

SKRIPSI

PEMBUANGAN AIR LIMBAH PUPUK BERBASIS MIKROKONTROLER ARM STM32F4



Oleh

MUHAMMAD SHALAKHUDIN ALYAZIDI

11.632.007

**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK ELEKTRO
UNIVERSITAS MUHAMMADIYAH
GRESIK
2018**

**PEMBUANGAN AIR LIMBAH PUPUK BERBASIS MIKROKONTROLER
ARM STM32F4**

**Disusun Sebagai Salah Satu Syarat Memperoleh Gelar Sarjana Teknik
Program Studi Teknik Elektro Jenjang S-1 Fakultas Teknik
Universitas Muhammadiyah Gresik**



Disusun Oleh :

Nama : Muhammad Shalakhudin Alyazidi

No. Reg : 11.632.007

**PROGRAM STUDI TEKNIK ELEKTRO FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH GRESIK**

2018

LEMBAR PERSETUJUAN SKRIPSI

**PEMBUANGAN AIR LIMBAH PUPUK BERBASIS MIKROKONTROLER
ARM STM32F4**

Disusun Oleh :

Nama : Muhammad Shalakhudin Alyazidi

No. Reg : 11.632.007

Gresik, 4 Januari 2018

**Mengetahui,
Ketua Program Studi**

**Menyetujui,
Dosen Pembimbing I,**

Pressa Perdana S.S.,S.T.,M.T.

NIP. 06311503179

Misbah, S.T., M.T.

NIP. 197606282005011001

**PROGRAM STUDI TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH
GRESIK
2018**

PENGESAHAN SKRIPSI

**PEMBUANGAN AIR LIMBAH PUPUK BERBASIS MIKROKONTROLER
ARM STM32F4**



Disusun Oleh :
Nama : Muhammad Shalakhudin Alyazidi
No. Reg : 11.632.007

Telah di pertahankan di depan Tim Penguji
Pada tanggal 28 Desember 2017

Dosen Penguji I,

Dosen Penguji II,

Rini Puji Astutik, S.T., M.T.
NIP.06330409134

Denny Irawan, S.T., M.T.
NIP.06330409138

Dosen Penguji III,

Dosen Pembimbing I,

Pressa Perdana S.S., S.T.,M.T.
NIP.06311503179

Misbah, S.T.,M.T.
NIP.197606282005011001

Mengetahui,
Ketua Program Studi

Mengetahui,
Dekan Fakultas Teknik

Pressa Perdana S.S., S.T.,M.T.
NIP.06311503179

Pregiwati Pusporini, S.T., M.T., Ph.D
NIP.197005032005012002

ABSTRACT

This final project presents an integrated and automated tool in waste monitoring at PT Petrokimia Gresik which includes pH measurement, pond level altitude, appropriate pH level setting and wastewater disposal. . The measurement results will be processed by the micro controller and will be sent via Bluetooth to the room control room that will be displayed on the computer.

The author will simulate a micro controller based fertilizer disposal device. Such devices include pH and water level control sensors. The pH sensor used for monitoring the pH level of waste water. The pH standard used in wastewater is 6.5-9, because at that pH the fish can live (Rivai, et al., 2010). Water level control is a device to control waste disposal when the level of waste water in ponds is high.

The purpose of this thesis research is to make microcontroller application tool for monitoring and regulation of pH level along with disposal of waste water to the sea which is tested in prototype form.

Keywords: waste, monitoring, microcontroller

KATA PENGANTAR



Assalaamu'alaikum Wr. Wb.

Alhamdulillah kami panjatkan kehadiran Allah SWT, yang telah melimpahkan rahmat, hidayah dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul :

“PEMBUANGAN AIR LIMBAH PUPUK BERBASIS MIKROKONTROLER ARM STM32F4 ”.

Skripsi ini dilakukan untuk memenuhi beban SKS yang harus ditempuh sebagai persyaratan akademis di program studi Teknik Elektro Universitas Muhammadiyah Gresik untuk menyelesaikan program pendidikan Sarjana Srata-1 Teknik Elektro.

Dalam kesempatan ini saya sampaikan terima kasih kepada :

1. Ibu Pregiwati Pusporini, S.T., M.T., Ph.D selaku Dekan Fakultas Teknik, yang telah memberikan semangat dalam penyusunan skripsi.
2. Bapak Pressa Perdana S.S., S.T., M.T selaku Ketua Program Studi Teknik Elektro yang selalu memberi masukan dan membagi pengalaman dengan antusias untuk menyempurnakan skripsi ini.
3. Bapak Misbah, S.T., M.T selaku dosen pembimbing serta dosen wali Prodi Teknik Elektro angkatan 2011, yang telah membimbing dengan penuh kesabaran hingga selesainya skripsi ini.
4. Bapak Hendra Ari Winarno, S.T selaku pembimbing yang selalu membagi ilmu yang bermanfaat dalam rangka penyusunan skripsi ini hingga selesai.

5. Ibu Rini Puji Astutik, S.T., M.T selaku dosen penguji pertama serta Bapak Denny Irawan, S.T., M.T selaku pembimbing ke-2 yang selalu memberikan masukan yang bersifat membangun hingga menyempurnakan skripsi ini.
6. Bapak dan Ibu dosen serta seluruh staf pengajar Fakultas Teknik khususnya Fakultas Teknik Elektro Universitas Muhammadiyah Gresik atas wawasan keilmuan yang diberikan selama mengikuti pendidikan.
7. Bapak Warno Suwandi dan Ibu Zulaikhatus Sa'diyah, terima kasih atas dukungan, bimbingan, pengorbanan, kasih sayang yang tak terhingga dan doa yang tak ada henti-hentinya serta senantiasa memberikan motivasi bagi penulis untuk terus bisa bermanfaat bagi sesama.
8. Bapak Abdul Wahab dan Ibuk Sunarsih terima kasih yang sangat dalam teruntuk ibu mertua yang selalu mencurahkan doa, dan memberikan dukungan serta semangat dalam penyusunan skripsi ini hingga selesai.
9. Istriku Izzatul Ulfana yang telah mencurahkan seluruh kasih sayang serta mendampingi dengan penuh kesabaran saat penyusunan skripsi ini.
10. Putriku tercinta Azkadina Almahira Ramadhani yang telah memberikan harapan, semangat dan kebahagiaan untuk segera menyelesaikan skripsi ini.
11. Mbak Zeni Ika Sari, Mas Zuhri Ramadhan, Mas Taufan Saputra, Dek Siti Maghfirotn Ni'mah, dek Muhammad Iqbal Maulana serta dek Muhammad Syifaul Umam terimakasih banyak karena selalu memberi pertanyaan kapan lulus kuliah, sehingga saya terpacu untuk menyelesaikan skripsi ini, walau penuh perjuangan.

12. Cak Askuri, Eko, Cak Khusnan, Bang Mail, Cak Arip, Debi, Agoos, Cak Mad, serta Rekan Listrik Har II yang selalu memberikan motivasi dan dukungan dalam penyusunan skripsi ini.
13. Rekan seperjuanganku Aang, Haris, Ucup, Udin, Majid, Irawan, Angga, Rohsikul, Devit, Firman, dan teman angkatan 2011 yang tidak saya sebutkan satu persatu terima kasih atas dukungan serta motivasinya dan petualangan yang tak terlupakan.
14. Keluarga besar mahasiswa Fakultas Teknik Elektro dan seluruh civitas akademika serta semua pihak yang tidak disebutkan tetapi sangat membantu dalam proses pelaksanaan penelitian dan penyusunan skripsi ini.
15. Persegres GRESIK UNITED, yang selalu mengajarkan saya tentang arti loyalitas, pengorbanan, cinta, dan harga diri. Dan saya tidak bisa berpaling darimu (Siro Dak Ngara Mlaku Ijen Gress).

Penulis menyadari bahwa tulisan ini masih jauh dari sempurna, untuk itu penulis mengharap kritikan dan saran sebagai upaya penyempurnaan skripsi ini. Semoga penelitian ini dapat menjadi informasi yang berharga bagi khalayak umum dan dapat berguna bagi perkembangan ilmu pengetahuan terutama dalam bidang Elektro. Akhir kata penulis mengucapkan terima kasih.

Wassalaamu'alaikum Wr.Wb.

Gresik, 4 Januari 2018

Penulis

DAFTAR ISI

	Halaman
HALAMAN PENEGASAN	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN	iv
ABSTRAK	v
KATA PENGANTAR	vi
DAFTAR ISI ix
DAFTAR TABEL.....	x
DAFTAR GAMBAR	xi
DAFTAR LAMPIRAN	xii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penelitian	2
1.4 Batasan Masalah.....	2
1.5 Manfaat Penelitian	3
BAB 2 TINJAUAN PUSTAKA	4
2.1 Mikrokontroler STM32 Discovery	4
2.2 Stackable Bluetooth Shield (Master/Slave)	5
2.3 LCD(liquid crystal display).....	6
2.4 Motor DC	7
2.4.1 Prinsip Kerja Motor DC	7
2.5 Sensor pH	8
2.6 Water Level Control (WLC)	9
2.7 Standar Limbah Cair	11
BAB 3 METODE PENELITIAN.....	18
3.1 Studi Literatur	18
3.2 Perencanaan dan Perancangan	18
3.3 Perancangan Hardware	20
3.3.1 Perancangan <i>Power Supply</i> (Catu Daya)	20
3.3.2 Perancangan Rangkaian Mikrokontroler ARM STM32F4 <i>Discovery</i>	21
3.3.3 Perancangan Rangkaian Stackabke Bluetooth Shield (<i>Master/Slave</i>)	24
3.3.4 Perancangan Rangkaian LCD	24
3.3.5 Perancangan Rangkaian Motor Pompa Aquarium	26
3.3.6 Perancangan Rangkaian Sensor pH	27
3.3.7 Perancangan Rangkaian Water Level Control (WLC)	28

3.3.8 Perancangan Rangkaian Relay	30
3.4 Perancangan Software	31
3.4.1 Pemrograman Menggunakan <i>Visual Basic</i>	35
BAB 4 PENGUJIAN DAN ANALISA ALAT	39
4.1 Pengujian Mikrokontroller, Bluetooth, Sensor pH, Sensor WLC, Relay dan Motor	39
BAB 5 KESIMPULAN DAN SARAN	54
DAFTAR PUSTAKA	55
LAMPIRAN	56
SURAT PERNYATAAN	69
CURICULUM VITAE	70

DAFTAR TABEL

Tabel		Halaman
2.1	Model Baku Mutu Air Limbah Industri	13
2.2	Model Baku Mutu berdasarkan Peruntukan Badan Air Penerima (Stream Standart).....	13
2.3	Indikator Baku Mutu Air Limbah untuk Industri	15
4.1	Hasil Nilai pH Buffer dan Sensor pH.....	42
4.2	Hasil Pengujian Bluetooth	43
4.3	Hasil Pengujian Jarak Bluetooth HC-05.....	44
4.4	Hasil Tegangan pada Output	48
4.5	Hasil Percobaan Relay 1 dan 2 serta Pompa Asam dan Basa	49

DAFTAR GAMBAR

Gambar	Halaman
2.1 Bentuk Fisik STM32 Discovery	4
2.2 Bentuk Fisik Bluetooth Shield (Master/Slave)	5
2.3 Bentuk Fisik LCD Display 16X2 M1632.....	6
2.4 Bentuk Fisik Konstruksi Motor DC	7
2.5 Bentuk Fisik Reed Switch	10
2.6 Gambar Rangkaian Water Level Control	10
3.1 Alur Kerja Sistem	19
3.2 Rangkaian Power Supply 545043	21
3.3 Blok Diagram Mikrokontroler ARM STM32F4	22
3.4 Skematik ARM STM32F4	23
3.5 Rangkaian Stackable Bluetooth Shield	24
3.6 Rangkaian LCD 2x16	26
3.7 Rangkaian Sensor pH	28
3.8 Bentuk Fisik Reed Switch	29
3.9 Gambar Rangkaian Water Level Control	29
3.10 Konstruksi Relay	30
3.11 Rangkaian Relay serta Motor	31
3.12 Flowchart Alur Kerja Sistem Monitoring dan Pengaturan Kadar pH beserta Pembuangan Air Limbah ke Laut Berbasis Mikrokontroler ARM32F4	33
3.13 Compiler Cocox CoIDE	34
3.14 Tampilan Awal Visual Basic	35
3.15 Desain Tampilan Visual Basic	36
4.1 Tampilan LCD 16x2 M1632	41
4.2 Blok Diagram Mikrokontroler.....	47
4.3 Tampilan Monitor Kadar pH pada PC.....	50

DAFTAR LAMPIRAN

Tabel	Halaman
Lampiran 1. Source Code Mikrokontroler ARM-STM32F4	56
Lampiran 2. Source Code Visual Basic.....	66
Lampiran 3. Gambar Tampilan Monitor Kadar pH pada LCD	68
Lampiran 4. Gambar Tampilan Hardware Monitor Kadar pH.....	68
Lampiran 5. Data Sheet Mikrokontroler ARM-STM32F4.....	69
Lampiran 6. Data Sheet pH Meter.....	91
Lampiran 7. Data Sheet Bluetooth HC-05	93
Lampiran 8. Data Sheet LCD 16x2	98
Lampiran 9. Data Sheet Relay Omron G5LA	106

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Semakin kompleks kebutuhan manusia akan menyebabkan semakin tinggi pula kebutuhan manusia akan teknologi. Kemajuan teknologi telah membuat aktifitas menjadi lebih mudah. Teknologi tidak hanya mempermudah kegiatan manusia namun dapat menggantikan kerja manusia dalam fungsi tertentu. Teknologi memegang peranan yang sangat penting di era modernisasi seperti sekarang ini. Teknologi telah menjadi bagian yang tidak dapat dipisahkan dalam kehidupan manusia sehari-hari. Perkembangan teknologi sangatlah cepat dan merambah berbagai aspek kehidupan. Banyak alat hasil teknologi yang dapat memanjakan kehidupan manusia yang dapat memberikan kemudahan.

Terbatasnya tenaga operator produksi dan tingginya volume pekerjaan dalam pembuatan produk usaha di PT Petrokimia Gresik, menyebabkan terbengkalainya pekerjaan yang secara tidak langsung berhubungan dengan sistem produksi diantaranya masalah pembuangan limbah pembuatan pupuk yang meliputi pembuangan air limbah ketika level kolam tinggi, monitoring kadar pH air limbah, pengaturan kadar pH yang sesuai standar. Standar pH yang digunakan pada limbah cair adalah 6,5-9, karena pada pH tersebut ikan dapat hidup (Rivai, dkk.,2010). Ikan menjadi salah satu indikator air limbah yang layak dibuang ke laut tanpa menimbulkan efek pada lingkungan. Pembuangan air limbah perlu diperhatikan oleh perusahaan karena berhubungan dengan kelangsungan ekosistem lingkungan sekitar perusahaan.

Berdasarkan permasalahan diatas, diperlukan suatu alat yang terintegritas

dan otomatis agar dapat menggantikan peran manusia dan mempermudah dalam pekerjaan. Hasil dari pengukuran pH, ketinggian level kolam, pengaturan kadar pH yang sesuai dan pembuangan air limbah akan diolah oleh mikrokontroler dan akan dikirim melalui Bluetooth ke ruang control room dan ditampilkan di komputer.

Pembuatan system ini diharapkan mampu memberikan kemudahan dalam monitoring limbah dan dapat menggantikan tenaga pekerja sehingga dapat fokus pada tugas pekerjaan yang utama.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang diatas, maka rumusan masalah yang dapat dibuat adalah bagaimanakah aplikasi dan pembuatan alat yang terintegritas dari monitoring kadar pH sampai air limbah siap dibuang ke laut dalam mikrokontroller?

1.3 Tujuan

Membuat alat aplikasi mikrokontroller untuk monitoring dan pengaturan kadar pH beserta pembuangan air limbah ke laut yang diujikan dalm bentuk prototipe.

1.4 Batasan Masalah

Pada penulisan tugas akhir ini permasalahan dibatasi sebagai berikut:

1. Pengujian dilakukan dengan pembuatan prototipe
2. Program simulasi menggunakan code vision AVR dan visual basic (VB).

1.5 Manfaat Penelitian

Manfaat penelitian ini adalah:

1. Peralatan otomatis ini diharapkan memberikan kontribusi di perusahaan
2. Efisiensi tenaga dan waktu
3. Menambah wawasan dan bahan rujukan untuk penulis berikutnya

BAB II

TINJAUAN PUSTAKA

2.1 Mikrokontroler STM32 Discovery

STM32 Discovery merupakan suatu development board dengan mikrokontroler ARM Cortex M4 yang memiliki kecepatan sampai 168 MHz. Mikrokontroler ini memiliki 1 MByte Flash PEROM, 192 Kbyte SRAM.

Berikut merupakan gambar mikrokontroler STM32 discovery (M.Rizhal, 2010):



Gambar 2.1 STM32 Discovery

Mikrokontroler STM32 memiliki antarmuka kamera yang dapat terhubung dengan kamera melalui 8 bit sampai 14 bit antarmuka parallel. Gambar yang diterima oleh mikrokontroler disimpan dalam RAM dengan menggunakan DMA (Direct Memory Acces). DMA merupakan suatu teknik perpindahan data dari suatu alamat memori ke alamat memori yang lain tanpa mengganggu kerja dari mikroprosesor (Lian, 2014).

2.2 Stackable Bluetooth Shield (Master/Slave)

Stackable *Bluetooth Shield (Master/Slave)* merupakan salah satu modul *bluetooth* yang dikembangkan untuk dapat digunakan pada aplikasi mikrokontroler khususnya pada arduino. *Stackable Bluetooth Shield* yang umum

ditemukan di pasaran ada dua jenis yaitu *Stackable Bluetooth Shield (Master/Slave)* dan *Stackable Bluetooth Shield (Slave)*. Perbedaan *bluetooth shield* ini terdapat pada fungsinya, *Stackable Bluetooth Shield (Master/Slave)* dapat digunakan sebagai *pairing* data dua arah, sebagai *transmitter* dan sebagai *receiver* sedangkan *Stackable Bluetooth Shield (Slave)* hanya digunakan sebagai *pairing* data *receiver* saja. Bentuk fisik dari *Stackable Bluetooth Shield (Master/Slave)* nampak pada gambar 1 berikut ini: 2.4 pH Meter. (*Eko Kristanto,2012*).

Bentuk Fisik *Stackable Bluetooth Shield (Master/Slave)* ditampilkan pada gambar 2.2 (<http://cdn2.bigcommerce.com>):



Gambar 2.2 Bentuk Fisik *Stackable Bluetooth Shield (Master/Slave)*.

