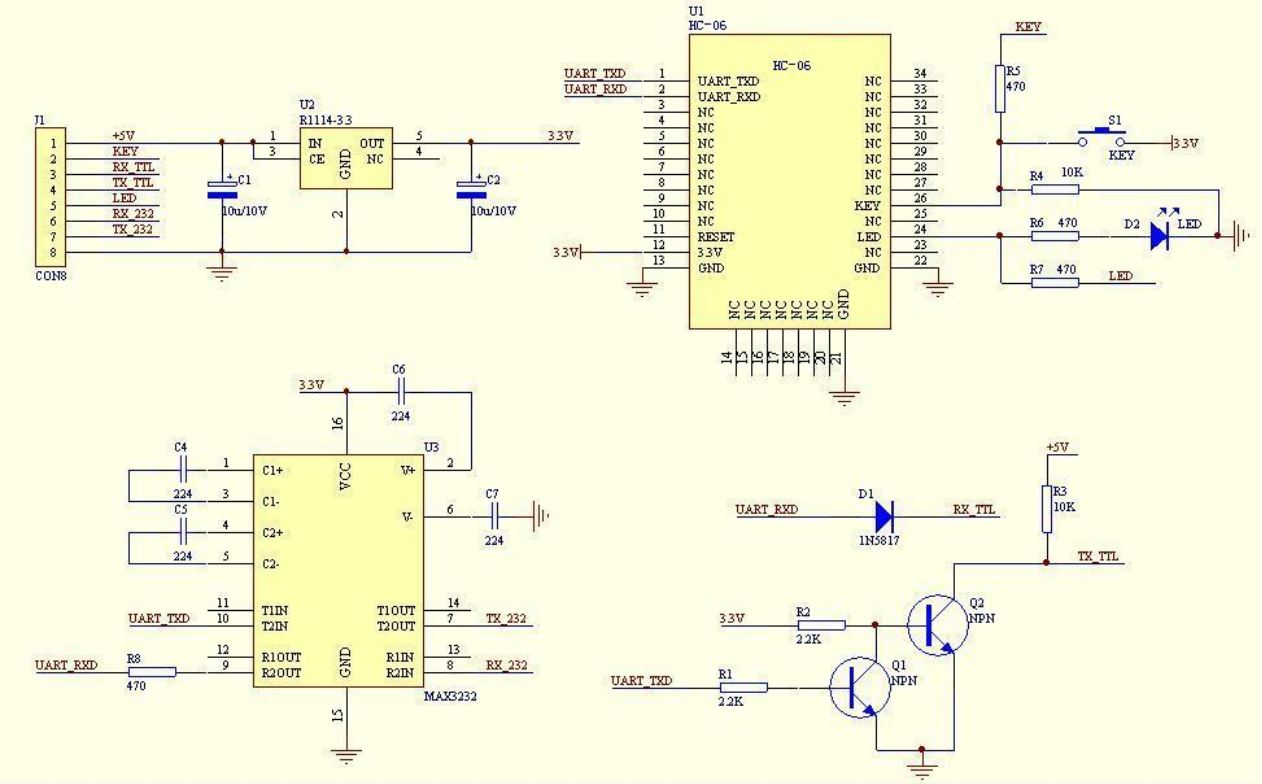


Figure 5 Block diagram 2

www.wavesen.com Phone: 020-84083341 Fax: 020-84332079 QQ:1043073574
Address: Room 527, No.13, Jiangong Road, Tianhe software park, Tianhe district, Guangzhou Post: 510660
Technology consultant: support@wavesen.com Business consultant:sales@wavesen.com
Complaint and suggestion: sunbirdit@hotmail.com

HC-04/06 master device has a function of remembering the last paired slave device. As a master device, it will search the last paired salve device until the connection is built. But if the WAKEUP bottom is pressed, HC-04/06 will lose the memory and research the new slave device.

6. Debugging device

6.1 Device

PC, hardware, 3G，3G Frequency Counter (SP3386), 3.15V DC power supply, Shielding，Bluetooth Test box.

6.2 Software

7. Characteristic of test

Test Condition 25℃ RH 65%

		Min	Typ	Max	Unit
1.	Carrier Freq. (*ISM Band*)	2.4		2.4835	MHz
2.	RF O/P Power	-6	2	4	dBm
3.	Step size of Power control	2		8	dB
4.	Freq. Offset (*Typical Carrier freq.*)	-75		75	KHz
5.	Carrier Freq. drift (*Hopping on, drift rate/50uS*)	-20		20	KHz
	1 slot packet	-25		25	KHz
	3 slot packet	-40		-40	KHz
6.	Average Freq. Deviations (*Hopping off, modulation*)	140		175	KHz
	Freq. Deviation	115			KHz
	Ratio of Freq. Deviation	0.8			
7 .	Receive Sensitivity @< 0.1% BER*(Bit error rate*)	-83			dBm

www.wavesen.com Phone: 020-84083341 Fax: 020-84332079 QQ:1043073574
Address: Room 527, No.13, Jiangong Road, Tianhe software park, Tianhe district, Guangzhou Post: 510660
Technology consultant: support@wavesen.com Business consultant:sales@wavesen.com
Complaint and suggestion: sunbirdit@hotmail.com