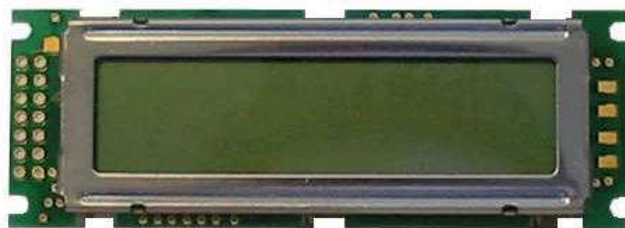
2.3 LCD

LCD adalah suatu jenis media tampilan yang menggunakan kristal cair sebagai penampil utama. LCD sudah digunakan di berbagai bidang misalnya dalam alat-alat elektronik seperti televisi, kalkulator, ataupun layar komputer.

Pada LCD berwarna semacam monitor terdapat banyak sekali titik cahaya yang terdiri dari satu buah kristal cair sebagai sebuah titik cahaya. Walau disebut sebagai titik cahaya, namun kristal cair ini tidak dapat memancarkan cahaya sendiri. Sumber cahaya didalam sebuah perangkat LCD adalah lampu neon berwarna putih di bagian belakang susunan kristal cair tadi. Titik cahaya yang jumlahnya puluhan ribu bahkan jutaan inilah yang membentuk tampilan citra. Kutub kristal cair yang dilewati arus listrik akan berubah karena pengaruh polarisasi medan magnetik yang timbul dan oleh karenanya hanya akan membiarkan beberapa warna diteruskan sedangkan warna lainnya tersaring (Putri, 2014). LCD ini mempunyai keunggulan antara lain adanya panel pengatur kekontrasan cahaya tampilan LCD, tampilan terdiri dari 2 baris yang masing-masing terdiri 16 karakter, selain itu LCD ini membutuhkan konsumsi daya yang rendah. (<http://delta-electronic.com>).

Bentuk fisik LCD Display 16X2 M1632 ditampilkan pada gambar 2.3 (Ari Heryanto, M dan Wisnu Adi P, 2008).

:



Gambar 2.3. LCD Display 16X2 M1632

2.4 Motor DC

Motor DC memiliki dua elemen listrik yang terdiri dari bagian stator (kumparan medan utama) dan rotor (kumparan jangkar). Kumparan jangkar

mengalirkan arus yang berasal dari komutator. Pada saat tegangan dialirkan ke kumparan medan di stator dengan kutub utara dan selatan buatan (electromagnet), akan dihasilkan medan magnet statis (bukan medan magnet putar). Untuk menjelaskanya, stator akan dianggap sebagai magnet permanen.

Motor DC berputar sebagai akibat adanya dua medan magnet yang saling berinteraksi satu dengan yang lainnya. Medan pertama adalah medan magnet utama yang berada pada kumparan stator dan medan kedua adalah medan magnet yang berada pada jangkar.

Bentuk fisik dari konstruksi Motor DC ditampilkan pada gambar 2.4 (Ari Heryanto, M dan Wisnu Adi P, 2008).

:



Gambar 2.4 Konstruksi Motor DC

2.4.1 Prinsip Kerja Motor DC

Pada motor DC, kumparan medan yang dialiri arus listrik akan menghasilkan medan magnet yang melingkupi kumparan jangkar dengan arah tertentu. Konverter energi baik energi listrik menjadi energy mekanik (motor) maupun sebaliknya dari energi mekanik menjadi energi listrik (generator) berlangsung melalui medium medan magnet. Energi yang akan diubah dari suatu sistem ke sistem yang lain, sementara akan tersimpan pada medium medan

magnet untuk kemudian dilepaskan menjadi energi sistem lainnya. Dengan demikian, medan magnet disini selain berfungsi sebagai tempat penyimpanan energi juga sekaligus proses perubahan energy, dimana proses perubahan energi pada motor searah.

Motor DC terdapat dalam berbagai ukuran dan kekuatan, masing-masing didesain untuk keperluan yang berbeda-beda namun secara umum memiliki fungsi dasar yang sama yaitu mengubah energi elektrik menjadi energi mekanik. Sebuah motor DC sederhana dibangun dengan menempatkan kawat yang dialiri arus di dalam medan magnet. Kawat yang membentuk loop ditempatkan sedemikian rupa diantara dua buah magnet permanen. Bila arus mengalir pada kawat, arus akan menghasilkan medan magnet sendiri yang arahnya berubah-ubah terhadap arah medan magnet permanen sehingga menimbulkan putaran.

2.5 Sensor pH

Sensor adalah komponen elektronika yang digunakan untuk merubah besaran fisik menjadi besaran listrik sehingga bisa di analisa dengan menggunakan rangkaian listrik. Pada prinsipnya sistem sensor pH (pH meter) terdiri dari elektroda pH yang digunakan untuk mendeteksi banyaknya ion H^+ dari suatu cairan. Pengukuran pH dilakukan dengan menggunakan elektroda potensiometrik. Elektroda ini memonitor perubahan voltase yang disebabkan oleh perubahan aktivitas ion hidrogen (H^+) dalam larutan. Elektroda pH yang paling modern terdiri dari kombinasi tunggal elektroda referensi (reference electrode) dan elektroda sensor (sensing electrode) yang lebih mudah dan lebih murah daripada elektroda terpisah. Elektroda kombinasi ini mempunyai fungsi yang sama

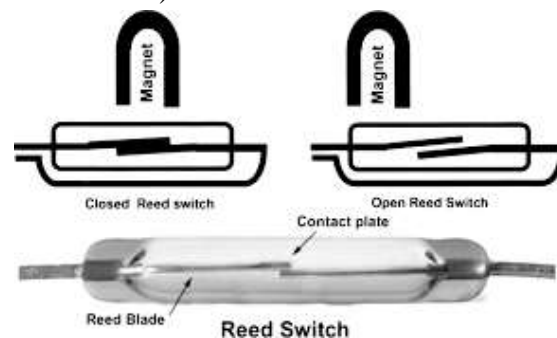
dengan elektroda pasangan. Keluaran dari pH meter sudah dikalibrasi dalam mV dan kondisi ideal dari elektroda pH pada suhu 25°C. Dengan memonitor perubahan tegangan yang disebabkan oleh perubahan aktifitas ion hidrogen (H⁺) dalam larutan maka pH larutan dapat diketahui. Tegangan keluaran dari elektroda akan menunjukkan 0mV ketika dipakai untuk mengukur pH 7.00 (netral) (Rivai,dkk.,2010)

Standar pH yang digunakan adalah 6,5-9, karena pada pH tersebut ikan dapat hidup (Rivai, dkk.,2010). Ikan menjadi salah satu indikator air limbah yang layak dibuang ke laut tanpa menimbulkan efek pada lingkungan

2.6 Water Level Control (WLC)

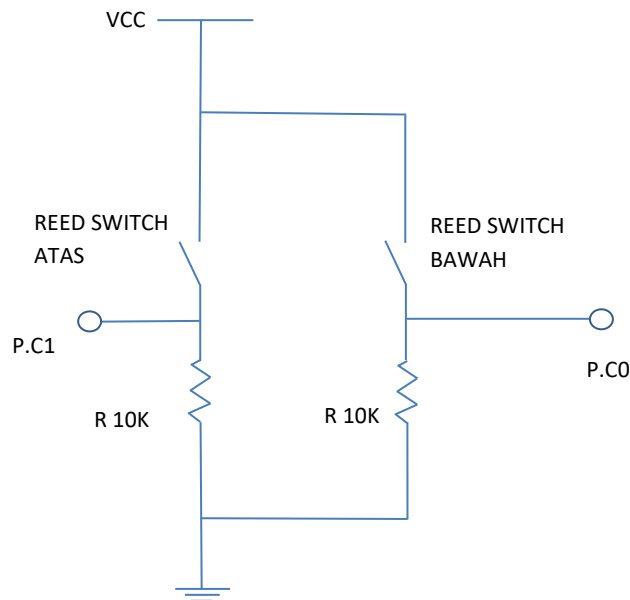
Water Level Control (WLC) adalah alat yang berfungsi untuk mengontrol ketinggian level air, terdapat banyak komponen elektronik yang dapat digunakan untuk membuat WLC, dan salah satunya menggunakan reed swith dan magnet. Ketika magnet yang digerakkan oleh ketinggian level air, menyentuh reedswitch maka reedswitch akan berlogika 1:

Berikut adalah rangkaian water level control ditampilkan pada gambar 2.5 (dunia elektronika.com)



Gambar 2.5 Reed Switch
(sumber:dunia elektronika.com)

Berikut adalah rangkaian water level control disajikan pada gambar 2.6:



Gambar 2.6 Gambar Rangkaian Water Level Control

2.7 Standar Limbah Cair

Limbah Cair

a. Pengertian Limbah

Limbah adalah hasil samping dari proses produksi yang tidak akan digunakan, dapat berbentuk padat, cair, gas, suara, dan getaran yang dapat menimbulkan pencemaran apabila tidak dikelola dengan benar (Winarno, 1992).

Limbah merupakan suatu bahan yang terbuang atau yang dibuang dari hasil aktivitas manusia maupun proses alam yang tidak atau belum mempunyai nilai ekonomis, bahkan dapat mempunyai nilai negatif karena penanganan untuk membuang atau membersihkan membutuhkan biaya yang cukup besar, disamping itu juga dapat mencemari lingkungan (Mahida, 1992).

Sedangkan menurut Murthado dan Said (1987), limbah pada dasarnya berarti suatu bahan yang terbuang atau dibuang dari suatu sumber hasil aktivitas manusia, maupun proses-proses alam, dan tidak atau belum mempunyai nilai ekonomi, bahkan dapat mempunyai nilai ekonomi yang negatif. Limbah dikatakan mempunyai nilai ekonomi yang negatif karena penanganan untuk membuang atau membersihkannya memerlukan biaya yang cukup besar, disamping itu juga dapat mencemari lingkungan.

Dalam rangka mengendalikan pencemaran air limbah oleh pelaku usaha, pemerintah pusat dan daerah telah menetapkan berbagai peraturan yang berkaitan dengan kualitas air limbah, debit air limbah, dan beban maksimum air limbah yang diperbolehkan untuk dibuang ke badan air. Peraturan tersebut dikenal dengan peraturan *BAKU MUTU AIR LIMBAH INDUSTRI*.

Penetapan baku mutu air limbah didasarkan pada dua (2) aspek yaitu

1. Berdasarkan air limbah yang dihasilkan oleh setiap industri disebut sebagai standar air limbah (*Effluent Standard*).
2. Berdasarkan peruntukan dari badan air penerima disebut sebagai standar air badan penerima (*Stream Standard*).

Dalam penentuan baku mutu air limbah diperkenalkan berbagai istilah diantaranya

Limbah cair, merupakan limbah dalam bentuk cair yang dihasilkan suatu aktifitas yang dibuang ke lingkungan hidup dan diduga dapat menurunkan kualitas lingkungan hidup

Baku mutu air limbah, adalah batas maksimum limbah cair yang diperbolehkan dibuang ke lingkungan

Mutu air limbah, merupakan keadaan air limbah yang dinyatakan dengan debit, kadar dan beban pencemar.

Debit maksimum, merupakan debit tertinggi yang masih diperbolehkan dibuang ke lingkungan hidup

Kadar maksimum, merupakan kadar tertinggi yang masih diperbolehkan dibuang ke lingkungan hidup

Beban pencemaran maksimum, merupakan beban pencemaran tertinggi yang masih diperbolehkan dibuang ke lingkungan hidup

Baku mutu air limbah berdasarkan standar air limbah (Effluent Standard) seperti tercantum dalam tabel 2.1 berikut ini.

Tabel 2.1. Model Baku Mutu Air Limbah Industri

Parameter	Kadar Maksimum (mg/L)	Beban Pencemaran Maksimum (gram/satuan produk)
BOD ₅	75	22,5
COD	125	37,5
TSS	50	15
Fenol	0,25	0,08
Amonia total (sebagai N)	4	1,2
pH	6-9	
Debit limbah maksimum	0,3 m ³ / satuan produk	

Perhitungan beban pencemaran maksimum dengan persamaan berikut

$$\text{BPM} = (\text{Cm})_j \times \text{Dm} \times f$$

Keterangan :

BPM : Beban Pencemaran maksimum yang diperbolehkan, dinyatakan dalam kg parameter per hari.

(Cm)_j : Kadar maksimum parameter j dinyatakan dalam mg/l.

Dm : Debit Limbah cair maksimum dinyatakan dalam L limbah cair per detik

f : Faktor konversi [1m³ x {mg/L} x (1/1000 kg)]

Baku mutu air limbah berdasarkan standar peruntukan badan air penerima (Stream Standard) seperti tercantum dalam tabel 2.2 berikut ini

Tabel 2.2. Model Baku Mutu berdasarkan Peruntukan Badan Air Penerima (Stream Standard)

BAKU MUTU AIR LIMBAH CAIR					
(TERMASUK PENGOLAH LIMBAH CAIR TERPUSAT (KAWASAN))					
Parameter	Satuan	Golongan Baku Mutu Limbah Cair			
		(Golongan Sungai)			
		I	II	III	IV
A. FISIKA					
Temperatur	°C	35	38	40	45
Zat padat terlarut	mg/L	1500	2000	4000	5000
Zat padat tersuspensi	mg/L	100	200	200	500
B. KIMIA					
pH		6-9	6-9	6-9	6-9
Besi (Fe)	mg/L	5	10	15	20
Mangan (Mn)	mg/L	0,5	2	5	10
Barium (Ba)	mg/L	1	2	3	5
Tembaga (Cu)	mg/L	1	2	3	5
Seng (Zn)	mg/L	5	10	15	20

Lanjutan Tabel 2.2

Parameter	Satuan	Golongan Baku Mutu Limbah Cair (Golongan Sungai)			
		I	II	III	IV
Kromium total (Cr tot)	mg/L	0,1	0,5	1	2
Kadmium (Cd)	mg/L	0,01	0,05	0,1	1
Raksa (Hg)	mg/L	0,001	0,002	0,005	0,01
Timbal (Pb)	mg/L	0,1	0,5	1	3
Timah putih (Sn)	mg/L	2	3	4	5
Arsen (As)	mg/L	0,05	0,1	0,5	1
Kobal (Co)	mg/L	0,2	0,4	0,6	1
Sianida (CN)	mg/L	0,05	0,1	0,5	1
Sulfida (H ₂ S)	mg/L	0,01	0,06	0,1	1
Flourida (F)	mg/L	1,5	15	20	30
Klorin bebas (Cl ₂)	mg/L	0,02	0,03	0,04	0,05
Amonia bebas (NH ₃ -N)	mg/L	0,5	1	5	20
Nitrat (NO ₃ -N)	mg/L	10	20	30	50
Nitrit (NO ₂ -N)	mg/L	0,06	1	3	5
BOD ₅	mg/L	30	50	150	300
COD	mg/L	80	100	300	600
Detergen anionik	mg/L	0,5	1	10	15
Phenol	mg/L	0,01	0,05	1	2
Minyak & Lemak	mg/L	1	5	15	20
PCB	mg/L	NIHIL	NIHIL	NIHIL	NIHIL
Selenium (Se)	mg/L	0,01	0,05	0,1	1
Nikel (Ni)	mg/L	0,1	0,2	0,5	1

Lanjutan Tabel 2.2

Parameter	Satuan	Golongan Baku Mutu Limbah Cair (Golongan Sungai)			
		I	II	III	IV
C. BIOLOGI					
Coliform Group	MPN / 100 ml	100	1000	2000	2000
Coliform tinja	MPN / 100 ml	1000	5000	10000	10000

Ø Baku mutu air limbah umumnya akan mengalami peninjauan setelah lima (5) tahun.

Berbagai jenis industri mempunyai indikator baku mutu air limbah yang berbeda-beda seperti tercantum dalam tabel 2.3.

Tabel 2.3. Indikator byaku mutu air limbah untuk industri

No	Jenis Industri	Indikator Baku Mutu
1	Industri Soda Kostik	COD, TSS, Raksa (Hg), Timbal (Pb); Tembaga (Cu); Seng (Zn), pH
2	Industri Pelapisan Logam (pelapisan Tembaga, Nikel, Galvanisasi Seng)	TSS, Kadmium (Cd), Sianida (CN); Logam Total, Tembaga (Cu), Nikel (Ni), Krom Total (Cr); Krom Heksavalen (Cr+6)
3	Industri Penyamakan Kulit	BOD5, COD, TSS, Sulfida (sebagai H2S); Krom Total (Cr); Minyak dan Lemak, Amonia Total, pH
4	Industri Minyak Sawit	BOD5, COD, TSS, Minyak dan Lemak; AmoniaTotal (sebagai NH3-N); pH

No	Jenis Industri	Indikator Baku Mutu
5	Industri Pulp dan Kertas	BOD5, COD; TSS; pH
6	Industri Karet	BOD5, COD; TSS; Amonia Total (sebagai NH3-N); pH
7	Industri Gula	BOD5; COD; TSS; Sulfida (sebagai H2S); pH
8	Industri Tapioka	BOD5; COD; TSS; Sianida (CN); pH
9	Industri Tekstil	BOD5; COD; TSS; Fenol Total; Krom Total (Cr); Minyak dan Lemak; pH
10	Industri Pupuk Area	BOD5; COD; TSS; Minyak dan Lemak; pH
11	Industri Mono Sodium Glutamate (MSG)	BOD5; COD; TSS; pH
12	Industri Kayu Lapis	BOD5; COD; TSS; Fenol Total; pH
13	Industri susu dan makanan yang terbuat dari susu	BOD5; COD; TSS; pH
14	Industri Minuman Ringan	BOD5; COD; TSS; Minyak dan Lemak; pH
15	Industri Sabun, Deterjen dan produk-produk Minyak Nabati	BOD5; COD; TSS; Minyak dan Lemak; Fosfat (sebagai PO4); MBAS, pH
16	Industri Bir	BOD5; COD; TSS; pH
17	Industri Baterai Kering	COD; TSS; NH3-N Total; Minyak dan Lemak, Seng (Zn); Merkuri (Hg), Mangan (Mn), Krom (Cr); Nikel (Ni); pH
18	Industri Cat	BOD5; TSS; Merkuri (Hg), Seng (Zn), Timbal (Pb); Tembaga (Cu); Krom Heksavalen (Cr+6); Titanium (Ti), Kadmium (Cd), Fenol; Minyak dan lemak

Lanjutan Tabel 2.3

No	Jenis Industri	Indikator Baku Mutu
19	Industri Farmasi	BOD5, COD, TSS, Total-N, Fenol, pH
20	Industri Pestisida	BOD5, COD, TSS, Fenol, Total-CN, Tembaga(Cn), Krom Aktif Total, pH
21	Hotel	BOD5, COD, TSS, pH
22	Kegiatan Rumah Sakit	BOD5, COD, TSS, pH, Mikrobiologik (golongankoli); dapat ditambahkan radioaktivitas
23	Limbah Rumah Tangga(Domestik)	BOD5, COD, TSS, pH, Deterjen, Mikrobiologik(golongan koli)

BAB III

METODOLOGI PENELITIAN

3.1. Studi Literatur

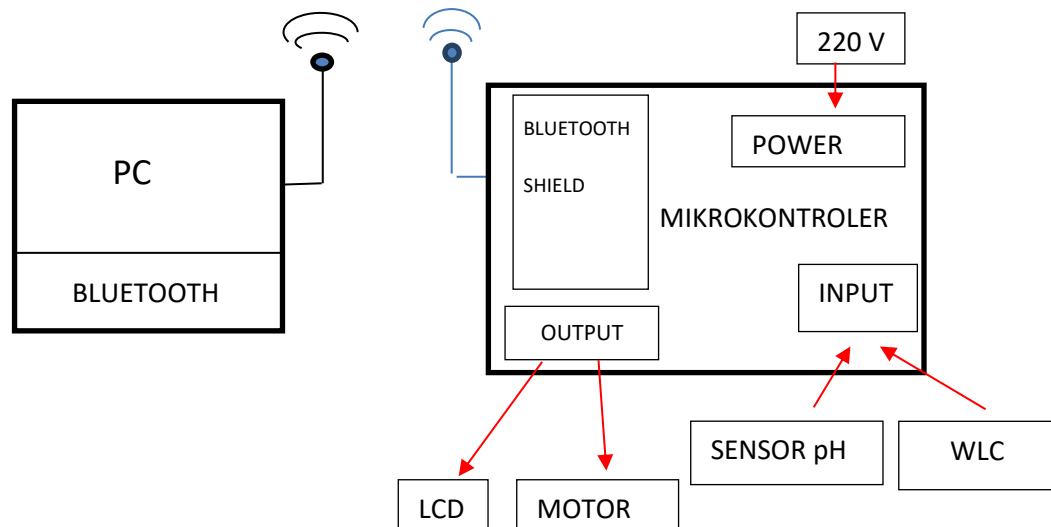
Dalam perancangan sistem monitoring dan pengaturan kadar pH beserta pembuangan air limbah ke laut berbasis mikrokontroler ARM dibutuhkan sumber-sumber referensi sebagai bahan acuan dan pertimbangan. Sumber referensi didapatkan dari sumber langsung dan tak langsung. Sumber langsung didapat dari hasil diskusi atau konsultasi dengan dosen, sedangkan sumber tak langsung didapat dari tulisan laporan penelitian-penelitian yang dilakukan sebelumnya, buku, internet serta referensi-referensi lain yang berkaitan dengan perancangan dan pembuatan alat.

3.2. Perencanaan dan perancangan

Sistem monitoring dan pengaturan kadar pH beserta pembuangan air limbah ke laut berbasis mikrokontroler ARM, secara garis besar terdiri dari perangkat keras (*hardware*) dan perangkat lunak (*software*). Perangkat keras terdiri dari rangkaian mikrokontroler ARM menggunakan STM32F4 *Discovery* yang memiliki banyak keunggulan dibanding jenis mikrokontroler lain dengan harga yang relatif terjangkau. Pada skripsi ini dirancang alat yang dapat memonitor kadar pH air limbah dan ketinggian level kolam sampai air limbah terbuang ke laut. Sistem ini menggunakan teknologi Bluetooth dalam komunikasi data dan ditampilkan pada PC, dan inputan dari mikrokontroler menggunakan sensor pH dan WLC.

Dalam mengoperasikan atau menjalankan alat monitoring dan pengaturan kadar pH beserta pembuangan air limbah ke laut berbasis mikrokontroler ARM,

dibantu dengan implementasi program sebagai alur kerja sistem. Gambar kerja sistem ditampilkan pada gambar 3.1



Gambar 3.1 Alur Kerja Sistem

Berdasarkan alur kerja sistem diatas, cara kerjanya adalah yaitu sensor pH akan membaca besaran nilai limbah, apabila nilai limbah tersebut kurang dari 6,5 disebut cairan asam maka motor 1 akan bekerja dan menambah cairan basa sampai nilai besaran limbah sampai 7, dan apabila nilai limbah lebih dari 7,5 disebut cairan basa maka motor 2 akan bekerja dan menambah cairan asam sampai nilai besaran limbah sebesar 7. Motor 3 bekerja apabila level kolam tinggi dan mati sampai level kolam rendah. Semua data dan penunjukan akan ditampilkan lewat LCD di kolam local serta akan dikirim melalui Bluetooth dan dimonitor oleh PC diruangan control room

Berdasarkan gambar 3.1, bagian-bagian yang dibutuhkan untuk membuat alat ini adalah sebagai berikut :

1. Rangkaian pengendali untuk mengatur semua proses kerja alat menggunakan sistem minimum Mikrokontroller ARM STM32F4 Discovery.
2. Catu daya 5 Volt digunakan sebagai sumber bagi mikrokontroller serta IC pendukung lainnya.
3. Sensor pH, sensor reed switch (WLC) sebagai inputan pada mikrokontroller.
4. Output sistem adalah relay, motor pompa aquarium, serta LCD.
5. Bluetooth sebagai pengirim data ke PC.
6. Papan plastik sebagai tempat mikrokontroller dan piranti elektronik lainnya

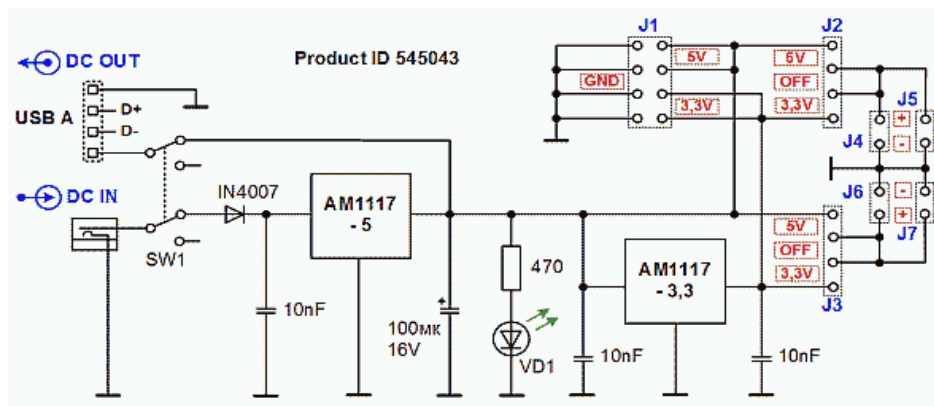
3.3. Perancangan *Hardware*

Rancangan *hardware* sistem pengaman rumah ini terbagi atas beberapa bagian yaitu bagian utama berupa mikrokontroler ARM STM32F4, bagian inputan berupa sensor pH, WLC dan bagian outputnya adalah motor pompa aquarium, LCD dan juga bluetooth yang berfungsi mengirim data yang akan ditampilkan oleh PC.

3.3.1. Perancangan *Power Supply* (Catu Daya)

Power Supply (Catu Daya) adalah sebuah piranti elektronika yang berguna sebagai sumber daya untuk piranti lain, terutama daya listrik. Dalam perancangan system pengaman rumah catu daya yang saya gunakan Rangkaian *power supply* ini berfungsi untuk mensuplai arus dan tegangan ke seluruh rangkaian yang ada. Rangkaian *power supply* ini mempunyai keluaran, yaitu keluaran 5 Volt DC. Adaptor 12 Volt DC. Kemudian di searahkan dengan menggunakan dioda IN4007,

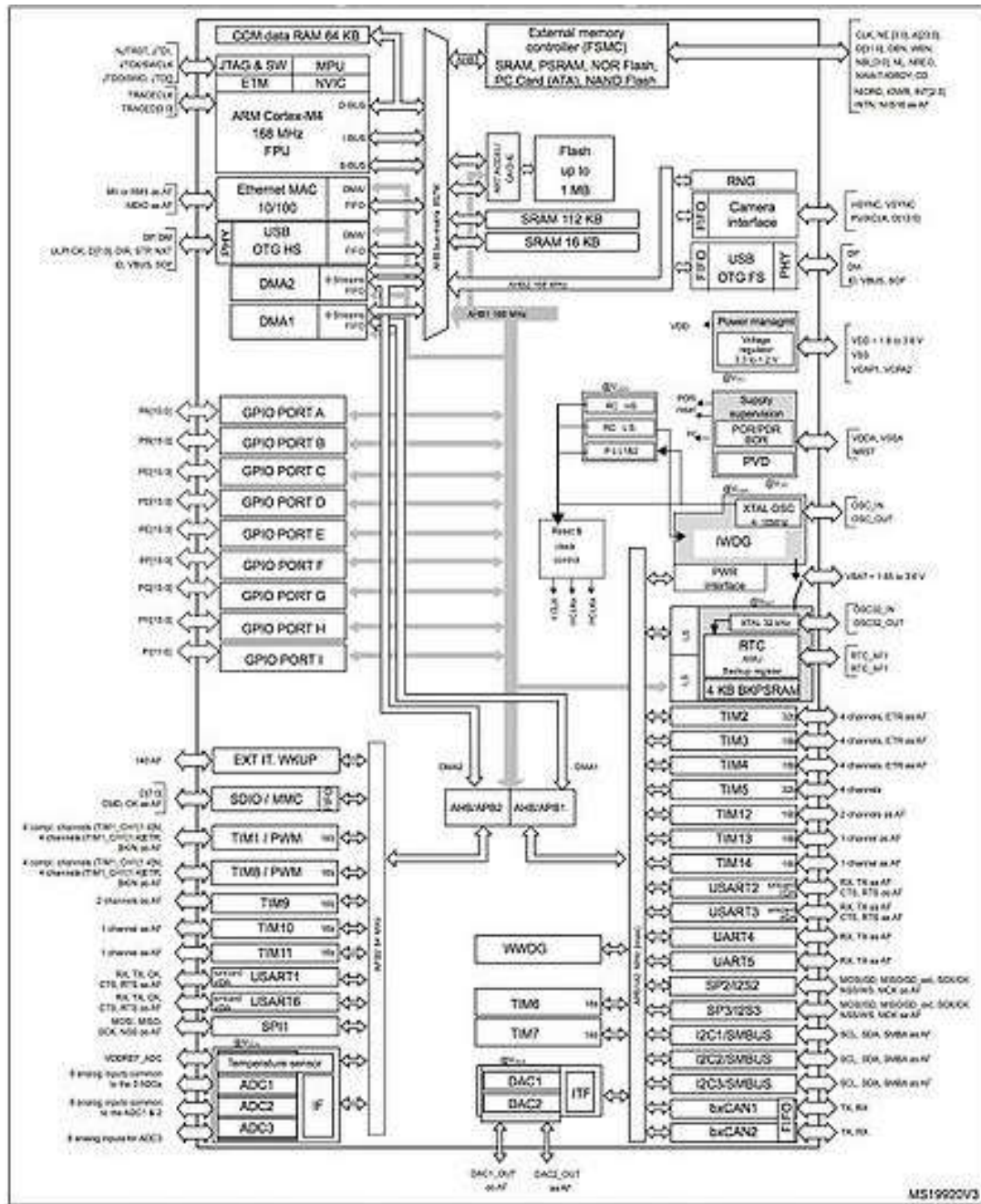
selanjutnya tegangan 12 Volt DC dihubungkan ke IC AM1117-3,3 difilter oleh elco C1 100 mk dan dihubungkan ke input tegangan mikrokontroler STM32F4. Selanjutnya tegangan 12 Volt DC tersebut dihubungkan ke IC AM1117-5 dan satu buah elco 100mk sebagai filter digunakan untuk mensuplai arus dan tegangan untuk Sensor. Rangkaian *power supply* ditampilkan pada gambar 3.2



Gambar 3.2. Rangkaian *power supply* 545043

3.3.2 Perancangan Rangkaian Mikrokontroler ARM STM32F4 *Discovery*

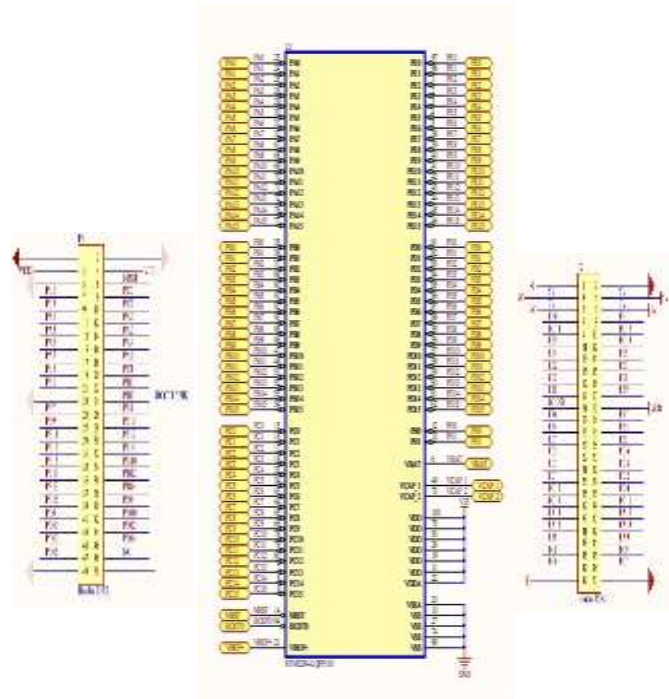
Mikrokontroler bekerja pada tegangan 5V dc yang di suplay dari USB computer ditunjukkan pada gambar 3.3



Gambar 3.3. Blok Diagram Mikrokontroler ARM STM32F4

Mikrokontroler ARM STM32F4 dalam tugas akhir ini berfungsi sebagai pengendali utama untuk memproses data. Mikrokontroler STM32F4 karena *clock* kerjanya tinggi sampai 168 MHz kecepatan maksimal, ukuran flash memorinya cukup besar yaitu sebesar 1024 Kbyte, kapasitas SRAM sebesar 192 Kbyte. Sistem

I/O yang ada pada STM32F4 sebanyak 82 I/O dimana masing-masing I/O tersebut digunakan untuk beberapa aplikasi diantaranya yaitu untuk sensor pH, WLC, LCD, relay, berikut ini gambar 3.4 adalah skematik ARM STM32F4.

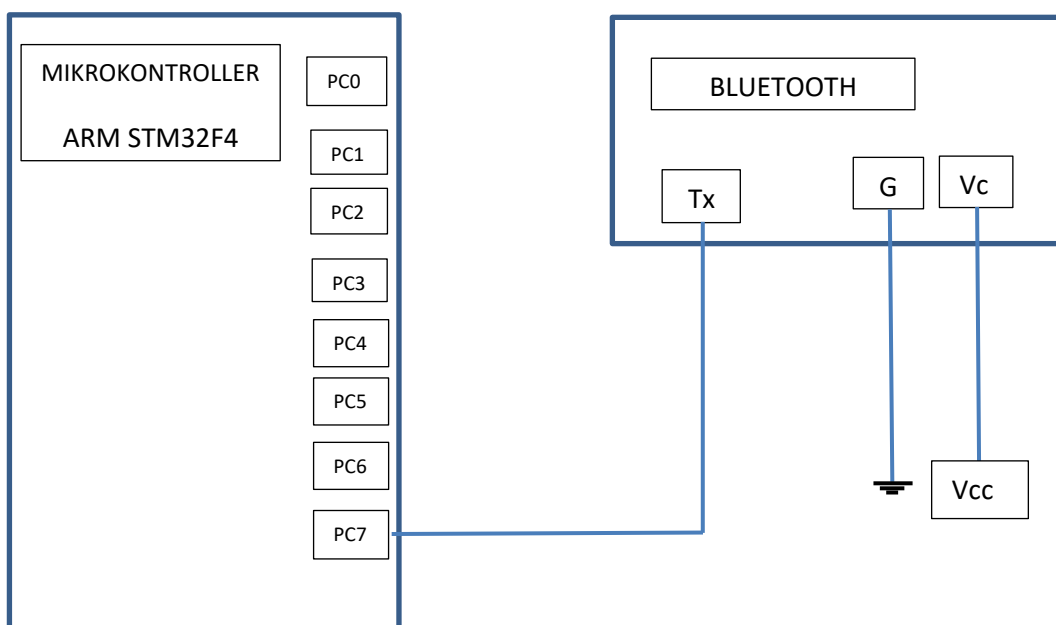


Gambar 3.4 Skematik ARM STM32F4

ARM STM32F4 input dan output yang akan digunakan untuk menjalankan system alat atau modul, dengan Pin A3 digunakan untuk inputan sensor pH. Pin B1, Pin C5 digunakan untuk inputan WLC atas dan bawah. Output nya berupa relay untuk motor 1 (pompa basah) yang dipasang pada Pin A9, motor 2 (pompa asam) yang dipasang pada Pin A13, sarta motor 3 (pompa pembuangan) yang dipasang pada Pin A15. Serta Pin C7 untuk ouputan bluetooth shield untuk komunikasi data pengiriman pada PC.