8. Test diagram

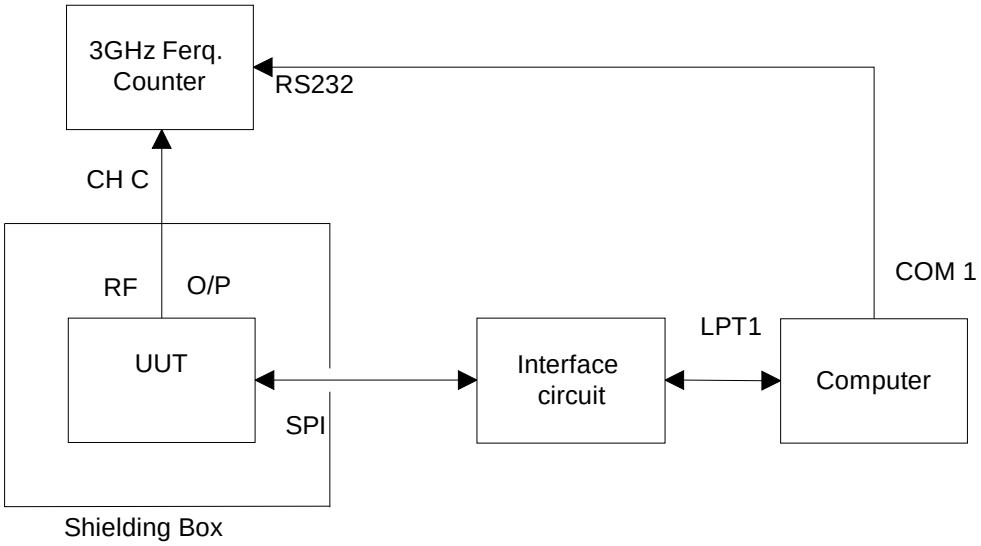


Fig 1. Programming and Freq. Alignment

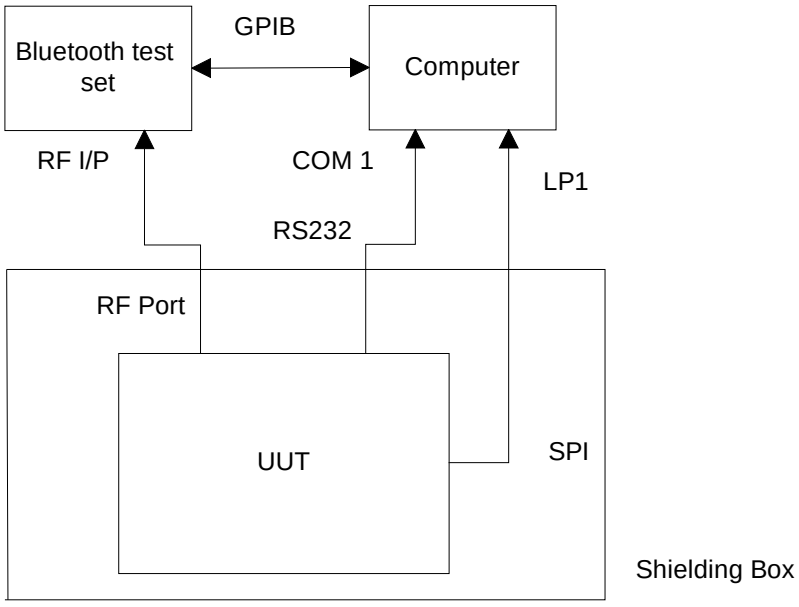


Fig 2 RF parameter Test Procedure

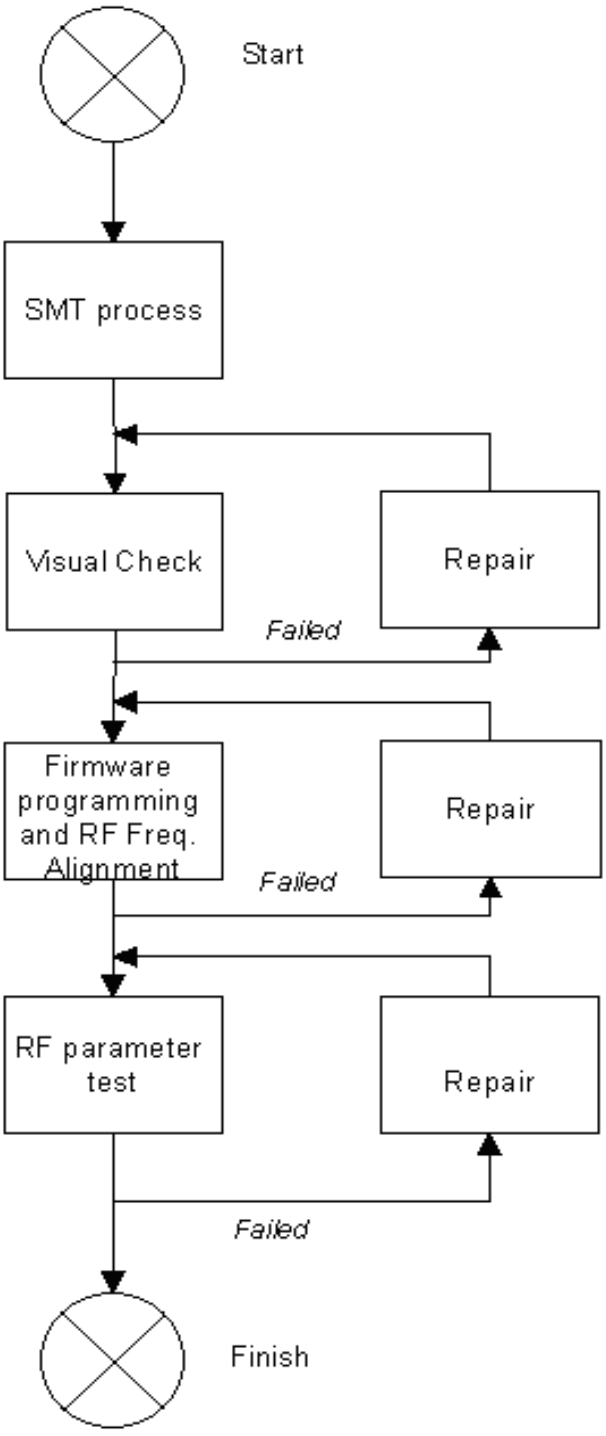


Fig 3 Assemble/Alignment/Testing Flow Chart

9. AT command set

The way to the AT command mode: supply power to the module, it will enter to the AT mode if it needn't pair. The interval of command is about 1 second.

Default parameter: Baud rate:9600N81, ID: linvor, Password:1234

1. Test communication

Send: AT (please send it every second)

Back: OK

2. Reset the Bluetooth serial baud rate

Send: AT+BAUD1

Back: OK1200

Send: AT+BAUD2

Back: OK2400

……

1---------1200

2---------2400

3---------4800

4---------9600 (Default)

5---------19200

6---------38400

7---------57600

8---------115200

9---------230400

A---------460800

B---------921600

C---------1382400

PC can't support the baud rate lager than 115200. The solution is: make the MCU have higher baud rate (lager than 115200) through programming, and reset the baud rate to low level through the AT command.

The baud rate reset by the AT command can be kept for the next time even though the power is cut off.

3. Reset the Bluetooth name

Send: AT+NAMEname

Back: OKname

Parameter name: Name needed to be set (20 characters limited)
Example:
Send: AT+NAMEbill_gates
Back: OKname

Now, the Bluetooth name is reset to be “bill_gates”

The parameter can be kept even though the power is cut off. User can see the new Bluetooth name in PDA refresh service. (Note: The name is limited in 20 characters.)

4. change the Bluetooth pair password
Send: AT+PINxxxx
Back:OKsetpin

Parameter xxxx: The pair password needed to be set, is a 4-bits number. This command can be used in the master and slave module. At some occasions, the master module may be asked to enter the password when the master module tries to connect the slave module (adapter or cell-phone). Only if the password is entered, the successful connection can be built. At the other occasions, the pair can be finish automatically if the master module can search the proper slave module and the password is correct. Besides the paired slave module, the master can connect the other devices who have slave module, such as Bluetooth digital camera, Bluetooth GPS, Bluetooth serial printer etc.
Example:
Send: AT+PIN8888
Back: OKsetpin
Then the password is changed to be 8888, while the default is 1234.
This parameter can be kept even though the power is cut off.