3.3.3 Perancangan Rangkaian Stackable Bluetooth Shield (Master/Slave)

Stackable Bluetooth Shield (Master/Slave) berfungsi untuk mengirim data pada PC di control room untuk kemudian di Perbedaan bluetooth shield ini terdapat pada fungsinya, Stackable Bluetooth Shield (Master/Slave) dapat digunakan sebagai pairing data dua arah, sebagai transmitter dan sebagai receiver sedangkan Stackable Bluetooth Shield (Slave) hanya digunakan sebagai pairing data receiver. Skema Stackable Bluetooth Shield (Master/Slave) ditampilkan pada gambar 3.5

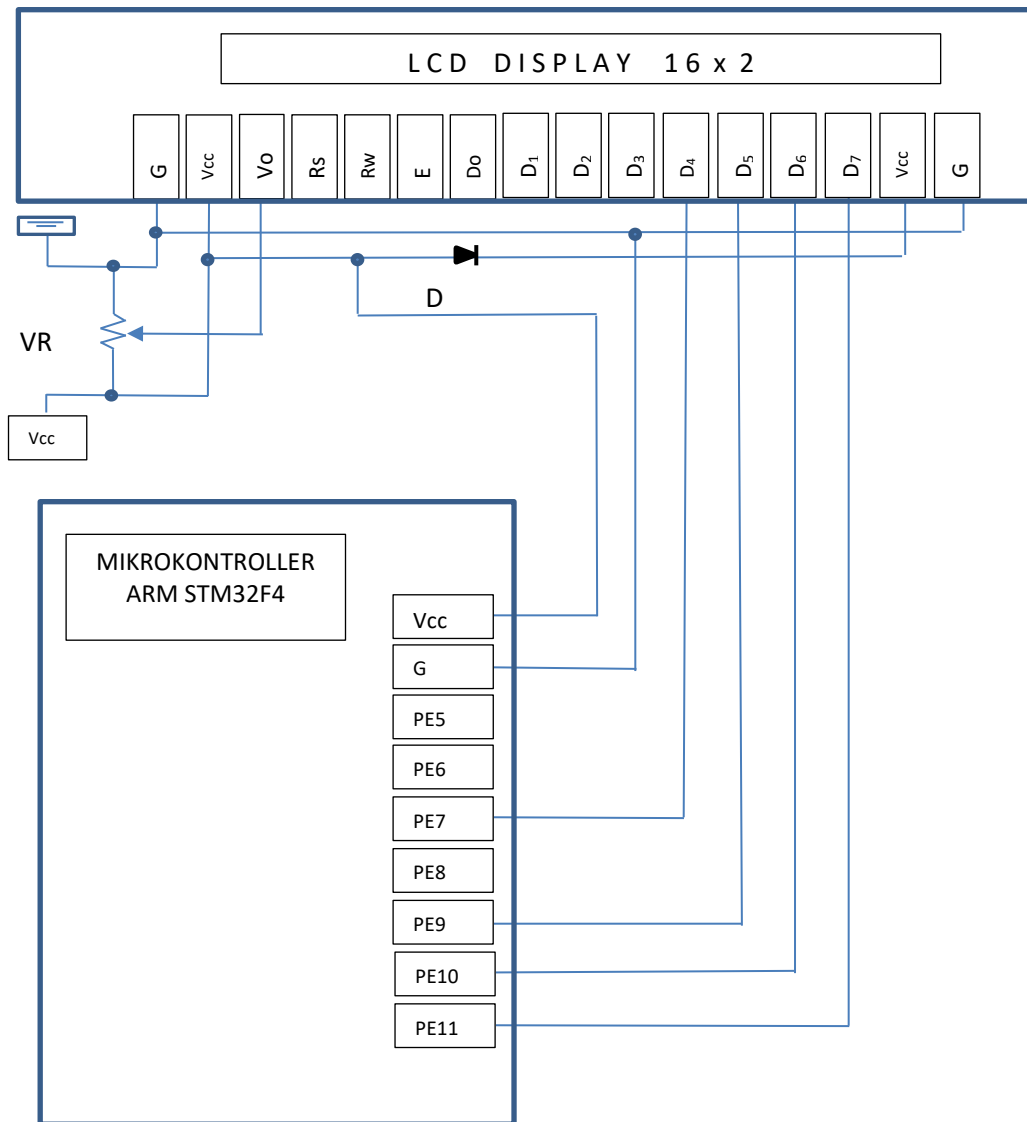


Gambar 3.5 Rangkaian Stackable Bluetooth Shield

3.3.4 Perancangan Rangkaian LCD

LCD 2x16 sangat mudah diaplikasikan pada mikrokontroler ARM STM32F4. LCD 2X16 artinya LCD tersebut didesain untuk menampilkan data sebanyak 2 baris 16 kolom. Pada gambar 4.3. merupakan rangkaian *interface* LCD dengan mikrokontroler AVR ATmega32. Pada rangkaian tersebut dapat dilihat Pin 1 dari LCD terhubung dengan *ground*, Pin 2 merupakan VCC dari LCD dihubungkan dengan pin VCC mikrokontroler ARM STM32F4. Sedangkan pin 3

berfungsi mengatur kontras LCD, sehingga pada rangkaian diatas Pin 3 terhubung dengan sebuah *varibel resistor* (VR) 10 K Ω . LCD memerlukan tiga jalur kontrol dan delapan jalur data (untuk mode 8 *bit*) atau empat jalur data (untuk mode 4 *bit*). Ketiga jalur kontrol yang dimaksud adalah Pin 4 (RS), Pin 5 (R/W), Pin 6 (E). Pin 4 (RS) merupakan Pin *register selection signal* yang berfungsi memilih *register* yang akan digunakan, Pin ini terhubung pada Pin 14 (PC.0). Untuk Pin 5 (R/W) merupakan Pin *read/write signal* yang bila diberi logika '0', akan terpilih *write*. dan bila diberi logika '1', maka akan terpilih *read*. Pada saat Pin R/W berlogika rendah, informasi pada jalur data berupa pengiriman data ke LCD (*write*). Sedangkan ketika Pin R/W berlogika tinggi, berarti sedang dilaksanakan pengambilan data dari LCD (*read*). Pin 5 LCD ini dihubungkan dengan Pin 15 (PC.1). Sedangkan Pin 6 (E) adalah Pin *Enable* yang berfungsi mengaktifkan *read/write* data, yang digunakan untuk memberitahu LCD kalau kita akan berkomunikasi dengannya. Sebelum mengirim data ke LCD Pin ini di buat berlogika tinggi dahulu. Kemudian jalur kontrol yang lain di-setting, pada saat bersamaan data yang akan dikirim ditempatkan pada jalur data. Setelah semua siap, Pin *Enable* dibuat berlogika rendah. Transisi dari logika tinggi ke logika rendah ini akan memberitahu LCD untuk mengambil data pada jalur kontrol dan jalur data. Pin *Enable* dihubungkan dengan Pin 16 (PC.2) pada mikrokontroler AVR ATmega32. Rangkaian LCD ditampilkan pada 3.6



Gambar 3.6. Rangkaian LCD 2x16

3.3.5 Perancangan Rangkaian Motor Pompa Aquarium

Pompa adalah mesin atau peralatan mekanis yang digunakan untuk menaikkan cairan dari dataran rendah ke dataran tinggi atau untuk mengalirkan cairan dari daerah bertekanan rendah ke daerah bertekanan tinggi dan juga sebagai penguat laju aliran pada suatu sistem jaringan perpipaan. Hal ini dicapai dengan membuat suatu tekanan yang rendah pada sisi masuk atau *suction* dan tekanan yang

tinggi pada sisi keluar atau *discharge* dari pompa.

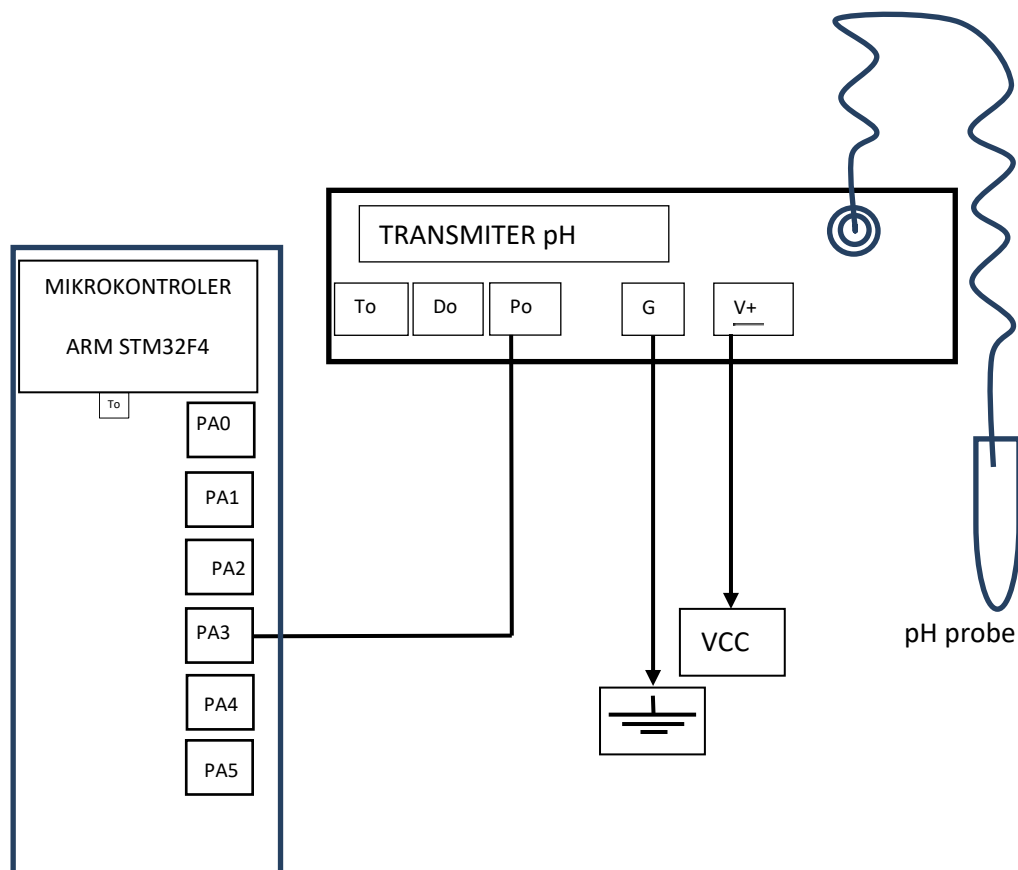
Pompa juga dapat digunakan pada proses-proses yang membutuhkan tekanan *hidraulik* yang besar. Hal ini bisa dijumpai antara lain pada peralatan-peralatan berat. Dalam operasi, mesin-mesin peralatan berat yang membutuhkan tekanan discharge yang besar dan tekanan isap yang rendah. Akibat tekanan yang rendah pada sisi isap pompa maka *fluida* akan naik dari kedalaman tertentu. Sedangkan akibat tekanan yang tinggi pada sisi discharge akan memaksa fluida untuk naik sampai pada ketinggian yang diinginkan dan pada penggunaan pompa pada saat ini adalah pompa air aquarium yang digunakan untuk daerah indoor saja.

3.3.6 Perancangan Rangkaian Sensor pH

Sensor adalah komponen elektronika yang digunakan untuk merubah besaran fisik menjadi besaran listrik sehingga bisa di analisa dengan menggunakan rangkaian listrik. Pada prinsipnya sistem sensor pH (pH meter) terdiri dari elektroda pH yang digunakan untuk mendeteksi banyaknya ion H^+ dari suatu cairan. Pengukuran pH dilakukan dengan menggunakan elektroda potensiometrik. Elektroda ini memonitor perubahan voltase yang disebabkan oleh perubahan aktivitas ion hidrogen (H^+) dalam larutan. Elektroda pH yang paling modern terdiri dari kombinasi tunggal elektroda referensi (reference electrode) dan elektroda sensor (sensing electrode) yang lebih mudah dan lebih murah daripada elektroda terpisah. Elektroda kombinasi ini mempunyai fungsi yang sama dengan elektroda pasangan. Keluaran dari pH meter sudah dikalibrasi dalam mV dan kondisi ideal dari elektroda pH pada suhu $25^{\circ}C$. Dengan memonitor perubahan tegangan yang disebabkan oleh perubahan aktifitas ion hidrogen (H^+) dalam larutan maka pH

larutan dapat diketahui. Tegangan keluaran dari elektroda akan menunjukkan 0mV ketika dipakai untuk mengukur pH 7.00 (netral) (Rivai dkk., 2010)

Standar pH yang digunakan adalah 6,5 sampai 9, karena pada pH tersebut ikan dapat hidup (Rivai dkk., 2010). Ikan menjadi salah satu indikator air limbah yang layak dibuang ke laut tanpa menimbulkan efek pada lingkungan. Skema sensor pH ditampilkan pada gambar 3.7.

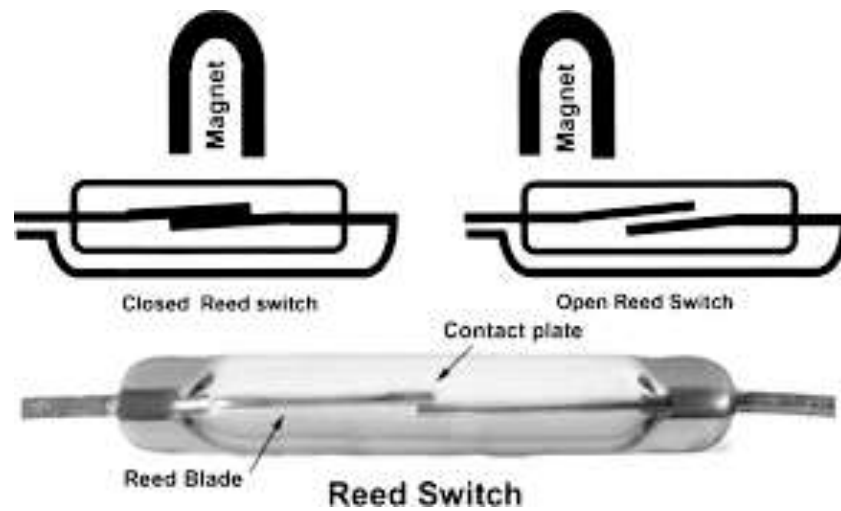


Gambar 3.7 Rangkaian Sensor pH

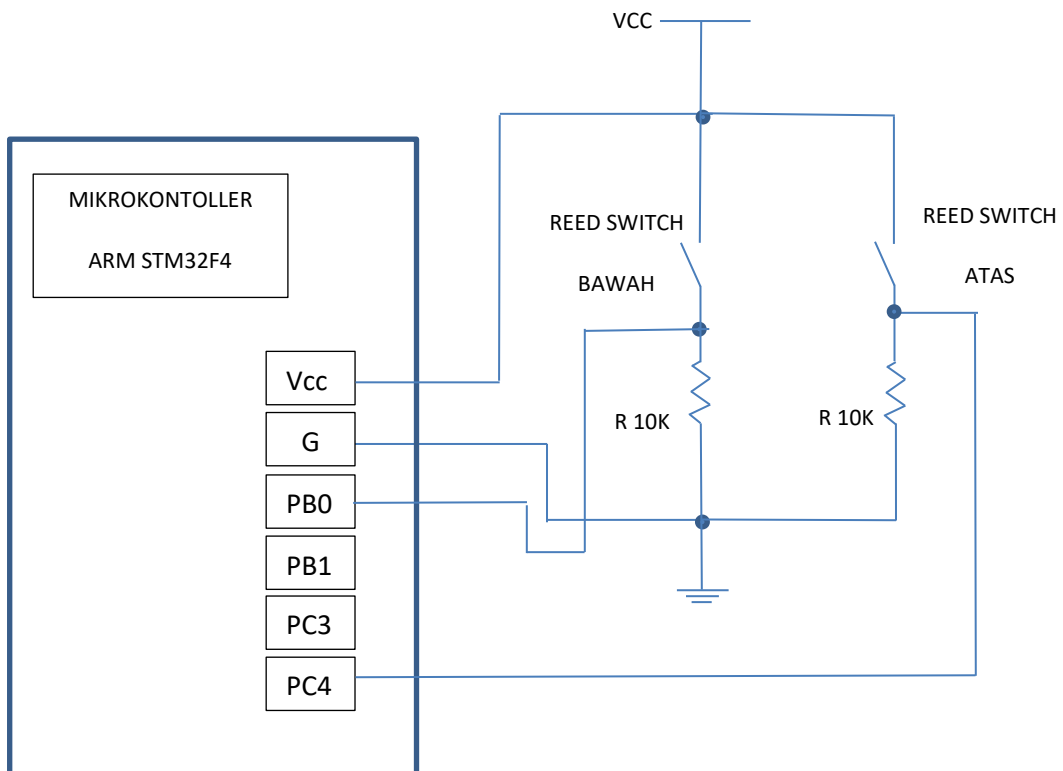
3.3.7 Perancangan Rangkaian Water Level Control (WLC)

Water Level Control (WLC) adalah alat yang berfungsi untuk mengontrol ketinggian level air, terdapat banyak komponen elektronik yang dapat digunakan untuk membuat WLC, dan salah satunya menggunakan reed swith dan magnet.

Ketika magnet yang digerakkan oleh ketinggian level air, menyentuh reedswitch maka reedswitch akan berlogika 1. Bentuk fisik Reed Switch ditampilkan pada gambar 3.9 dan rangkaian water level control disajikan pada gambar 3.10 (dunia elektronika.com).



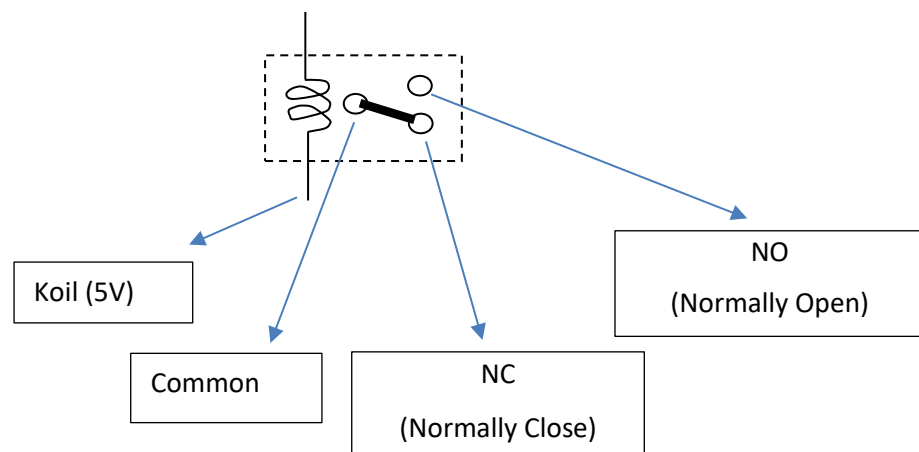
Gambar 3.8 Bentuk Fisik Reed Switch



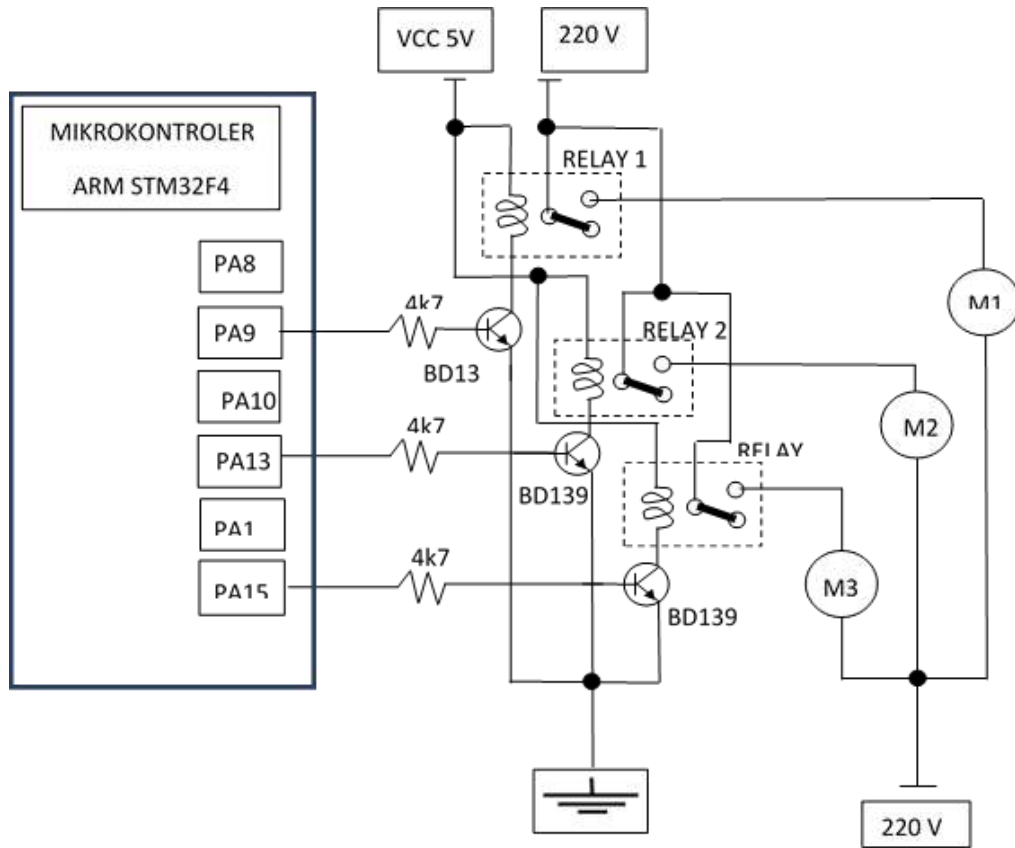
Gambar 3.9 Gambar Rangkaian Water Level Control

3.3.8 Perancangan Rangkaian Relay

Prinsip kerja relay ialah ketika koil relay dialiri tegangan 5V maka koil tersebut akan mengeluarkan medan magnet sehingga kontak-kontak dari relay tersebut akan bekerja sesuai dengan hukum Faraday. Terdapat 2 kontak yaitu kontak NO (Normally Open) dan NC (Normally Close) dimana ujung dari kontak tersebut sama yang disebut common. Dan kontak-kontak tersebut bebas untuk tegangan 5 – 220 V. Konstruksi relay dapat dilihat seperti gambar 3.10 dan rangkaian relay serta motor ditunjukkan pada gambar 3.11



Gambar 3.10 Konstruksi Relay

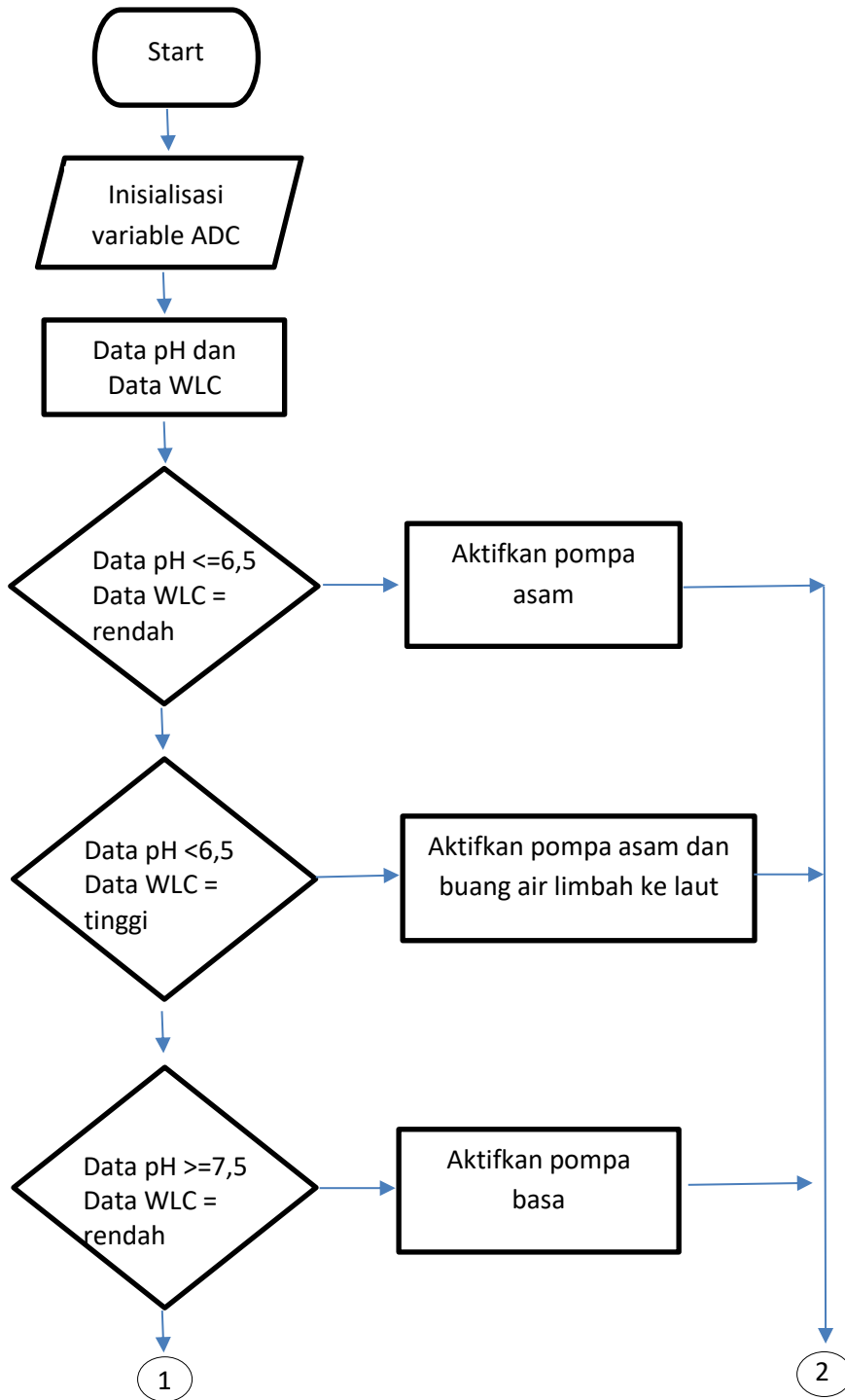


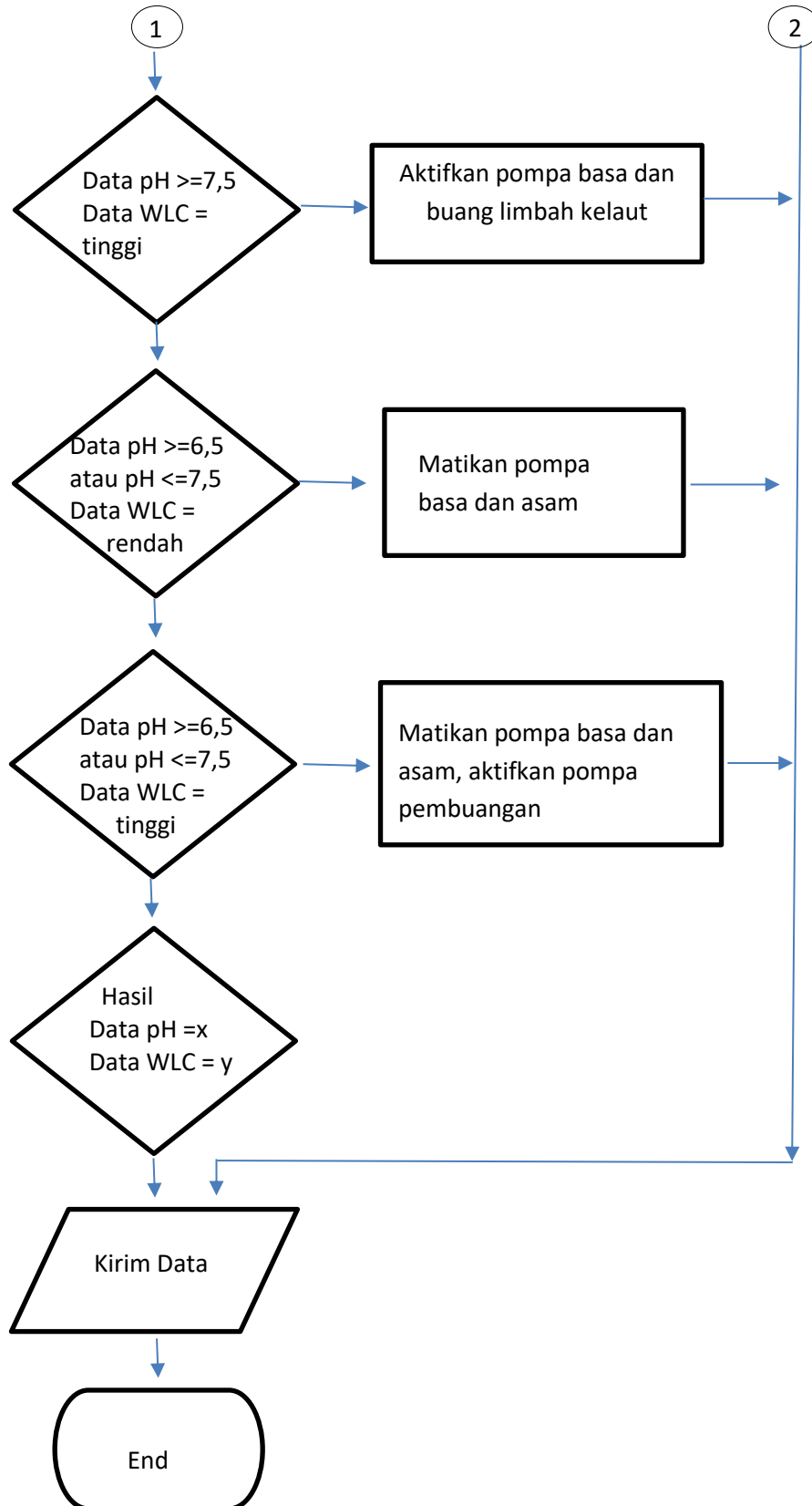
Gambar 3.11 Rangkaian Relay serta Motor

3.4. Perancangan *Software*

Pada pembuatan alat pembuangan air limbah pupuk ini, kode program yang telah ditulis akan didownload ke prosesor STM32F4 Discovery menggunakan program CoIDE. CoIDE adalah sebuah program yang telah dikembangkan secara umum untuk memberikan akses kemudahan dalam pemrograman prosesor ARM. Program dibuat dan disesuaikan sehingga sistem monitoring dan pengaturan kadar pH beserta pembuangan air limbah ke laut berbasis mikrokontroler ARM32F4 berjalan dengan baik.

Secara umum *flowchart* utama program menggunakan mikrokontroler ARM terlihat pada gambar 3.12





Gambar 3.12 *flowchart* Alur Kerja Sistem Monitoring dan Pengaturan Kadar pH beserta Pembuangan Air Limbah ke Laut Berbasis Mikrokontroler ARM32F4

Pada perancangan *software* sistem monitoring dan pengaturan kadar pH beserta pembuangan air limbah ke laut berbasis mikrokontroler ARM32F4 ini menggunakan bahasa C dan Coocox CoIDE sebagai *compiler* nya dapat dilihat pada gambar 3.13. Bahasa C digunakan untuk membuat program ini, di butuhkan untuk mengatur kinerja *hardware* sehingga dapat berjalan sesuai dengan yang di harapkan.



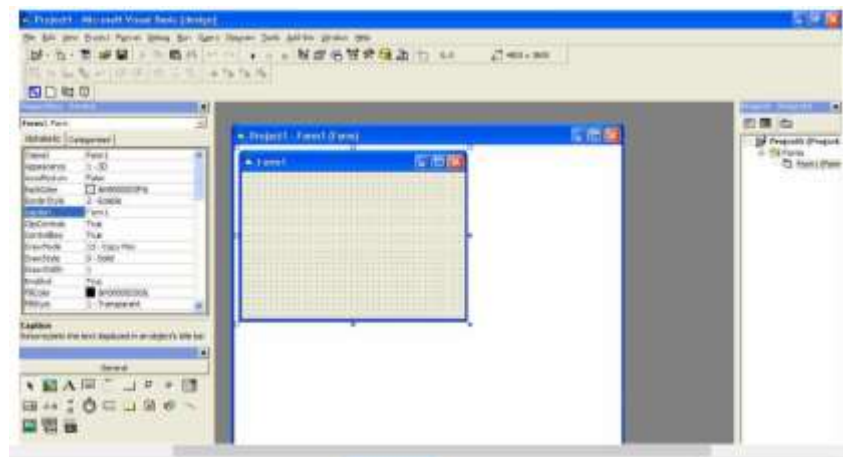
Gambar 3.13 *Compiler CoocoxCoIDE*

Proses awal pemrograman adalah inisialisasi mikrokontroler kemudian jika sensor pH membaca nilai besaran kadar pH di kolam lalu data di kirim oleh transmitter ke mikrokontroller. Sensor water level (reed switch) juga mengirimkan data level kolam ke mikrokontroller. Di mikrokontroller data diolah apabila kadar pH kurang dari 6,5 maka motor 1 akan bekerja dan menambah cairan basa sampai kadar pH normal yaitu 6,6-7,4. Apabila kadar pH lebih dari 7,5 maka motor 2 akan bekerja dan menambah cairan asam sampai kadar pH normal 6,6-7,4. Apabila sensor level kolam menunjukkan posisi air kolam tinggi maka motor 3 akan bekerja dan membuang air kolam ke laut sampai sensor level kolam menunjukkan posisi

rendah. Setiap data yang diolah mikrokontroller akan dikirim ke PC di controlroom melalui Bluetooth.

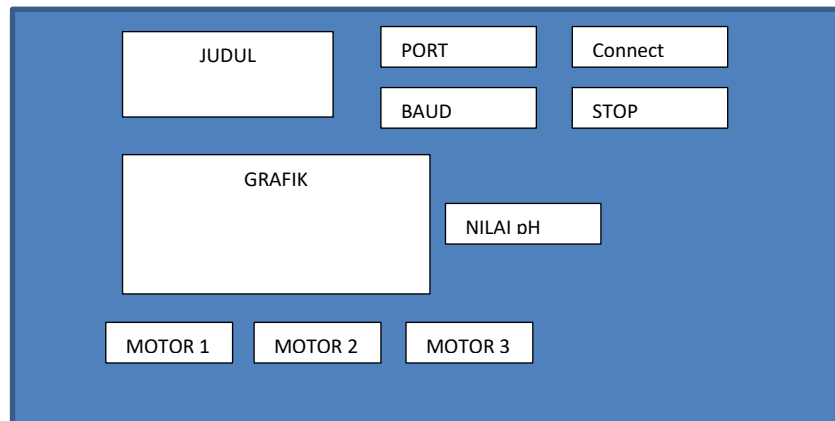
3.4.1. Pemrograman Menggunakan *Visual Basic*

Visual Basic adalah sebuah bahasa pemrograman komputer untuk menghasilkan program-program berbasis windows. Visual Basic juga merupakan bahasa pemrograman OOP, yaitu pemrograman yang berorientasi pada objek, di mana VB 6.0 menyediakan objek-objek yang berguna dan mudah dipakai.



Gambar 3.14 Tampilan Awal Visual Basic

Agar data bisa ditampilkan dilayar PC maka diperlukan program visual basic serta desain tampilan agar data bisa dimonitor dengan baik. Menu tampilan terdiri dari data nilai pH berupa grafik serta nilai real, kondisi motor pompa yang bekerja, serta tombol untuk mengconnect dan menstop tampilan data seperti pada gambar 3.15



Gambar 3.15 Desain Tampilan Visual Basic

Membuat Tombol Port

Tombol Port ini bertujuan sebagai penghubung antara PC dan Mikrokontroler, data yang telah dikirim mikrokontroler ARM STM32F4 melalui Bluetooth HC-05 dan diterima Bluetooth internal PC yang terletak di Port

4. Berikut program untuk tombol Port

```
Public Class Form1
    '-----
    Private data1 As Double
    Private datam As Integer

    Private iterasi As Integer

    Dim myPort As Array
    Delegate Sub SetTextCallback(ByVal [text] As String)
    'Added to prevent threading errors during receiveing of data
    '-----
    Private Sub Form1_Load(sender As System.Object, e As
    System.EventArgs) Handles MyBase.Load

        myPort = IO.Ports.SerialPort.GetPortNames()
        ComboBox1.Items.AddRange(myPort)

    End Sub
```

Membuat Tombol Connect

Tombol ini berfungsi untuk mengaktifkan data agar dapat terbaca di PC, berikut program untuk tombol connect

```
Private Sub Button1_Click(sender As System.Object, e As
    System.EventArgs) Handles Button1.Click

    SerialPort1.PortName = ComboBox1.Text
    SerialPort1.BaudRate = ComboBox2.Text
    SerialPort1.Open() 'perintah untuk membuka port serial
    Button1.Enabled = False
    Button4.Enabled = True

End Sub
```

Membuat Tombol Stop

Tombol stop ini berfungsi untuk menonaktifkan data agar tidak terhubung di PC, berikut program untuk tombol stop

```
Private Sub Button4_Click(sender As System.Object, e As
    System.EventArgs) Handles Button4.Click
    SerialPort1.Close() 'perintah untuk menutup port serial
    Button1.Enabled = True
    Button4.Enabled = False
End Sub
```

Membuat pH Value dan Kondisi Motor

pH Value dan kondisi motor merupakan tampilan nilai pH yang real dalam bentuk grafik serta menampilkan kondisi motor yang bekerja. Berikut tampilan program untuk pH value dan kondisi motor

```

Private Sub ReceivedText(ByVal [text] As String) 'input from
ReadExisting
    'If text.Contains("#") Then
    'text = text.Replace("#", "")
    If IsNumeric(text) Then
        datam = CDb1(text / 10000)
        data1 = CDb1((text Mod 10000) / 100)
        Label1.Invoke(New updatebox(AddressOf update_),
data1.ToString)
        Label4.Invoke(New updatebox2(AddressOf update_2),
datam.ToString)
        iterasi = iterasi + 1
        If (datam = 0) Then
            Button3.BackColor = Color.Red
            Button5.BackColor = Color.Red
            Button6.BackColor = Color.Red
        ElseIf (datam = 1) Then
            Button3.BackColor = Color.Red
            Button5.BackColor = Color.Red
            Button6.BackColor = Color.Green
        ElseIf (datam = 2) Then
            Button3.BackColor = Color.Red
            Button5.BackColor = Color.Green
            Button6.BackColor = Color.Red
        ElseIf (datam = 3) Then
            Button3.BackColor = Color.Red
            Button5.BackColor = Color.Green
            Button6.BackColor = Color.Green
        ElseIf (datam = 4) Then
            Button3.BackColor = Color.Green
            Button5.BackColor = Color.Red
            Button6.BackColor = Color.Red
        ElseIf (datam = 5) Then
            Button3.BackColor = Color.Green
            Button5.BackColor = Color.Red
            Button6.BackColor = Color.Green
        ElseIf (datam = 6) Then
            Button3.BackColor = Color.Green
            Button5.BackColor = Color.Green
            Button6.BackColor = Color.Red
        ElseIf (datam = 7) Then
            Button3.BackColor = Color.Green
            Button5.BackColor = Color.Green
            Button6.BackColor = Color.Green
        End If
    Delegate Sub updatebox(ByVal t As String)
        Public Sub update_(ByVal t As String)
            Label1.Text = t
            Me.Chart1.Series("Series1").Points.AddXY(iterasi,
CDb1(t))
        End Sub
    Delegate Sub updatebox2(ByVal t As String)
        Public Sub update_2(ByVal t As String)
            Label4.Text = t
        End Sub

```

BAB IV

PENGUJIAN DAN ANALISA ALAT

Untuk mengetahui bekerja atau tidaknya perangkat yang telah dibuat harus dilakukan suatu pengujian berupa kinerja maupun dengan pengukuran terhadap perangkat juga terhadap respon yang dihasilkan. Suatu sistem dinyatakan bekerja dengan baik bila sistem itu bekerja sesuai dengan tujuan awal dicapai data pertama kali dilakukan perancangan. Pengujian alat yang akan dilakukan, meliputi perangkat lunak (*software*) dan perangkat keras (*hardware*) apakah dapat bekerja sesuai dengan yang dikehendaki.