5. No parity check (The version, higher than V1.5, can use this command)
Send: AT+PN (This is the default value)
Back: OK NONE

6. Set odd parity check (The version, higher than V1.5, can use this command)
Send: AT+PO
Back: OK ODD

7. Set even parity check(The version, higher than V1.5, can use this command)
Send: AT+PE
Back: OK EVEN

8. Get the AT version

Send: AT+VERSION

Back: LinvorV1.n

www.wavesen.com Phone: 020-84083341 Fax: 020-84332079 QQ:1043073574
Address: Room 527, No.13, Jiangong Road, Tianhe software park, Tianhe district, Guangzhou Post: 510660
Technology consultant: support@wavesen.com Business consultant:sales@wavesen.com
Complaint and suggestion: sunbirdit@hotmail.com

MANTECH ELECTRONICS
International Components Distributor
A MOBICON COMPANY

TEL JHB : (011) 493-9307
CAPE : (021) 535-3150
KZN : (031) 309-7686

FAX : (011) 493-9319

sales@mantech.co.za

www.mantech.co.za

I2C interface for LCD

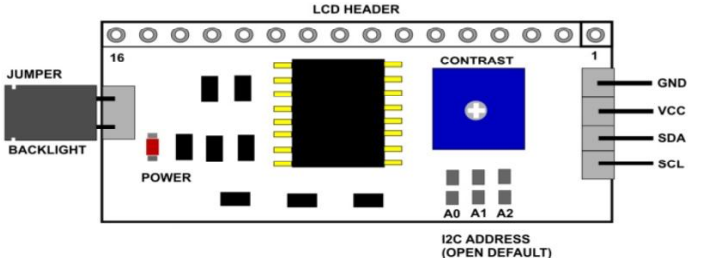
Discription:

This LCD2004 is a great I2C interface for 2x16 and 4x20 LCD displays. With the limited pin resources, your project may be out of resources using normal LCD shield. With this I2C interface LCD module, you only need 2 lines (I2C) to display the information. If you already has I2C devices in your project, this LCD module actually cost no more resources at all. Fantastic for Arduino based projects.

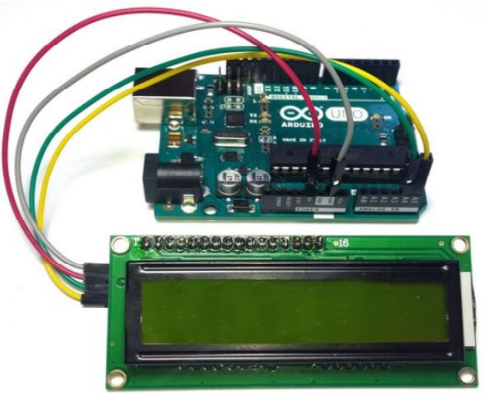
Specification:

Compatible with 16x2 and 20x4 LCD's
Default I2C Address = 0X27
Address selectable - Range 0x20 to 0x27

Board Layout:



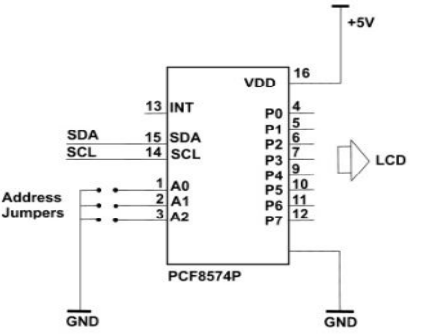
I2C Address Setup:

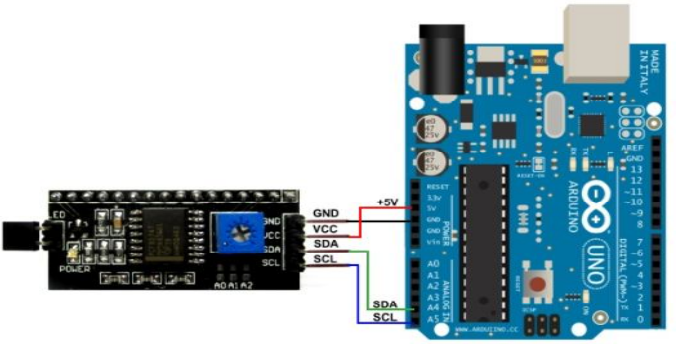


The LCD2004 board utilized the PCF8574 I/O expander. This nifty little chip provides eight bits of parallel I/O addressabl by a I2C bus address – 0x00 to 0x27. SainSmart tied all address leads to Vcc, so the LCD2004 board's I2C address is permanently fixed at hex 27. This is rather limiting since no additional LCD2004s can be added to the bus. Anyway, you simply address the board and write an eight bit value which is then presented on the output pins of the PCF8574, which, in this case, are connected to the HD44780 based LCD screen.

INPUTS			I2C SLAVE ADDRESS
A2	A1	A0	
L	L	L	0x20
L	L	H	0x21
L	H	L	0x22
L	H	H	0x23
H	L	L	0x24
H	L	H	0x25
H	H	L	0x26
H	H	H	0x27

H = Open Jumper L = Close Jumper






```
//Arduino Code
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27,2,1,0,4,5,6,7,3, POSITIVE); // Initialize LCD Display at address 0x27
                                                       // unmodified backpack
void setup() {
  // activate LCD module
  lcd.begin (16,2); // for 16 x 2 LCD module
  lcd.setBacklightPin(3,POSITIVE);
  lcd.setBacklight(HIGH);
  }

void loop() {
  lcd.home (); // set cursor to 0,0
  lcd.print(" Hello, world!");
  lcd.setCursor (0,1);        // go to start of 2nd line
  lcd.print(millis());
  delay(1000);
  lcd.setBacklight(LOW);      // Backlight off
  delay(500);
  lcd.setBacklight(HIGH);     // Backlight on

  delay(1000);
}       // END
```

Check for more info at
https://arduino-info.wikispaces.com/LCD-Blue-I2C

Systronix 20x4 LCD
Brief Technical Data

July 31, 2000

Here is brief data for the Systronix 20x4 character LCD. It is a DataVision part and uses the Samsung KS0066 LCD controller. It's a clone of the Hitachi HD44780. We're not aware of any incompatabilities between the two - at least we have never seen any in all the code and custom applications we have done.

This 20x4 LCD is electrically and mechanically interchangeable with 20x4 LCDs from several other vendors. The only differences we've seen among different 20x4 LCDs are:

1) LED backlight brightness, voltage and current vary widely, as does the quality of the display

2) There is a resistor "Rf" which sets the speed of the LCD interface by controlling the internal oscillator frequency. Several displays we have evaluated have a low resistor value. This makes the display too slow. Looking at the Hitachi data sheet page 56, it appears that perhaps the "incorrect" resistor is really intended for 3V use of the displays.

At 5V the resistor Rf should be 91 Kohms. At 3V it should be 75 Kohms. Using a 3V display at 5V is acceptable from a voltage standpoint (the display can operate on 3-5V) but the oscillator will then be running too slowly. One fix is to always check the busy flag and not use a fixed time delay in your code, then it will work regardless of the LCD speed. The other option is to always allow enough delay for the slower display.

All Systronix 20x4 LCDs have the 91 Kohm resistor and are intended for 5V operation.

Thank you for purchasing Systronix embedded control products and accessories. If you have any other questions please email to support@systronix.com or phone +1-801-534-1017, fax +1-801-534-1019.