4.1. Pengujian Mikrokontroler, Bluetooth, Sensor pH, Sensor WLC, Relay dan Motor

1) Tujuan

- a. Untuk mengetahui apakah Mikrokontroler STM32F4 ini berfungsi atau tidak.
- b. Untuk mengetahui apakah Bluetooth HC-05 bekerja dengan baik atau tidak.
- c. Untuk mengetahui sensitifitas dan respon dari sensor pH.
- d. Untuk mengetahui apakah WLC, relay dan motor bekerja dengan baik atau tidak.

2) Alat-alat yang Diperlukan Untuk Pengujian

- a. Mikrokontroler STM32F4.
- b. Bluetooth HC-05.
- c. WLC, relay dan motor.
- d. Laptop/Personal Computer.

e. DC Power Supply 5V.

f. Software *CoocoxCoIDE*

3) Tahap-Tahap Pengujian yang Dilakukan

a. Pengujian Rangkaian Catu Daya

Pengujian rangkaian catu daya bertujuan untuk mengetahui kesesuaian tegangan keluaran dari catu daya. Dari hasil pengujian diperoleh nilai tegangan keluaran sebesar 5,3V. Nilai tersebut adalah nilai tegangan tanpa beban, yang berarti rangkaian catu daya 5V dapat menghasilkan tegangan maksimal (tanpa beban) sebesar 5,3 V. Dari hasil pengujian diperoleh nilai tegangan keluaran 5,1V. Nilai tersebut adalah nilai tegangan dengan beban mikrokontroller, yang berarti rangkaian catu daya 5V dapat menghasilkan tegangan maksimal (dengan beban) sebesar 5,1V.

Dari hasil pengujian catu daya tersebut, dapat disimpulkan bahwa catu daya yang digunakan dapat diaplikasikan ke mikrokontroller

b. Pengujian LCD

Pengujian pada LCD 2x16 ini dilakukan untuk mengetahui apakah LCD dapat berfungsi atau tidak. Pada pengujian ini, port E pada mikrokontroler STM32F4 digunakan sebagai *output* yang dihubungkan dengan pin LCD, Source code program LCD adalah sebagai berikut:

```
lcd_init(16);

lcd_gotoxy(0,0);lcd_putsf("TUGAS AKHIR");
motor_1_mati;
motor_2_mati;
motor_3_mati;
```

```

delay_ms(1000);

lcd_gotoxy(0,0);
sprintf(lcd,"%0.2f  ",nilaiph);
lcd_puts(lcd);

lcd_gotoxy(0,1);
sprintf(lcd,"%d%d ||
%d%d%d",reed_switch_atas,reed_switch_bawah,motor1,motor2,mot
or3);

lcd_puts(lcd);

```

Hasil pengujian LCD dengan menggunakan program diatas seperti pada gambar 4.1. berikut ini :



Gambar 4.1 Tampilan LCD 16x2 M1632

Pada gambar 4.1. dapat dilihat LCD mampu menampilkan karakter yang ditulispada program. Hal ini menunjukkan bahwasannya LCD dalam keadaan baik.

c. Pengujian Sensor pH

Pengujian ini bertujuan untuk mengetahui besar penyimpangan yang terjadi pada sensor Ph, pengujian ini adalah membandingkan nilai Ph di larutan Ph buffer, dan sensor Ph.

Tabel 4.1 Hasil nilai pH buffer dan sensor pH

Ph BUFFER	NILAI Ph	NILAI TEGANGAN (V)	ERROR (%)
Ph 4	4,3	3,8	7,5%
Ph 7	6,7	3,35	4,28%
Ph 9	8.8	2,9	2.,2,8%
RATA-RATA KESALAHAN			4,66%

Dari hasil kalibrasi berdasar tabel 4.1 diketahui bahwa rata-rata kesalahan pengukuran Ph sebesar 4,66%. Error ini disebabkan oleh kurang presisinya kertas lakmus dan sensitivitas sensor Ph kurang. Berikut source code sensor pH:

```

while(1)
{
    nilaiph = (-0.02 * datane)+18.8;

    //logika
    if (reed_switch_atas == 1)
    {
        s_case_kondisi = 1;
    }

    switch (s_case_kondisi)
    {
    case 0:
        motor_3_mati;
        motor3 =0;
        if(nilaiph <= 6)
        {
            motor_1_aktif;
            motor1 = 1;
        }
        else if (nilaiph >= 6.5)
        {
            motor_1_mati;
            motor1 =0;
        }

        if(nilaiph >= 8)
        {
            motor_2_aktif;

```



```

        motor2= 1;
    }
    else if(nilaiph <= 7.5)
    {
        motor_2_mati;
        motor2=0;
    }
    break;
    case 1:
        motor_1_mati;
        motor1=0;
        motor_2_mati;
        motor2=0;
        motor_3_aktif;
        motor3 = 1;
        if(reed_switch_bawah == 1)
        {
            motor_3_mati;
            motor3 =0;
        }
        s_case_kondisi = 0;
        break;
    default:break;
}

```

d. Pengujian Bluetooth HC-05

Bluetooth merupakan komponen yang digunakan untuk mengirim data dari mikrokontroller ke PC. Tujuan pengujian bluetooth adalah untuk mengetahui tegangan yang masuk ke bluetooth dan mengetahui koneksi Bluetooth. Pengujian awal dilakukan dengan menguji tegangan bluetooth dan yang kedua dengan menguji konektivitas bluetooth.

Tabel 4.2 Hasil Pengujian Bluetooth

Point yang diuji	Tegangan (V)	Indikator	Terkoneksi/tidak	Status
Tegangan	2,7 V	Led berkedip	Tidak	OK
Koneksi	2,7 V	Lampu led menyala tidak berkedip	Terkoneksi	OK

Tegangan yang dibutuhkan oleh bluetooth adalah 3.6-6 volt, pada pengujian terukur tegangan yang masuk ke bluetooth adalah 4.6 volt sehingga hal ini telah memenuhi syarat untuk menyalakan bluetooth. Kemudian dilihat dari lampu indicator diketahui bahwa bluetooth ketika belum dikoneksikan led indicator akan berkedip, kemudian setelah terkoneksi akan menyala tetapi tidak berkedip.

Tabel 4.3 Pengujian Jarak Bluetooth HC-05

Percobaan	Jarak	Data Ter kirim	Keterangan
1	1 m	Ter kirim	Data diterima PC
2	2 m	Ter kirim	Data diterima PC
3	3 m	Ter kirim	Data diterima PC
4	4 m	Ter kirim	Data diterima PC
5	5 m	Ter kirim	Data diterima PC
6	6 m	Tidak	Data tidak diterima PC

Dari table di atas dapat diketahui bahwa data akan yang dikirim oleh Bluetooth hanya sampai jarak 5 meter, selebihnya data tidak dapat terkirim dengan baik. Berikut source code Bluetooth:

```

kirim[0]='#';
                                USART_SendData(USART6,kirim[0]); //header
                                while(USART_GetFlagStatus(USART6,USART_FLAG_TXE)
== 0);

                                //pengiriman status motor
                                if (motor1 == 0 && motor2 == 0 && motor3 ==
0) kirim[0]='0';

```

```

1) kirim[0]='1';
0) kirim[0]='2';
1) kirim[0]='3';
0) kirim[0]='4';
1) kirim[0]='5';
0) kirim[0]='6';
1) kirim[0]='7';

else if (motor1 == 0 && motor2 == 0 && motor3 ==
else if (motor1 == 0 && motor2 == 1 && motor3 ==
else if (motor1 == 0 && motor2 == 1 && motor3 ==
else if (motor1 == 1 && motor2 == 0 && motor3 ==
else if (motor1 == 1 && motor2 == 0 && motor3 ==
else if (motor1 == 1 && motor2 == 1 && motor3 ==
else if (motor1 == 1 && motor2 == 1 && motor3 ==

USART_SendData(USART6, kirim[0]); //header
while(USART_GetFlagStatus(USART6, USART_FLAG_TXE)
== 0);

nilai = nilaiph*100;

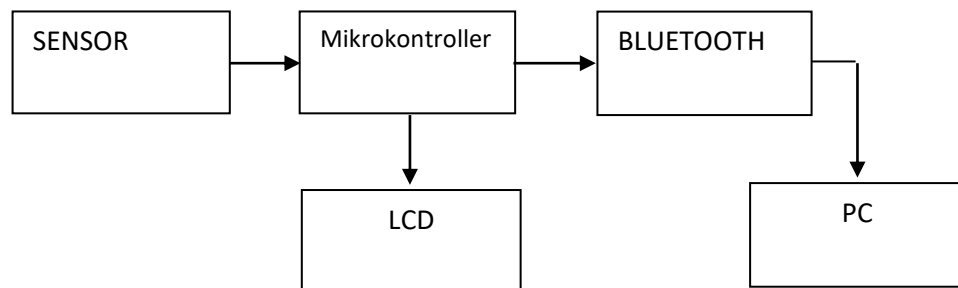
switch (nilai/1000)
{
case 0:kirim[0]='0';break;
case 1:kirim[0]='1';break;
case 2:kirim[0]='2';break;
case 3:kirim[0]='3';break;
case 4:kirim[0]='4';break;
case 5:kirim[0]='5';break;
case 6:kirim[0]='6';break;
case 7:kirim[0]='7';break;
case 8:kirim[0]='8';break;
case 9:kirim[0]='9';break;
default:kirim[0]='0';break;
}
USART_SendData(USART6, kirim[0]); //header
while(USART_GetFlagStatus(USART6, USART_FLAG_TXE)
== 0);

//satuan
switch ((nilai%1000)/100)
{
case 0:kirim[0]='0';break;
case 1:kirim[0]='1';break;
case 2:kirim[0]='2';break;
case 3:kirim[0]='3';break;
case 4:kirim[0]='4';break;
case 5:kirim[0]='5';break;
case 6:kirim[0]='6';break;
case 7:kirim[0]='7';break;
case 8:kirim[0]='8';break;
case 9:kirim[0]='9';break;
default:kirim[0]='0';break;
}

```


e. Pengujian Mikrokontroler STM32F04

Pengujian mikrokontroler ini bertujuan untuk mengetahui apakah mikrokontroler bekerja dengan baik dalam memproses data dari sensor serta pengiriman data, sampai tampilan data di LCD dan PC. Prosedur pengujian dilakukan dengan menghubungkan mikrokontroler dengan semua sensor. LCD dan PC digunakan sebagai media untuk menampilkan hasil dari data sensor.



Gambar 4.2 Blok Diagram Mikrokontroler

Dari pengujian mikrokontroler yang dilakukan dapat disimpulkan bahwa mikrokontroler STM32F4 dapat membaca sensor secara realtime dan bekerja sesuai dengan perancangan yang dilakukan

f. Pengujian WLC, Relay, dan Pompa Pembuangan

Perubahan kondisi relay on/off adalah karena perubahan nilai pH dan perubahan level kolam. Pada kondisi nilai pH kurang dari 6,5 maka relay 1 akan bekerja, bila kondisi nilai pH lebih dari 7,5 maka relay 2 akan bekerja, bila level kolam tinggi maka relay 3 akan bekerja.

Berikut table percobaan sensor WLC, Relay3 dan Pompa Pembuangan

Tabel 4.4 Hasil Tegangan pada Output

Level Kolam	Tegangan (V)		Tegangan (V)	
	Port C.4	Port B.0	Relay 3	Pompa Pembuangan
Tinggi	0	2,9	4,6	213 Vac
Rendah	2,9	0	0	0

Berikut source code relay dan pompa

```

while(1)
{
    nilaiph = (-0.02 * datane)+18.8;

    //logika
    if (reed_switch_atas == 1)
    {
        s_case_kondisi = 1;
    }

    switch (s_case_kondisi)
    {
    case 0:
        if(nilaiph <= 6)
        {
            motor_1_aktif;
            motor1 = 1;
        }
        else if (nilaiph >= 6.5)
        {
            motor_1_mati;
            motor1 =0;
        }

        if(nilaiph >= 8)
        {
            motor_2_aktif;
            motor2= 1;
        }
        else if(nilaiph <= 7.5)
        {
            motor_2_mati;
            motor2=0;
        }
    }
    break;
    case 1:
        motor_1_mati;
    }
}

```

```

motor1=0;
motor_2_mati;
motor2=0;
motor_3_aktif;
motor3 = 1;
if(reed_switch_bawah == 1)
{
    s_case_kondisi = 0;
    motor_3_mati;
    motor3 =0;
}
break;
default:break;
}

```

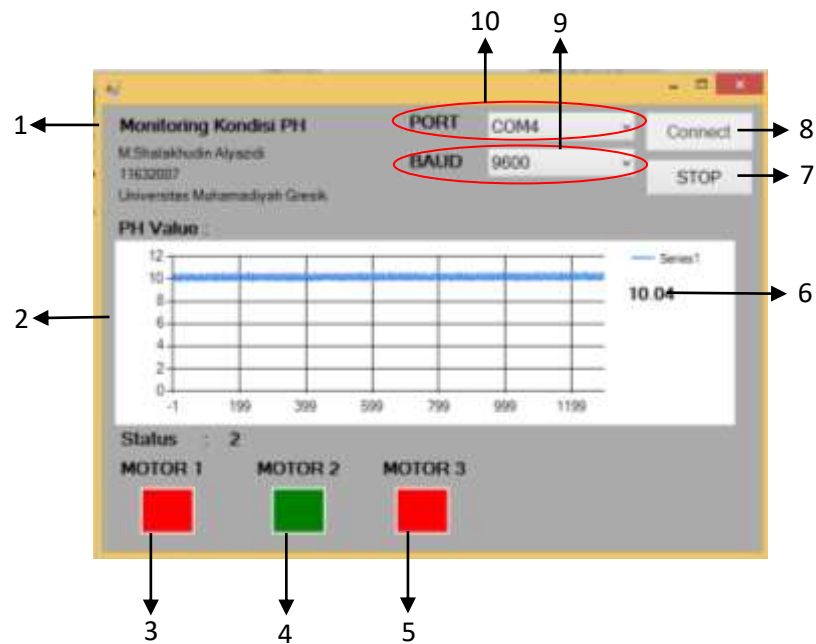
Tabel 4.5 Hasil Percobaan Relay1 dan 2 serta Pompa Asam dan Basa

Sensor pH		Tegangan (V)		Tegangan (V)		Tegangan (Vac)	
Nilai	(V)	Port A.9	Port A.13	Relay 1	Relay 2	Pompa Asam	Pompa Basa
11	1,8	0	2,9	0	4,6	211	0
10,3	1,9	0	2,9	0	4,6	213	0
9,3	2,2	0	2,9	0	4,6	213	0
7,3	2,8	0	0	0	0	0	0
5,6	3,2	2,9	0	4,6	0	0	215
3,2	3,8	2,9	0	4,6	0	0	213

g. Pengujian Sistem Keseluruhan

Pengujian sistem secara keseluruhan bertujuan untuk mengetahui error yang terjadi ketika semua sensor memberikan inputan kepada mikrokontroller dalam mengkondisikan data yang sesuai dengan setpoint dari system yang telah dibuat. Pengujian sistem secara keseluruhan dilakukan pada semua sensor dengan membandingkan nilai yang terbaca oleh sensor dengan setpoint yang ditentukan kemudian mengirimkan data

untuk dimonitoring melalui LCD atau PC untuk mengetahui apakah perintah yang dikirim sudah benar.



Gambar 4.3 Tampilan monitor kadar pH pada PC

Keterangan gambar:

1. Judul layar tampilan
2. Tampilan chart
3. Status motor 1 (pompa basa), warna hijau apabila motor bekerja dan berwarna merah apabila motor mati
4. Status motor 2 (pompa asam), warna hijau apabila motor bekerja dan berwarna merah apabila motor mati
5. Status motor 3 (pompa pembuangan), warna hijau apabila motor bekerja dan berwarna merah apabila motor mati
6. Hasil nilai kadar pH
7. Untuk menghentikan proses pemantauan kadar pH

8. Untuk menjalankan proses pemantauan kadar pH
9. Kecepatan pengiriman data tiap detik
10. Untuk media penghubung antara PC dan bluetooth

Berdasar data hasil pengujian, dapat diketahui bahwa data yang dikirim oleh sensor pH dan WLC dapat diterima dengan baik oleh mikrokontroler serta data dapat terkirim dengan baik ke PC maupun LCD. Tampilan hardware keseluruhan dapat dilihat di lampiran

h. Pengujian Visual Basic

Berikut source code visual basic:

```
Public Class Form1
    '-----
    Private data1 As Double
    Private datam As Integer

    Private iterasi As Integer

    Dim myPort As Array
    Delegate Sub SetTextCallback(ByVal [text] As String) 'Added to prevent
    threading errors during receiveing of data
    '-----
    Private Sub Form1_Load(sender As System.Object, e As System.EventArgs)
    Handles MyBase.Load

        myPort = IO.Ports.SerialPort.GetPortNames()
        ComboBox1.Items.AddRange(myPort)

    End Sub
    '-----

    Private Sub Button1_Click(sender As System.Object, e As
    System.EventArgs) Handles Button1.Click

        SerialPort1.PortName = ComboBox1.Text
        SerialPort1.BaudRate = ComboBox2.Text
        SerialPort1.Open() 'perintah untuk membuka port serial
        Button1.Enabled = False
    End Sub
End Class
```

```

        Button4.Enabled = True

    End Sub
'-----

    Private Sub Button4_Click(sender As System.Object, e As
System.EventArgs) Handles Button4.Click
        SerialPort1.Close() 'perintah untuk menutup port serial
        Button1.Enabled = True
        Button4.Enabled = False
    End Sub

    Private Sub SerialPort1_DataReceived(sender As System.Object, e As
System.IO.Ports.SerialDataReceivedEventArgs) Handles
SerialPort1.DataReceived
        ReceivedText(SerialPort1.ReadExisting())
    End Sub

    Private Sub ReceivedText(ByVal [text] As String) 'input from
ReadExisting
        'If text.Contains("#") Then
        'text = text.Replace("#", "")
        If IsNumeric(text) Then
            datam = CDb1(text / 10000)
            data1 = CDb1((text Mod 10000) / 100)
            Label1.Invoke(New updatebox(AddressOf update_), data1.ToString)
            Label4.Invoke(New updatebox2(AddressOf update_2),
datam.ToString)
            iterasi = iterasi + 1

            If (datam = 0) Then
                Button3.BackColor = Color.Red
                Button5.BackColor = Color.Red
                Button6.BackColor = Color.Red
            ElseIf (datam = 1) Then
                Button3.BackColor = Color.Red
                Button5.BackColor = Color.Red
                Button6.BackColor = Color.Green
            ElseIf (datam = 2) Then
                Button3.BackColor = Color.Red
                Button5.BackColor = Color.Green
                Button6.BackColor = Color.Red
            ElseIf (datam = 3) Then
                Button3.BackColor = Color.Red
                Button5.BackColor = Color.Green
                Button6.BackColor = Color.Green
            ElseIf (datam = 4) Then
                Button3.BackColor = Color.Green
                Button5.BackColor = Color.Red
                Button6.BackColor = Color.Red
            ElseIf (datam = 5) Then
                Button3.BackColor = Color.Green
                Button5.BackColor = Color.Red
                Button6.BackColor = Color.Green
            ElseIf (datam = 6) Then
                Button3.BackColor = Color.Green

```

```
        Button5.BackColor = Color.Green
        Button6.BackColor = Color.Red
    ElseIf (datam = 7) Then
        Button3.BackColor = Color.Green
        Button5.BackColor = Color.Green
        Button6.BackColor = Color.Green

    End If

    End If

End Sub

Delegate Sub updatebox(ByVal t As String)
Public Sub update_(ByVal t As String)
    Label1.Text = t
    Me.Chart1.Series("Series1").Points.AddXY(iterasi, Cdbl(t))

End Sub

Delegate Sub updatebox2(ByVal t As String)
Public Sub update_2(ByVal t As String)
    Label14.Text = t
End Sub

Private Sub Button2_Click(sender As Object, e As EventArgs)

    End Sub
End Class
```

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil data yang diperoleh melalui pengujian dapat disimpulkan sebagai berikut:

1. Sistem monitoring dan pengontrolan kolam limbah secara otomatis dapat menggunakan Mikrokontroler STM32F4 dan sensor pH meter. Perangkat ini disetting pada range pH 6,6 - 7,4 sehingga pH air limbah dapat terkontrol sebelum dibuang ke laut.
2. Perangkat secara otomatis akan mengaktifkan pompa asam apabila pH air limbah diatas 7,5 dan akan mengaktifkan pompa basa apabila air limbah dibawah 6,5
3. Terdapat nilai rata-ata penyimpangan pembacaan nilai pH pada perangkat ini sebesar 4,66%. Nilai penyimpangan ini diperoleh dari tidak stabilnya pembacaan pH pada sensor tersebut.

B. Saran

1. Pada penelitian selanjutnya menggunakan elektroda pH yang sensitive sehingga data pengukuran yang didapat mendekati nilai yang sebenarnya

DAFTAR PUSTAKA

1. Heryanto, Ari. M dan Wisnu Adi P. 2008." *Pemrograman Bahasa C Untuk Mikrokontroler ATMEGA8535*. Yogyakarta: Andi.Putra Arianto
Eko,2010,*Mikrokontroller AT89 dan AVR*, Yogyakarta: Gava Media.
2. <http://atmel.com>
3. <http://cdn2.bigcommerce.com>
4. <http://dunia elektronik.com>
5. Liyan, Derry Deksa. *PEMANFAATAN IC ATMEGA32 SEBAGAI PENGONTROL ALAT ELEKTRONIK MENGGUNAKAN SMARTPHONE ANDROID*. Diss. Politeknik Negeri Sriwijaya, 2014.
6. Putri, Mareta Tri. *PENGENDALI LAMPU RUMAH OTOMATIS DENGAN SMS BERBASIS MIKROKONTROLLER ATMEGA8535*. Diss. Politeknik Negeri Sriwijaya, 2014.
7. Rivai, M., Dikairono, R., & Tomi, A. (2010). Sistem Monitoring PH dan Suhu Air dengan Transmisi Data Nirkabel. *JAVA Journal of Electrical and Electronics Engineering*, 8(2).
8. Sholihul Hadi M,2013, *Mengenal Mikrokontroller AVR ATMega16*, Bandung:Ilmu Komputer

LAMPIRAN

Lampiran 1. Sorce Code Mikrokontroler ARM-STM32F4

```
#include "stm32f4xx_conf.h"
#include "delay.h"
#include "lcd_biasa.h"
#include "math.h"
#include "stdio.h"
#include "stdlib.h"
#include "stm32f4xx_adc.h"
#include "stm32f4xx_i2c.h"
#include "stm32f4xx_usart.h"

#define motor_1_aktif          GPIO_WriteBit(GPIOA, GPIO_Pin_8, 1) //
#define motor_1_mati          GPIO_WriteBit(GPIOA, GPIO_Pin_8, 0) //

#define motor_2_aktif          GPIO_WriteBit(GPIOA, GPIO_Pin_10, 1) //
#define motor_2_mati          GPIO_WriteBit(GPIOA, GPIO_Pin_10, 0) //

#define motor_3_aktif          GPIO_WriteBit(GPIOA, GPIO_Pin_14, 1) //
#define motor_3_mati          GPIO_WriteBit(GPIOA, GPIO_Pin_14, 0) //

#define reed_switch_atas      GPIO_ReadInputDataBit (GPIOC,
GPIO_Pin_4)==0
#define reed_switch_bawah      GPIO_ReadInputDataBit (GPIOB,
GPIO_Pin_0)==0

//-----Variabel gyro(KOMPAS) -----//
char data_usart2=0, penanda_usart2=0, penanda_usart_data2=0,
data_gyro[6]={10,10,10,10,10,10};
char data_gyro[6];
char *join_gyro;
double derajat =0,derajat_2,gyro=0,dt_gyro=0,offset_gyro =
0,dt_gyro_2=0,dt_pitch=0,offset_gyro_2 = 0 ;
int datane;

char lcd[16];
char kirim[100];
int NILAI_SENSOR=1;
int nilai;
float nilaiph;
char status_motor,motor1,motor2,motor3;
//////////
unsigned long int avgValue; //Store the average value of the sensor
feedback
float b;
int i,j;
int buf[10],temp;
//////////

void inisialisasi_ADC(void);
uint16_t baca_ADC(uint8_t pin_ADC);
```

```

void inisialisasi_USART6();

void inisialisasi_ADC(void)
{
    ADC_InitTypeDef          ADC_InitStructure;
    ADC_CommonInitTypeDef    ADC_CommonInitStructure;
    GPIO_InitTypeDef         GPIO_InitStructure;

    RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOA, ENABLE);
    RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOC, ENABLE);
    RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOB, ENABLE);
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_ADC1, ENABLE);

    /* Configure ADC1 Channell1 pin as analog input
    *****/

    /******Konfigurasi ADC1*****/
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_0|GPIO_Pin_1;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AN;
    GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_NOPULL;
    GPIO_Init(GPIOB, &GPIO_InitStructure);

    /******inisialisasi ADC common*****/
    ADC_CommonInitStructure.ADC_Mode = ADC_Mode_Independent;
    ADC_CommonInitStructure.ADC_Prescaler = ADC_Prescaler_Div2;
    ADC_CommonInitStructure.ADC_DMAAccessMode =
    ADC_DMAAccessMode_Disabled;
    ADC_CommonInitStructure.ADC_TwoSamplingDelay =
    ADC_TwoSamplingDelay_5Cycles;
    ADC_CommonInit(&ADC_CommonInitStructure);

    /******inisialisasi ADC1 *****/
    ADC_InitStructure.ADC_Resolution = ADC_Resolution_8b;
    ADC_InitStructure.ADC_ScanConvMode = ENABLE;
    ADC_InitStructure.ADC_ContinuousConvMode = DISABLE;
    ADC_InitStructure.ADC_ExternalTrigConvEdge =
    ADC_ExternalTrigConvEdge_None;
    ADC_InitStructure.ADC_ExternalTrigConv =
    ADC_ExternalTrigConv_T1_CC1;
    ADC_InitStructure.ADC_DataAlign = ADC_DataAlign_Right;
    ADC_InitStructure.ADC_NbrOfConversion = 1;
    ADC_Init(ADC1, &ADC_InitStructure);

    ADC_Cmd(ADC1, ENABLE);

}
#define pin_ADC_B0 8

int ADC_Val;
uint16_t baca_ADC(uint8_t pin_ADC)
{

```

```

switch(pin_ADC)
{
    case pin_ADC_B0:
        ADC_RegularChannelConfig(ADC1, ADC_Channel_8,1,
ADC_SampleTime_3Cycles);

        ADC_SoftwareStartConv(ADC1);
        while(ADC_GetSoftwareStartConvStatus(ADC1) !=
RESET){ADC_Val = 0;}
        ADC_Val = ADC_GetConversionValue(ADC1);
        break;

    default:
        ADC_Val = 0;
        break;
}
return ADC_Val;
}

char nilai_adc[16];

void inisialisasi_USART2() //gyro
{
    RCC_APB1PeriphClockCmd(RCC_APB1Periph_USART2, ENABLE);

    GPIO_InitTypeDef definisi_GPIO;
    USART_InitTypeDef definisi_USART;
    NVIC_InitTypeDef definisi_NVIC;

    definisi_GPIO.GPIO_Pin = GPIO_Pin_3;
    definisi_GPIO.GPIO_Mode = GPIO_Mode_AF;
    definisi_GPIO.GPIO_PuPd = GPIO_PuPd_UP;
    //definisi_GPIO.GPIO_OType = GPIO_OType_PP;
    definisi_GPIO.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_Init(GPIOA, &definisi_GPIO);

    GPIO_PinAFConfig(GPIOA, GPIO_PinSource3, GPIO_AF_USART2);

    definisi_USART.USART_BaudRate = 9600;
    definisi_USART.USART_Mode = USART_Mode_Rx;
    definisi_USART.USART_WordLength = USART_WordLength_8b;
    definisi_USART.USART_StopBits = USART_StopBits_1;
    definisi_USART.USART_Parity = USART_Parity_No;
    definisi_USART.USART_HardwareFlowControl =
USART_HardwareFlowControl_None;
    USART_Init(USART2, &definisi_USART);

    definisi_NVIC.NVIC_IRQChannel = USART2_IRQn;
    definisi_NVIC.NVIC_IRQChannelCmd = ENABLE;
    NVIC_Init(&definisi_NVIC);

    USART_ITConfig(USART2, USART_IT_RXNE, ENABLE);

    USART_Cmd(USART2, ENABLE);
}

```



```

void USART2_IRQHandler(void) //fungsi gyro
{
    if( USART_GetITStatus(USART2, USART_IT_RXNE))
    {
        data_usart2 = USART_ReceiveData(USART2);
        //while(USART_GetFlagStatus(USART2, USART_FLAG_RXNE) ==
RESET);
        // header untuk mengeluarkan data dari gyro
        if(penanda_usart2==0)
        {
            if(data_usart2=='a')
                penanda_usart2=1;
            else
                penanda_usart2=0;
        }
        else if(penanda_usart2==1)
        {
            if(data_usart2=='i')
                penanda_usart2=2;
            else
                penanda_usart2=0;
        }
        else if(penanda_usart2==2)
        {
            if(data_usart2=='u')
                penanda_usart2=3;
            else
                penanda_usart2=0;
        }
        else if(penanda_usart2==3)
        {
            if(data_usart2=='e')
                penanda_usart2=4;
            else
                penanda_usart2=0;
        }
        else if(penanda_usart2==4)
        {
            switch(penanda_usart_data2)
            {
                case 0      : data_gyro[0]      =
data_usart2; penanda_usart_data2 = 1; break;
                case 1      : data_gyro[1]      =
data_usart2; penanda_usart_data2 = 2; break;
                case 2      : data_gyro[2]      = data_usart2;
penanda_usart_data2 = 3; break;
                case 3      : data_gyro[3]      = data_usart2;
penanda_usart_data2 = 4; break;
                case 4      : data_gyro[4]      = data_usart2;
penanda_usart_data2 = 0;penanda_usart2=0;break;
            }
        }

        datane = strtold(data_gyro,&join_gyro);
        //gyro = -1*(gyro/100);
    }
}

```

```

        //pitch = strtold(data_p,&join_p);
        //pitch = -1*(pitch/10);

        USART_ClearITPendingBit(USART2,
USART_IT_RXNE);
    }
}

void inisialisasi_GPIO()
{
    GPIO_InitTypeDef definisi_GPIO;

    definisi_GPIO.GPIO_Pin = GPIO_Pin_4;
    definisi_GPIO.GPIO_Mode = GPIO_Mode_IN;
    definisi_GPIO.GPIO_PuPd = GPIO_PuPd_UP;
    definisi_GPIO.GPIO_OType = GPIO_OType_PP;
    definisi_GPIO.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_Init(GPIOC, &definisi_GPIO);

    definisi_GPIO.GPIO_Pin = GPIO_Pin_0;
    definisi_GPIO.GPIO_Mode = GPIO_Mode_IN;
    definisi_GPIO.GPIO_PuPd = GPIO_PuPd_UP;
    definisi_GPIO.GPIO_OType = GPIO_OType_PP;
    definisi_GPIO.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_Init(GPIOB, &definisi_GPIO);

    //////////////////////////////////////

    definisi_GPIO.GPIO_Pin = GPIO_Pin_8|GPIO_Pin_10 |GPIO_Pin_14 ;
    definisi_GPIO.GPIO_Mode = GPIO_Mode_OUT;
    definisi_GPIO.GPIO_PuPd = GPIO_PuPd_UP;
    definisi_GPIO.GPIO_OType = GPIO_OType_PP;
    definisi_GPIO.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_Init(GPIOA, &definisi_GPIO);
    //////////////////////////////////////
}

void inisialisasi_USART6()
{
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_USART6, ENABLE);

    GPIO_InitTypeDef definisi_GPIO;
    USART_InitTypeDef definisi_USART;

    definisi_GPIO.GPIO_Pin = GPIO_Pin_6;
    definisi_GPIO.GPIO_Mode = GPIO_Mode_AF;
    definisi_GPIO.GPIO_PuPd = GPIO_PuPd_UP;
    definisi_GPIO.GPIO_OType = GPIO_OType_PP;
    definisi_GPIO.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_Init(GPIOC, &definisi_GPIO);

    GPIO_PinAFConfig(GPIOC, GPIO_PinSource6, GPIO_AF_USART6);

    definisi_USART.USART_BaudRate = 9600;

```

```

        definisi_USART.USART_Mode = USART_Mode_Tx;
        definisi_USART.USART_WordLength = USART_WordLength_8b;//
USART_WordLength_8b;
        definisi_USART.USART_StopBits = USART_StopBits_1;
        definisi_USART.USART_Parity = USART_Parity_No;
        definisi_USART.USART_HardwareFlowControl =
USART_HardwareFlowControl_None;

        USART_Init(USART6, &definisi_USART);
        USART_Cmd(USART6, ENABLE);
    }

void inisialisasi_timer6() // timer interrupt eksekusi tiap 1ms
{
    RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM6, ENABLE);

    TIM_TimeBaseInitTypeDef definisi_timebase;

    definisi_timebase.TIM_Prescaler = 1000;
    definisi_timebase.TIM_Period =84; //periode-1;
    definisi_timebase.TIM_ClockDivision = TIM_CKD_DIV1; // lihat
reference manual hal.367
    definisi_timebase.TIM_CounterMode = TIM_CounterMode_Up;
    TIM_TimeBaseInit(TIM6, &definisi_timebase);

    /* Enable TIM3 Preload register on ARR */
    TIM_ARRPreloadConfig(TIM6, ENABLE);

    TIM_Cmd(TIM6, ENABLE);

    NVIC_InitTypeDef definisi_NVIC;

    definisi_NVIC.NVIC_IRQChannel = TIM6_DAC_IRQn;
    definisi_NVIC.NVIC_IRQChannelCmd = ENABLE;
    NVIC_Init(&definisi_NVIC);

    TIM_ITConfig(TIM6, TIM_IT_Update, ENABLE);
}
void TIM6_DAC_IRQHandler()
{
    ///////
    TIM_ClearITPendingBit(TIM6, TIM_IT_Update);
}

//masukkan

//
char s_case_kondisi;
int main(void)
{

    SystemInit();

    RCC_APB2PeriphClockCmd(RCC_APB2Periph_SYSCFG, ENABLE);
//enable interrupt

```

```

RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOA, ENABLE);
RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOB, ENABLE);
RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOC, ENABLE);
RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOD, ENABLE);
RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOE, ENABLE);
RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOH, ENABLE);

inisialisasi_GPIO();
inisialisasi_USART6();
inisialisasi_timer6();
inisialisasi_USART2();
// inisialisasi_ADC();

lcd_init(16);

lcd_gotoxy(0,0);lcd_putsf("TUGAS AKHIR");
motor_1_mati;
motor_2_mati;
motor_3_mati;

delay_ms(1000);

while(1)
{
    nilaiph = (-0.02 * datane)+18.8;

    //logika
    if (reed_switch_atas == 1)
    {
        s_case_kondisi = 1;
    }

    switch (s_case_kondisi)
    {
    case 0:
        motor_3_mati;
        motor3 =0;
        if(nilaiph <= 6)
        {
            motor_1_aktif;
            motor1 = 1;
        }
        else if (nilaiph >= 6.5)
        {
            motor_1_mati;
            motor1 =0;
        }

        if(nilaiph >= 8)
        {
            motor_2_aktif;
            motor2= 1;
        }
        else if(nilaiph <= 7.5)
        {