■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Standard Value			Unit
		Min.	Typ.	Max.	
Supply Voltage for Logic	V_{DD-}	0	—	7.0	V
Supply Voltage for LCD Driver	V_{DD}-V_{EE}	—	—	13.5	V
Input Voltage	V_I	V_{SS}	—	V_{DD}	V
Operature Temp.	Topr	0	—	50	°C
Storage Temp.	Tstg	-20	—	70	°C

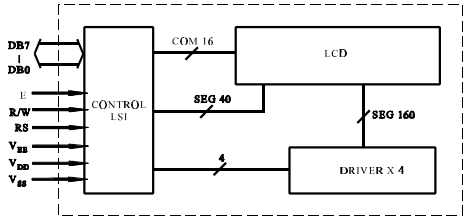
■ ELECTRICAL CHARACTERISTICS (REFLECTIVE TYPE)

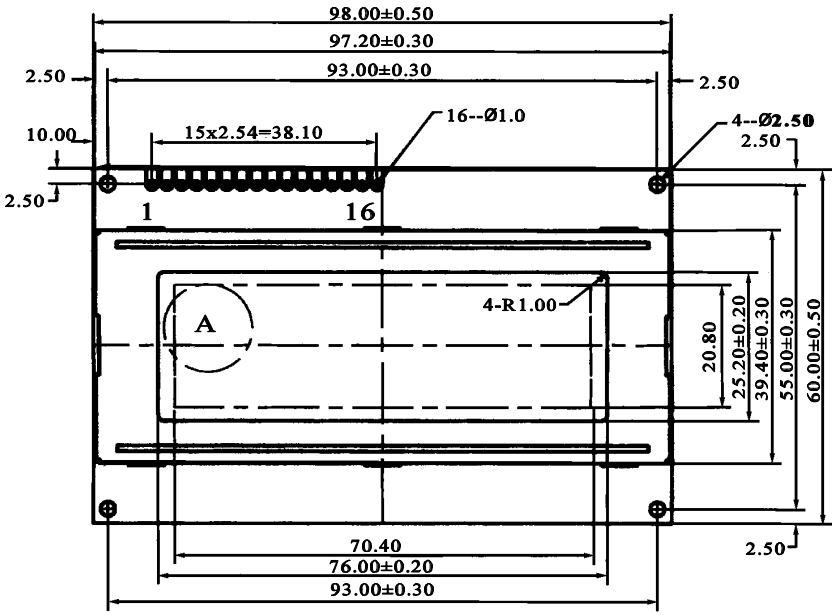
Item	Symbol	Test Condition	Standard Value			Unit
			Min.	Typ.	Max.	
Input "High" Voltage	V_{IH}	—	2.2	—	V_{EE}	V
Input "Low" Voltage	V_{IL}	—	—	—	0.6	V
Output "High" Voltage	V_{OH}	I_{OH}=0.2mA	2.2	—	—	V
Output "Low" Voltage	V_{OL}	I_{OL}=1.2mA	—	—	0.4	V
Supply Current	I_{DD}	V_{DD}=5.0A	—	2.5	4.0	mA

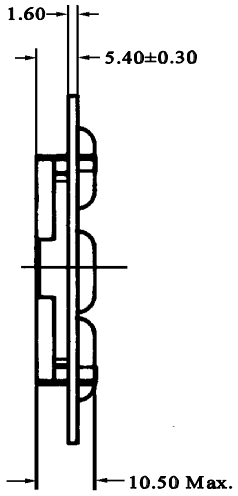
■ PIN FUNCTIONS

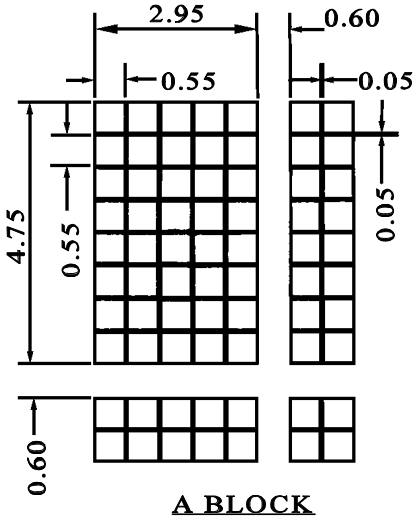
No	Symbol	Function	No	Symbol	Function
1	V_{SS}	GND, 0V	10	DB3	Data Bus
2	V_{DD}	+5V	11	DB4	—
3	V_{EE}	for LCD Drive	12	DB5	—
4	RS	Function Select	13	DB6	—
5	R/W	Read/Write	14	DB7	—
6	E	Enable Signal	15	LEDA	LED Power Supply
7-9	DB0-DB2	Data Bus Line	16	LEDA	

■ BLOCK DIAGRAM









A BLOCK

Table 4 Correspondence between Character Codes and Character Patterns (ROM Code: A00)

Upper 4 Bits / Lower 4 Bits	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000	CG RAM (1)															
xxxx0001	(2)															
xxxx0010	(3)															
xxxx0011	(4)															
xxxx0100	(5)															
xxxx0101	(6)															
xxxx0110	(7)															
xxxx0111	(8)															
xxxx1000	(1)															
xxxx1001	(2)															
xxxx1010	(3)															
xxxx1011	(4)															
xxxx1100	(5)															
xxxx1101	(6)															
xxxx1110	(7)															
xxxx1111	(8)															

Note: The user can specify any pattern for character-generator RAM.

Initializing by Instruction

If the power supply conditions for correctly operating the internal reset circuit are not met, initialization by instructions becomes necessary.

Refer to Figures 25 and 26 for the procedures on 8-bit and 4-bit initializations, respectively.

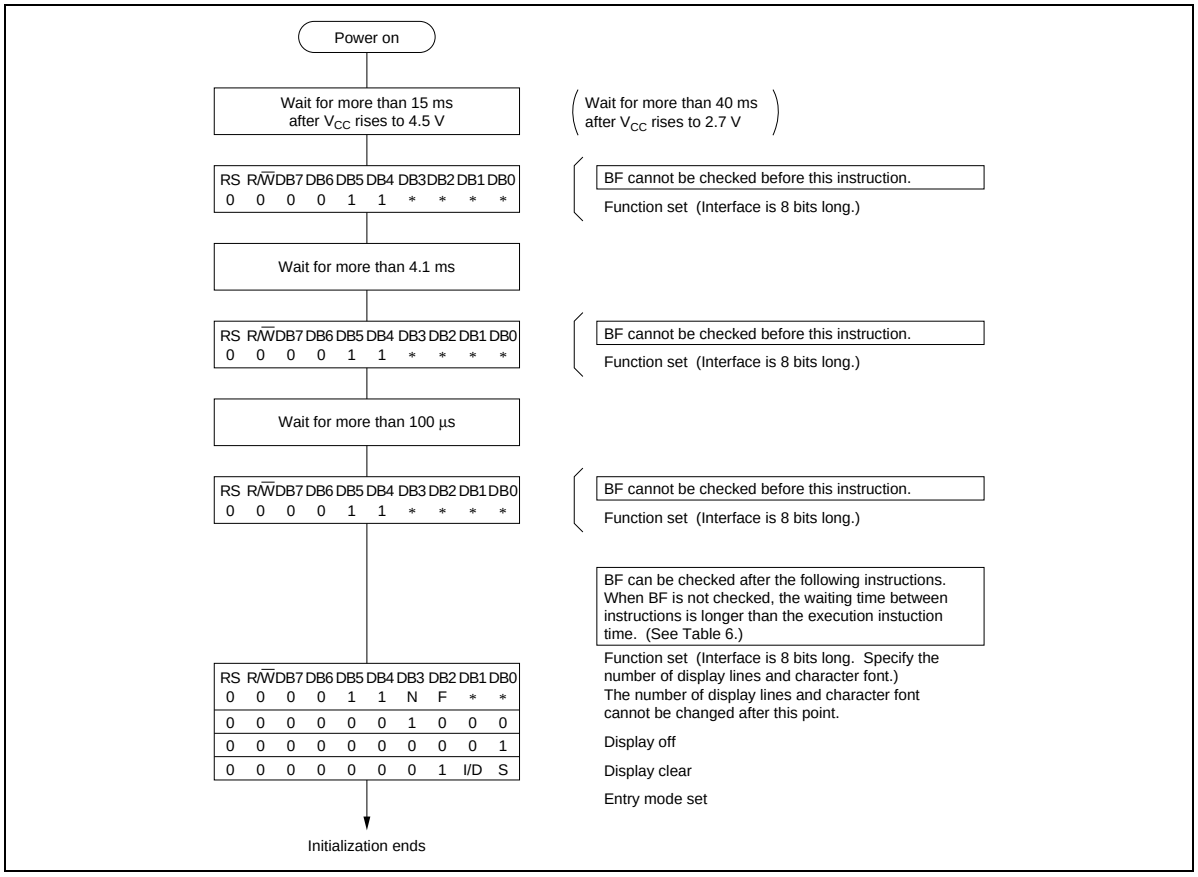


Figure 25 8-Bit Interface

Reset Function

Initializing by Internal Reset Circuit

An internal reset circuit automatically initializes the HD44780U when the power is turned on. The following instructions are executed during the initialization. The busy flag (BF) is kept in the busy state until the initialization ends (BF = 1). The busy state lasts for 10 ms after V_{CC} rises to 4.5 V.

1. Display clear
2. Function set:
   DL = 1; 8-bit interface data
   N = 0; 1-line display
   F = 0; 5 × 8 dot character font
3. Display on/off control:
   D = 0; Display off
   C = 0; Cursor off
   B = 0; Blinking off
4. Entry mode set:
   I/D = 1; Increment by 1
   S = 0; No shift

Note: If the electrical characteristics conditions listed under the table Power Supply Conditions Using Internal Reset Circuit are not met, the internal reset circuit will not operate normally and will fail to initialize the HD44780U. For such a case, initial-ization must be performed by the MPU as explained in the section, Initializing by Instruction.

Instructions

Outline

Only the instruction register (IR) and the data register (DR) of the HD44780U can be controlled by the MPU. Before starting the internal operation of the HD44780U, control information is temporarily stored into these registers to allow interfacing with various MPUs, which operate at different speeds, or various peripheral control devices. The internal operation of the HD44780U is determined by signals sent from the MPU. These signals, which include register selection signal (RS), read/

write signal (R/$\overline{W}$), and the data bus (DB0 to DB7), make up the HD44780U instructions (Table 6). There are four categories of instructions that:

- Designate HD44780U functions, such as display format, data length, etc.
- Set internal RAM addresses
- Perform data transfer with internal RAM
- Perform miscellaneous functions

Normally, instructions that perform data transfer with internal RAM are used the most. However, auto-incrementation by 1 (or auto-decrementation by 1) of internal HD44780U RAM addresses after each data write can lighten the program load of the MPU. Since the display shift instruction (Table 11) can perform concurrently with display data write, the user can minimize system development time with maximum programming efficiency.

When an instruction is being executed for internal operation, no instruction other than the busy flag/address read instruction can be executed.

Because the busy flag is set to 1 while an instruction is being executed, check it to make sure it is 0 before sending another instruction from the MPU.

Note: Be sure the HD44780U is not in the busy state (BF = 0) before sending an instruction from the MPU to the HD44780U. If an instruction is sent without checking the busy flag, the time between the first instruction and next instruction will take much longer than the instruction time itself. Refer to Table 6 for the list of each instruc-tion execution time.

Table 6 Instructions

	Code											Execution Time (max) (when f_{cp} or f_{osc} is 270 kHz)
Instruction	**RS**	**R/$\overline{W}$**	**DB7**	**DB6**	**DB5**	**DB4**	**DB3**	**DB2**	**DB1**	**DB0**	**Description**	
Clear display	0	0	0	0	0	0	0	0	0	1	Clears entire display and sets DDRAM address 0 in address counter.	
Return home	0	0	0	0	0	0	0	0	1	—	Sets DDRAM address 0 in address counter. Also returns display from being shifted to original position. DDRAM contents remain unchanged.	1.52 ms
Entry mode set	0	0	0	0	0	0	0	1	I/D	S	Sets cursor move direction and specifies display shift. These operations are performed during data write and read.	37 µs
Display on/off control	0	0	0	0	0	0	1	D	C	B	Sets entire display (D) on/off, cursor on/off (C), and blinking of cursor position character (B).	37 µs
Cursor or display shift	0	0	0	0	0	1	S/C	R/L	—	—	Moves cursor and shifts display without changing DDRAM contents.	37 µs
Function set	0	0	0	0	1	DL	N	F	—	—	Sets interface data length (DL), number of display lines (N), and character font (F).	37 µs
Set CGRAM address	0	0	0	1	ACG	ACG	ACG	ACG	ACG	ACG	Sets CGRAM address. CGRAM data is sent and received after this setting.	37 µs
Set DDRAM address	0	0	1	ADD	ADD	ADD	ADD	ADD	ADD	ADD	Sets DDRAM address. DDRAM data is sent and received after this setting.	37 µs
Read busy flag & address	0	1	BF	AC	AC	AC	AC	AC	AC	AC	Reads busy flag (BF) indicating internal operation is being performed and reads address counter contents.	0 µs

Table 6 Instructions (cont)

Instruction	Code RS	R/$\overline{W}$	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Description	Execution Time (max) (when f_{cp} or f_{OSC} is 270 kHz)
Write data to CG or DDRAM	1	0	Write data								Writes data into DDRAM or CGRAM.	37 µs t_{ADD} = 4 µs*
Read data from CG or DDRAM	1	1	Read data								Reads data from DDRAM or CGRAM.	37 µs t_{ADD} = 4 µs*
	I/D = 1: Increment I/D = 0: Decrement S = 1: Accompanies display shift S/C = 1: Display shift S/C = 0: Cursor move R/L = 1: Shift to the right R/L = 0: Shift to the left DL = 1: 8 bits, DL = 0: 4 bits N = 1: 2 lines, N = 0: 1 line F = 1: 5 × 10 dots, F = 0: 5 × 8 dots BF = 1: Internally operating BF = 0: Instructions acceptable										DDRAM: Display data RAM CGRAM: Character generator RAM ACG: CGRAM address ADD: DDRAM address (corresponds to cursor address) AC: Address counter used for both DD and CGRAM addresses	Execution time changes when frequency changes Example: When f_{cp} or f_{OSC} is 250 kHz, $37\ \mu s \times \frac{270}{250} = 40\ \mu s$

Note: — indicates no effect.

* After execution of the CGRAM/DDRAM data write or read instruction, the RAM address counter is incremented or decremented by 1. The RAM address counter is updated after the busy flag turns off. In Figure 10, t_{ADD} is the time elapsed after the busy flag turns off until the address counter is updated.

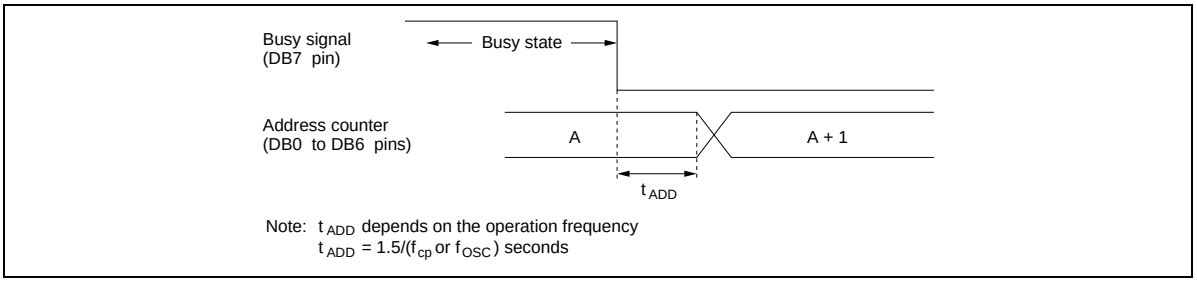


Figure 10 Address Counter Update

```
#include <SoftwareSerial.h>

SoftwareSerial btSerial(50, 51); // RX, TX

#include <PZEM004T.h>

PZEM004T pzem1(&Serial1);

PZEM004T pzem2(&Serial2);

PZEM004T pzem3(&Serial3);

IPAddress ip(192,168,1,1);


//#define BLYNK_PRINT Serial1

#include <BlynkSimpleSerialBLE.h>//menambahkan library bluetooth blynk

//BlynkTimer timer;// menambahkan timer(penggunaan);


char auth[] = "32cd33eb2574467fa7c295d55f263699";//nomor token untk terhubung aplikasi blynk(harus sama dengan yg di aplikasi blynk)


#include <EEPROMex.h>//menambahkan library eeprom(mempermudah menyimpan data sisa kwh dalam bentuk bilangan desimal/tipe data float)


#include <Wire.h> //menambahkan library i2c

#include <LiquidCrystal_I2C.h>//menambahkan library i2c lcd

#include <Keypad.h> //menambahkan library keypad


// menamai pin relay

#define relay1pin 12 // relay1
```

```
#define relay2pin 11 // relay2
#define relay3pin 10 // relay3

//inisisalisasi variable untuk hasil pembacaan tegangan puncak/vpp dari sensor arus
float result1=0;
float result2=0;
float result3=0;

//inisisalisasi variable untuk hasil pembacaan akhir sensor arus dan tegangan
float sensorteg = 0;
float sensorarus1 = 0;
float sensorarus2 = 0;
float sensorarus3 = 0;
float sensordaya1 = 0;
float sensordaya2 = 0;
float sensordaya3 = 0;
float cospi1 =1;
float cospi2 =1;
float cospi3 =1;

int alamateeprom=4 ; //inisialisasi variable untuk alamat eeprom pada alamat '0'
int alamateeprom2=0;
//inisisalisasi variable untuk pengaturan delay meggunakan millis
```

```
unsigned long msec = 0;
unsigned long delaydisplay = 0;
float time = 0.0;

//inisisalisasi variable untuk perhitungan arus perdetik
float ampSeconds1 = 0.0;
float ampSeconds2 = 0.0;
float ampSeconds3 = 0.0;
//inisisalisasi variable untuk perhitungan arus perjam
float ampHours1 = 0.0;
float ampHours2 = 0.0;
float ampHours3 = 0.0;
//inisisalisasi variable untuk perhitungan daya(watt) perjam
float wattHours1 = 0.0;
float wattHours2 = 0.0;
float wattHours3 = 0.0;
//inisisalisasi variable untuk perhitungan daya(kilowatt) perjam
float kilowattHours1 = 0.0;
float kilowattHours2 = 0.0;
float kilowattHours3 = 0.0;

float sisakwh;//inisisalisasi variable untuk nilai sisakwh
float kwhfinal=0.0; //inisisalisasi variable untuk nilai pembacaan kwh realtime
```

```
float kwhsimpan; //inisisalisasi variable untuk nilai kwh yg disimpan
//inisisalisasi variable untuk data beban yang dikirimkan ke aplikasi blynk
float beban1;
float beban2;
float beban3;
int bebantotal;
int maxbeban; ////inisisalisasi variable untuk batasan nilai beban untuk mode otomatis

int manualmode=0;////inisisalisasi variable untuk mode kwh(default 0=otomatis)
int manual1=1;//inisisalisasi membaca status manual/auto dari aplikasi blynk

int ledpin1 = 25;
int ledpin2 = 24;
int buzzer = 28;

float nilai=0;//inisisalisasi variable untuk nilai akhir input pulsa pada keypad
int nilai2=0;
char array[10],i=0;////inisisalisasi variabel berupa array untuk jumlah digit input pulsa pada keypad
const byte ROWS = 4; //four rows
const byte COLS = 4; //four columns
//define the cymbols on the buttons of the keypads
char hexaKeys[ROWS][COLS] = {
```

```
  {'1','4','7','*'},
  {'2','5','8','0'},
  {'3','6','9','#'},
  {'A','B','C','D'}
};
byte rowPins[ROWS] = {9, 8, 7, 6}; //connect to the row pinouts of the keypad
byte colPins[COLS] = {5, 4, 2, 3}; //connect to the column pinouts of the keypad
Keypad customKeypad = Keypad( makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS);

char lcdbuff[20];//inisisalisasi variable array untuk menyimpan karakter
char lcdbuff2[20];//inisisalisasi variable array untuk menyimpan karakter
char sisakwh2[7];//inisisalisasi variable array untuk menyimpan karakter
LiquidCrystal_I2C lcd ( 0x3f, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);

//fungsi untuk menampilkan nilai pulsa yg di input pada keypad
void simpan_nilai()
{
if(i==1)
{nilai=array[i];}
if(i>=2 && i<=11)
{nilai=(nilai*10)+array[i];}
}
```

```
//fungsi untuk membaca semua data yg di input pada keypad
void input_keypad()
{

  if (i<1)
  {i=0;nilai=0;}
  if (i>6)
  {nilai=(nilai-array[i])/10;i--; }
  if (i>0&&nilai==0)
  {i=0;}
  char customKey = customKeypad.getKey();
  switch (customKey)
   {
   //case NO_KEY:
   //break;

   case '0':
   i++;array[i]=0;simpan_nilai();digitalWrite (buzzer,HIGH);delay(100);digitalWrite (buzzer,LOW);break;
   case '1':
   i++;array[i]=1;simpan_nilai();digitalWrite (buzzer,HIGH);delay(100);digitalWrite (buzzer,LOW);break;
   case '2':
```

```
  i++;array[i]=2;simpan_nilai();digitalWrite (buzzer,HIGH);delay(100);digitalWrite
(buzzer,LOW);break;

  case '3':

  i++;array[i]=3;simpan_nilai();digitalWrite (buzzer,HIGH);delay(100);digitalWrite
(buzzer,LOW);break;

  case '4':

  i++;array[i]=4;simpan_nilai();digitalWrite (buzzer,HIGH);delay(100);digitalWrite
(buzzer,LOW);break;

  case '5':

  i++;array[i]=5;simpan_nilai();digitalWrite (buzzer,HIGH);delay(100);digitalWrite
(buzzer,LOW);break;

  case '6':

  i++;array[i]=6;simpan_nilai();digitalWrite (buzzer,HIGH);delay(100);digitalWrite
(buzzer,LOW);break;

  case '7':

  i++;array[i]=7;simpan_nilai();digitalWrite (buzzer,HIGH);delay(100);digitalWrite
(buzzer,LOW);break;

  case '8':

  i++;array[i]=8;simpan_nilai();digitalWrite (buzzer,HIGH);delay(100);digitalWrite
(buzzer,LOW);break;

  case '9':

  i++;array[i]=9;simpan_nilai();digitalWrite (buzzer,HIGH);delay(100);digitalWrite
(buzzer,LOW);break;

  case 'D':

    digitalWrite (buzzer,HIGH);delay(100);digitalWrite (buzzer,LOW);
```

```
    if(nilai==100000){kwhsimpan+=67.30;lcd.setCursor (1,3);lcd.print ("Benar
");EEPROM.writeFloat(alamateeprom,kwhsimpan);delay(1000);tampilkwh();}

    else if(nilai==50000){kwhsimpan+=35.6;lcd.setCursor (1,3);lcd.print ("Benar
");EEPROM.writeFloat(alamateeprom,kwhsimpan);delay(1000);tampilkwh();}

    else if(nilai==20000){kwhsimpan+=13.6;lcd.setCursor (1,3);lcd.print ("Benar
");EEPROM.writeFloat(alamateeprom,kwhsimpan);delay(1000);tampilkwh();}

    else {lcd.setCursor (1,3);lcd.print ("Salah              ");delay(1000);}

  nilai=0;i=0;break;

  case '#':

  nilai=(nilai-array[i])/10;i--;digitalWrite (buzzer,HIGH);delay(100);digitalWrite
(buzzer,LOW);break;

  case '*':

  digitalWrite (buzzer,HIGH);delay(100);digitalWrite (buzzer,LOW);

  kwhsimpan=0;

  EEPROM.writeFloat(alamateeprom,kwhsimpan);

  break;

  case 'A':

  digitalWrite (buzzer,HIGH);delay(100);digitalWrite (buzzer,LOW);

  if(nilai==1){kwhsimpan=0.01;lcd.setCursor (1,3);lcd.print ("Benar
");delay(1000);tampilkwh();}

  EEPROM.writeFloat(alamateeprom,kwhsimpan);

  break;

  case 'B':

  digitalWrite (buzzer,HIGH);delay(100);digitalWrite (buzzer,LOW);

  digitalWrite (ledpin1,HIGH);delay(500);digitalWrite (ledpin1,LOW);
```

```
  if(nilai==5){kwhsimpan=5.0;lcd.setCursor (1,3);lcd.print ("Benar
");delay(1000);tampilkwh();}

  EEPROM.writeFloat(alamateeprom,kwhsimpan);

  nilai=0;

  break;

  case 'C':

  if(nilai==0){lcd.setCursor (1,3);lcd.print (maxbeban);delay(1000);tampilkwh();}

  //if(nilai>0){nilai2 = (int)nilai; lcd.setCursor (1,3);lcd.print
("MaxBeban="+nilai2);EEPROM.writeInt(alamateeprom2,nilai2);delay(1000);tampil
kwh();nilai=0;}

  else if(nilai>0){maxbeban=(int)nilai; lcd.setCursor(1,3);lcd.print
(maxbeban);EEPROM.writeInt(alamateeprom2,maxbeban);delay(1000);tampilkwh();
nilai=0;}

  else break;

 }


}

void setup ()

{

 pzem1.setAddress(ip);

 pzem2.setAddress(ip);

 pzem3.setAddress(ip);

 kwhsimpan=EEPROM.readFloat(alamateeprom);//mengambil data yg disimpan di
eeprom ke variable kwhsimpan

 maxbeban=EEPROM.readInt(alamateeprom2);
```

```
pinMode (ledpin1,OUTPUT);
pinMode (ledpin2,OUTPUT);
pinMode (relay1pin,OUTPUT);
pinMode (relay2pin,OUTPUT);
pinMode (relay3pin,OUTPUT);
pinMode (buzzer,OUTPUT);
digitalWrite (relay1pin,HIGH);
digitalWrite (relay2pin,HIGH);
digitalWrite (relay3pin,HIGH);

Serial.begin(9600);
btSerial.begin(9600);

Wire.begin ();//memulai komunikasi i2c
//memulai aktifasi modul lcd i2c
lcd.begin (20,4);
lcd.setCursor (3,0);
lcd.print ("MOCHAMAD RIVAI");
lcd.setCursor (3,1);
lcd.print ("DWI RETNO A");
lcd.setCursor (1,2);
lcd.print ("TUGAS AKHIR 2018");
delay(1000);
```

```
  lcd.clear();


  Blynk.begin(btSerial, auth);//memulai komunikasi antara arduino ke aplikasi blynk via bluetooth menggunakan serial 1
}
//fungsi untuk membaca data sensor tegangan
//fungsi untuk membaca data puncak ke puncak/vpp dari sensor arus 1-3
//fungsi untuk enampilkan data sisa pulsa/kwh pada lcd baris 0
void tampilkwh()
{
 // input_keypad();
  //sisakwh=(sisakwh,3)-(,3);


  //dtostrf(kwhfinal,3,3,sisakwh2);
  //sisakwh=kwhsimpan;
  dtostrf(kwhsimpan,3,3,sisakwh2);
  sprintf(lcdbuff,"Pulsa: %s Kwh ",sisakwh2);
  lcd.setCursor (1,0);
  lcd.print(lcdbuff);
  //sisakwh = kwhbuff;
  //sisakwh2.toFloat();
}
```

```
//fungsi untuk menampilkan nilai pulsa yg diinput pada keypad ke lcd baris ke 3
void tampilpulsa()
{
  //input_keypad();
  dtostrf(nilai,0,0,lcdbuff2);
  sprintf(lcdbuff,"%s       ",lcdbuff2);
  lcd.setCursor (1,3);
  lcd.print (lcdbuff);
}

//fungsi untuk menampilkan nilai arus dan tegangan pada lcd
void tampilarusteg()
{

  //input_keypad();
  dtostrf(sensorarus1,2,2,lcdbuff2);
  sprintf(lcdbuff,"%sA ",lcdbuff2);
  lcd.setCursor (1,2);
  lcd.print (lcdbuff);
  dtostrf(sensorarus2,2,2,lcdbuff2);
  sprintf(lcdbuff,"%sA ",lcdbuff2);
  lcd.setCursor (7,2);
  lcd.print (lcdbuff);
```

```
  dtostrf(sensorarus3,2,2,lcdbuff2);
  sprintf(lcdbuff,"%sA ",lcdbuff2);
  lcd.setCursor (13,2);
  lcd.print (lcdbuff);
  dtostrf(sensorteg,2,2,lcdbuff2);
  sprintf(lcdbuff,"%sVAC ",lcdbuff2);
  lcd.setCursor (1,1);
  lcd.print (lcdbuff);
}

//fungsi untuk menghitung dayya yg terpakai
void hitungkwh()
{
  //input_keypad();

  sensordaya1= pzem2.power(ip);
  sensordaya2= pzem3.power(ip);
  sensordaya3= pzem1.power(ip);

  sensorteg = pzem2.voltage(ip);


  sensorarus1 = pzem2.current(ip);
```

```
if (sensorarus1<0){sensorarus1=0;cospi1=1;}
sensorarus2 = pzem3.current(ip);
if (sensorarus2<0){sensorarus2=0;cospi2=1;}
sensorarus3 = pzem1.current(ip);
if (sensorarus3<0){sensorarus3=0;cospi3=1;}

if (sensorarus1>0){cospi1=sensordaya1/(sensorteg*sensorarus1);}
if (sensorarus2>0){cospi2=sensordaya2/(sensorteg*sensorarus2);}
if (sensorarus3>0){cospi3=sensordaya3/(sensorteg*sensorarus3);}

//perhitungan arus perdetik
ampSeconds1 = sensorarus1*time;
ampSeconds2 = sensorarus2*time;
ampSeconds3 = sensorarus3*time;

//perhitungan arus per jam
ampHours1 = ampSeconds1/3600;
ampHours2 = ampSeconds2/3600;
ampHours3 = ampSeconds3/3600;

//perhitungan daya(watt) perjam
wattHours1 = sensorteg * ampHours1 *cospi1 ;
wattHours2 = sensorteg * ampHours2 *cospi2;
```

```
  wattHours3 = sensorteg * ampHours3 *cospi3;

  //perhitungan daya(kilo watt) perjam
  kilowattHours1 = wattHours1/1000;
  kilowattHours2 = wattHours2/1000;
  kilowattHours3 = wattHours3/1000;

  //menjumlahkan semua daya kwh
  kwhfinal = kilowattHours1+kilowattHours2+kilowattHours3;
}
//membaca data yg dikirim kan aplikasi blynk pada v7
BLYNK_WRITE(V7)
{
  manualmode = param.asInt(); // assigning incoming value from pin V1 to a variable
  // process received value
}
void loop ()
{
  input_keypad();
  unsigned long currentMillis = millis();
  unsigned long currentMillis2 = millis();
  unsigned long currentMillis3 = millis();
  unsigned long prevmillis;
```

```
  if (currentMillis-delaydisplay>=5000)//tiap 5detik menjalankan program didalamnya
  {
   delaydisplay=millis();
   if (sensorarus1>0.1||sensorarus2>0.1||sensorarus3>0.1)//jika arus salahsatu sensor melebihi 0.1A, maka kwhsimpan dikurangi kwhfinal(beban realtime)
   {
       kwhsimpan=kwhsimpan-kwhfinal;
       //tampilkwh();
       //sisakwh=kwhsimpan;
   }
      //kwhfinal=0;
  }
  if (currentMillis2-msec>=5000)//setiap 5 detik mereset waktu pembacaan kwh
  {
    msec=millis();
    //time=0;
  }
  if (currentMillis3-prevmillis>=60000)//setiap 1menit menyimpan nilai kwhsipan ke eeprom
  {
    prevmillis=millis();
    //lcd.setCursor (1,3);lcd.print ("Benar              ");
    EEPROM.writeFloat(alamateeprom,kwhsimpan);
```

```
  //time=0;
 }
 time =  (millis()-(float)msec) / 1000.0;//mengkonversi milisecond ke second
 hitungkwh();//memanggil fungsi hitungkwh
 tampilkwh();//memanggil fungsi tampilkwh
 tampilarusteg();//memanggil fungsi tampilarus teg
 tampilpulsa();//memanggil fungsi tampilpulsa


 if (kwhsimpan<=0.0){digitalWrite(relay1pin,LOW);digitalWrite(relay2pin,LOW);digitalWrite(relay3pin,LOW);digitalWrite(ledpin1,HIGH);delay(100); digitalWrite(ledpin1,LOW); }//jika kwh 0 maka semua relay aktif untuk mematikan sambungan ke beban
 if (manualmode==0&&kwhsimpan>10.0)//jika mode auto diaktifkan
 {
  if (bebantotal>maxbeban)
  {
   if (beban1>beban2&&beban1>beban3){digitalWrite(relay1pin,LOW);digitalWrite(relay2pin,HIGH);digitalWrite(relay3pin,HIGH);}//kondisi jika beban 1 lebih besar dari semua beban dan maksimal beban
   if (beban2>beban3&&beban2>beban1){digitalWrite(relay1pin,HIGH);digitalWrite(relay2pin,LOW);digitalWrite(relay3pin,HIGH);}//kondisi jika beban 2 lebih besar dari semua beban dan maksimal beban
   if (beban3>beban1&&beban3>beban2){digitalWrite(relay1pin,HIGH);digitalWrite(rela
```

```
y2pin,HIGH);digitalWrite(relay3pin,LOW);}//kondisi jika beban 3 lebih besar dari semua beban dan maksimal beban

    //else {digitalWrite(relay1pin,HIGH);digitalWrite(relay2pin,HIGH);digitalWrite(relay3pin,HIGH);}

   }

   //else {digitalWrite(relay1pin,HIGH);digitalWrite(relay2pin,HIGH);digitalWrite(relay3pin,HIGH);}

 }


 if (kwhsimpan>10.0&&manualmode==0)

 //jika mode manual diaktifkan dan pulsa   sudah diisi kembali

 {

  digitalWrite (ledpin2,HIGH);

  digitalWrite (ledpin1,LOW);

  digitalWrite (relay1pin,HIGH);digitalWrite(relay2pin,HIGH);digitalWrite(relay3pin,HIGH);

 }

 //if (manualmode==1){digitalWrite(relay1pin,manual1);digitalWrite(relay2pin,manual2);digitalWrite(relay3pin,manual3);}

 Blynk.run();//menjalankan fungsi blynk

 beban1=sensorarus1*sensorteg*cospi1;//perhitungan utuk mendapatkan nilai beban1

 beban2=sensorarus2*sensorteg* cospi2;//perhitungan utuk mendapatkan nilai beban2
```

```
  beban3=sensorarus3*sensorteg*cospi3; //perhitungan utuk mendapatkan nilai beban3

  bebantotal=(int)beban1+(int)beban2+(int)beban3;

  Blynk.virtualWrite(V0,kwhsimpan);//mengirim nilai kwh simpan ke blynk

  Blynk.virtualWrite(V1,beban1);//mengirim nilai beban1 ke blynk

  Blynk.virtualWrite(V2,beban2);//mengirim nilai beban2 ke blynk

  Blynk.virtualWrite(V3,beban3);//mengirim nilai beban3 ke blynk


}
```