```

```

        motor_2_mati;
        motor2=0;
    }
    break;
    case 1:
        motor_1_mati;
        motor1=0;
        motor_2_mati;
        motor2=0;
        motor_3_aktif;
        motor3 = 1;
        if(reed_switch_bawah == 1)
        {
            motor_3_mati;
            motor3 =0;
        }
    break;
    s_case_kondisi = 0;
    default:break;
}

//
lcd_clear();

    lcd_gotoxy(0,0);
    sprintf(lcd,"%0.2f  ",nilaiph);
    lcd_puts(lcd);

    lcd_gotoxy(0,1);
    sprintf(lcd,"%d%d ||
%d%d%d",reed_switch_atas,reed_switch_bawah,motor1,motor2,motor3);
    lcd_puts(lcd);

    ///////////////////////////////////
    kirim[0]='#';
    USART_SendData(USART6,kirim[0]); //header
    while(USART_GetFlagStatus(USART6,USART_FLAG_TXE) ==
0);

    //pengiriman status motor
    if      (motor1 == 0 && motor2 == 0 && motor3 ==
0)kirim[0]='0';
    else if (motor1 == 0 && motor2 == 0 && motor3 ==
1)kirim[0]='1';
    else if (motor1 == 0 && motor2 == 1 && motor3 ==
0)kirim[0]='2';
    else if (motor1 == 0 && motor2 == 1 && motor3 ==
1)kirim[0]='3';
    else if (motor1 == 1 && motor2 == 0 && motor3 ==
0)kirim[0]='4';
    else if (motor1 == 1 && motor2 == 0 && motor3 ==
1)kirim[0]='5';
    else if (motor1 == 1 && motor2 == 1 && motor3 ==
0)kirim[0]='6';
    else if (motor1 == 1 && motor2 == 1 && motor3 ==
1)kirim[0]='7';

```

```

0);

USART_SendData(USART6, kirim[0]); //header
while(USART_GetFlagStatus(USART6, USART_FLAG_TXE) ==

nilai = nilaiph*100;

switch (nilai/1000)
{
case 0:kirim[0]='0';break;
case 1:kirim[0]='1';break;
case 2:kirim[0]='2';break;
case 3:kirim[0]='3';break;
case 4:kirim[0]='4';break;
case 5:kirim[0]='5';break;
case 6:kirim[0]='6';break;
case 7:kirim[0]='7';break;
case 8:kirim[0]='8';break;
case 9:kirim[0]='9';break;
default:kirim[0]='0';break;
}
USART_SendData(USART6, kirim[0]); //header
while(USART_GetFlagStatus(USART6, USART_FLAG_TXE) ==

0);

//satuan
switch ((nilai%1000)/100)
{
case 0:kirim[0]='0';break;
case 1:kirim[0]='1';break;
case 2:kirim[0]='2';break;
case 3:kirim[0]='3';break;
case 4:kirim[0]='4';break;
case 5:kirim[0]='5';break;
case 6:kirim[0]='6';break;
case 7:kirim[0]='7';break;
case 8:kirim[0]='8';break;
case 9:kirim[0]='9';break;
default:kirim[0]='0';break;
}
USART_SendData(USART6, kirim[0]); //header
while(USART_GetFlagStatus(USART6, USART_FLAG_TXE) ==

0);

//dibelakang koma 1
switch (((nilai%1000)%100)/10)
{
case 0:kirim[0]='0';break;
case 1:kirim[0]='1';break;
case 2:kirim[0]='2';break;
case 3:kirim[0]='3';break;
case 4:kirim[0]='4';break;
case 5:kirim[0]='5';break;
case 6:kirim[0]='6';break;
case 7:kirim[0]='7';break;

```

```

        case 8: kirim[0]='8';break;
        case 9: kirim[0]='9';break;
        default: kirim[0]='0';break;
    }
    USART_SendData(USART6, kirim[0]); //header
    while(USART_GetFlagStatus(USART6, USART_FLAG_TXE) ==
0);

    //dibelakang koma 2
    switch ((nilai%1000)%100)%10)
    {
        case 0: kirim[0]='0';break;
        case 1: kirim[0]='1';break;
        case 2: kirim[0]='2';break;
        case 3: kirim[0]='3';break;
        case 4: kirim[0]='4';break;
        case 5: kirim[0]='5';break;
        case 6: kirim[0]='6';break;
        case 7: kirim[0]='7';break;
        case 8: kirim[0]='8';break;
        case 9: kirim[0]='9';break;
        default: kirim[0]='0';break;
    }
    USART_SendData(USART6, kirim[0]); //header
    while(USART_GetFlagStatus(USART6, USART_FLAG_TXE) ==
0);

        //////
//////////
        delay_ms(80);
    }
}

```

Lampiran 2. Sorce Code Visual Basic

```

Public Class Form1
    Private data1 As Double
    Private datam As Integer

    Private iterasi As Integer

    Dim myPort As Array
    Delegate Sub SetTextCallback(ByVal [text] As String) 'Added to prevent
threading errors during receiveing of data
    Private Sub Form1_Load(sender As System.Object, e As System.EventArgs)
Handles MyBase.Load

        myPort = IO.Ports.SerialPort.GetPortNames()
        ComboBox1.Items.AddRange(myPort)
    End Sub
End Class

```

```

End Sub
'-----

'-----

Private Sub Button1_Click(sender As System.Object, e As System.EventArgs)
Handles Button1.Click

    SerialPort1.PortName = ComboBox1.Text
    SerialPort1.BaudRate = ComboBox2.Text
    SerialPort1.Open() 'perintah untuk membuka port serial
    Button1.Enabled = False
    Button4.Enabled = True

End Sub
'-----

Private Sub Button4_Click(sender As System.Object, e As System.EventArgs)
Handles Button4.Click
    SerialPort1.Close() 'perintah untuk menutup port serial
    Button1.Enabled = True
    Button4.Enabled = False
End Sub

Private Sub SerialPort1_DataReceived(sender As System.Object, e As
System.IO.Ports.SerialDataReceivedEventArgs) Handles SerialPort1.DataReceived
    ReceivedText(SerialPort1.ReadExisting())
End Sub

Private Sub ReceivedText(ByVal [text] As String) 'input from ReadExisting
    'If text.Contains("#") Then
    'text = text.Replace("#", "")
    If IsNumeric(text) Then
        datam = CDb1(text / 10000)
        data1 = CDb1((text Mod 10000) / 100)
        Label1.Invoke(New updatebox(AddressOf update_), data1.ToString)
        Label4.Invoke(New updatebox2(AddressOf update_2), datam.ToString)
        iterasi = iterasi + 1

        If (datam = 0) Then
            Button3.BackColor = Color.Red
            Button5.BackColor = Color.Red
            Button6.BackColor = Color.Red
        ElseIf (datam = 1) Then
            Button3.BackColor = Color.Red
            Button5.BackColor = Color.Red
            Button6.BackColor = Color.Green
        ElseIf (datam = 2) Then
            Button3.BackColor = Color.Red
            Button5.BackColor = Color.Green
            Button6.BackColor = Color.Red
        ElseIf (datam = 3) Then
            Button3.BackColor = Color.Red
            Button5.BackColor = Color.Green
            Button6.BackColor = Color.Green
        ElseIf (datam = 4) Then
            Button3.BackColor = Color.Green
            Button5.BackColor = Color.Red

```



```
        Button6.BackColor = Color.Red
    ElseIf (datam = 5) Then
        Button3.BackColor = Color.Green
        Button5.BackColor = Color.Red
        Button6.BackColor = Color.Green
    ElseIf (datam = 6) Then
        Button3.BackColor = Color.Green
        Button5.BackColor = Color.Green
        Button6.BackColor = Color.Red
    ElseIf (datam = 7) Then
        Button3.BackColor = Color.Green
        Button5.BackColor = Color.Green
        Button6.BackColor = Color.Green

    End If

End If

End Sub

Delegate Sub updatebox(ByVal t As String)
Public Sub update_(ByVal t As String)
    Label1.Text = t
    Me.Chart1.Series("Series1").Points.AddXY(iterasi, Cdbl(t))

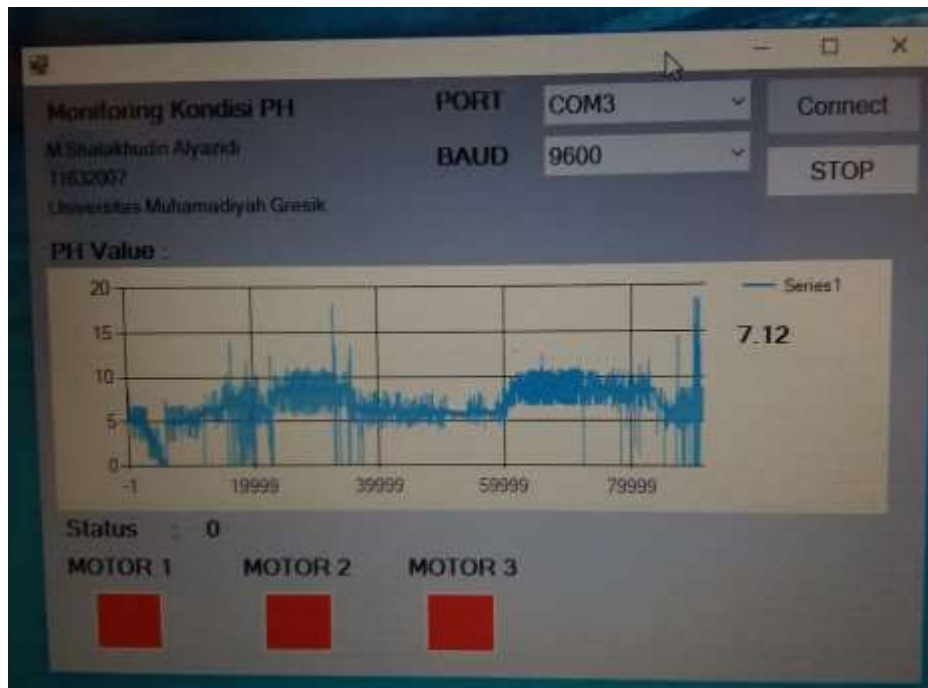
End Sub

Delegate Sub updatebox2(ByVal t As String)
Public Sub update_2(ByVal t As String)
    Label4.Text = t
End Sub

Private Sub Button2_Click(sender As Object, e As EventArgs)

End Sub
End Class
```

Lampiran 3. Gambar Tampilan Monitoring Kadar pH pada PC



Lampiran 4. Gambar Tampilan Hardware Monitoring Kadar pH

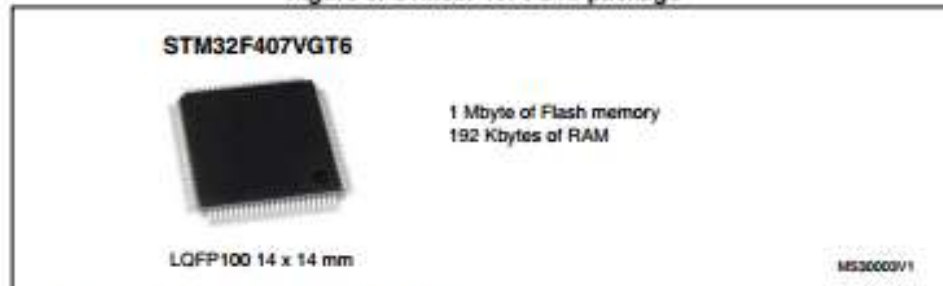


Lampiran 5. Data Sheet ARM-STM32F4

STM32F407VGT6 microcontroller

This ARM Cortex-M4 32-bit MCU with FPU has 210 DMIPS, up to 1 MB Flash/192+4 KB RAM, USB OTG HS/FS, Ethernet, 17 TIMs, 3 ADCs, 15 comm. interfaces and a camera.

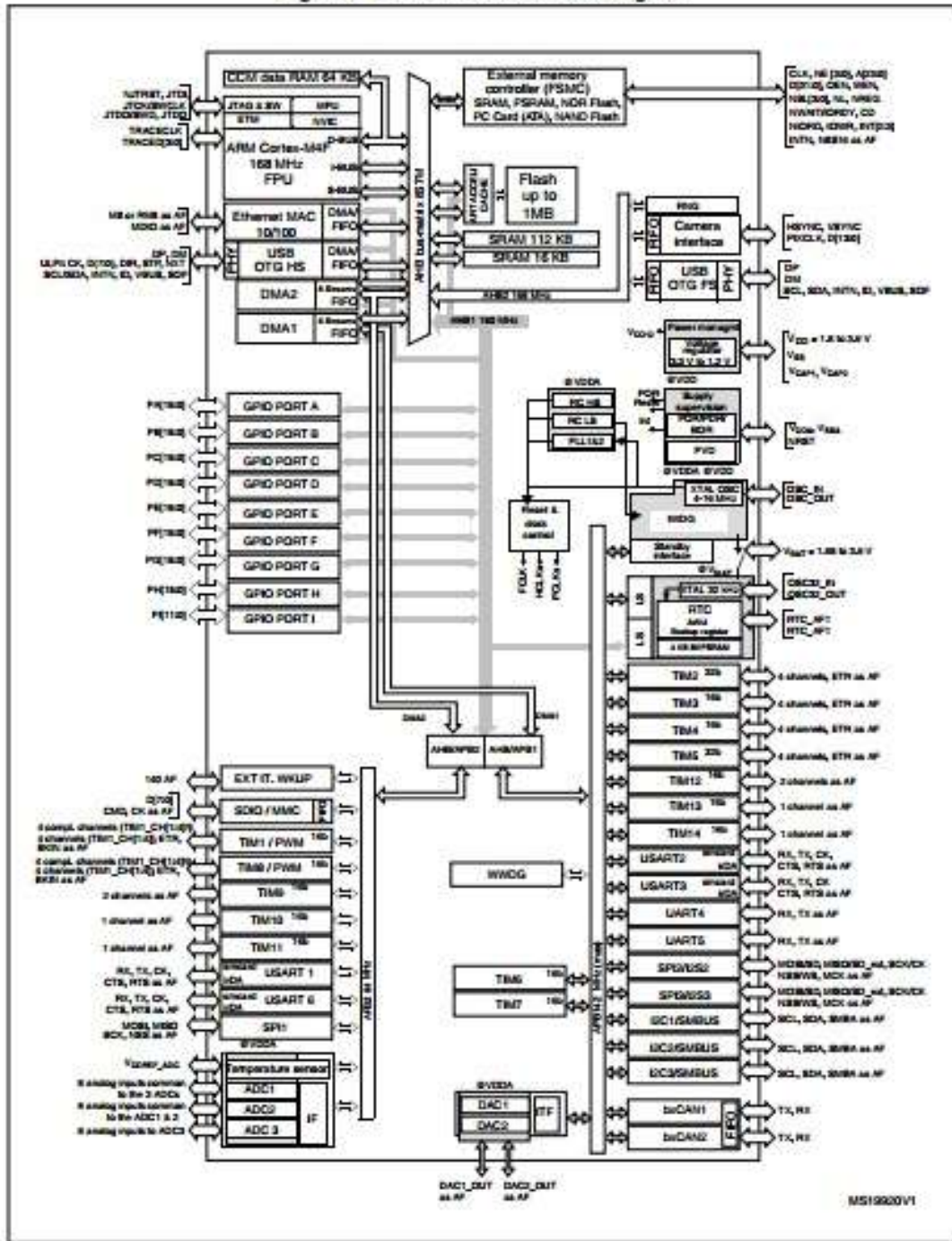
Figure 5. STM32F407VGT6 package



This device provides the following benefits.

- 168 MHz/210 DMIPS Cortex-M4 with single cycle DSP MAC and floating point unit providing:
 - Boosted execution of control algorithms
 - More features possible for your applications
 - Ease of use
 - Better code efficiency
 - Faster time to market
 - Elimination of scaling and saturation
 - Easier support for meta-language tools
- Designed for high performance and ultra fast data transfers; ART Accelerator, 32-bit, 7-layer AHB bus matrix with 7 masters and 8 slaves including 2 blocks of SRAM, Multi DMA controllers: 2 general purpose, 1 for USB HS, 1 for Ethernet, One SRAM block dedicated to the core, providing performance equivalent to 0-wait execution from Flash Concurrent execution and data transfers and simplified resource allocation
- Outstanding power efficiency; Ultra-low dynamic power, RTC <1 μ A typical in VBAT mode, 3.6 V down to 1.7 V VDD, Voltage regulator with power scaling capability, providing extra flexibility to reduce power consumption for applications requiring both high processing and low power performance when running at low voltage or on a rechargeable battery
- Maximum integration: Up to 1 Mbyte of on-chip Flash memory, 192 Kbytes of SRAM, reset circuit, internal RCs, PLLs, WLCSP package available, providing more features in space constrained applications
- Superior and innovative peripherals providing new possibilities to connect and communicate high speed data and more precision due to high resolution
- Extensive tools and software solutions providing a wide choice within the STM32 ecosystem to develop your applications.

Figure 6. STM32F407VGT6 block diagram



4.2 Embedded ST-LINK/V2

The ST-LINK/V2 programming and debugging tool is integrated on the STM32F4DISCOVERY. The embedded ST-LINK/V2 can be used in 2 different ways according to the jumper states (see [Table 2](#)):

- Program/debug the MCU on board,
- Program/debug an MCU in an external application board using a cable connected to SWD connector CN2.

The embedded ST-LINK/V2 supports only SWD for STM32 devices. For information about debugging and programming features refer to user manual UM1075 (*ST-LINK/V2 in-circuit debugger/programmer for STM8 and STM32*) which describes in detail all the ST-LINK/V2 features.

Figure 7. Typical configuration

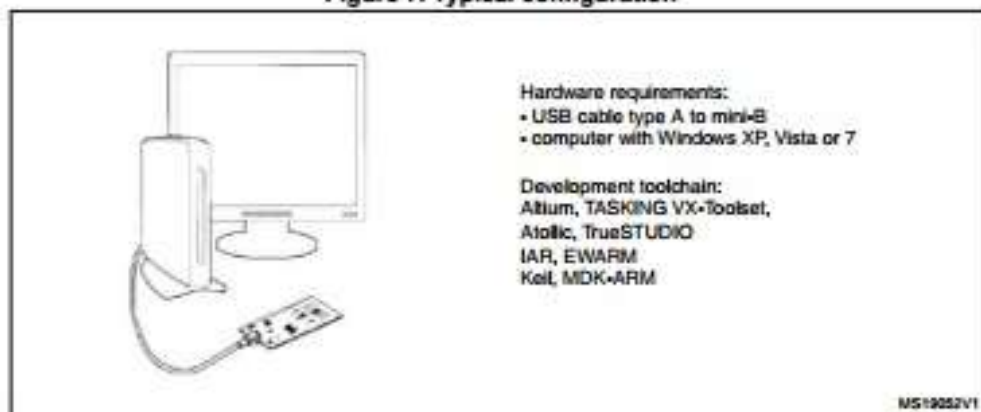


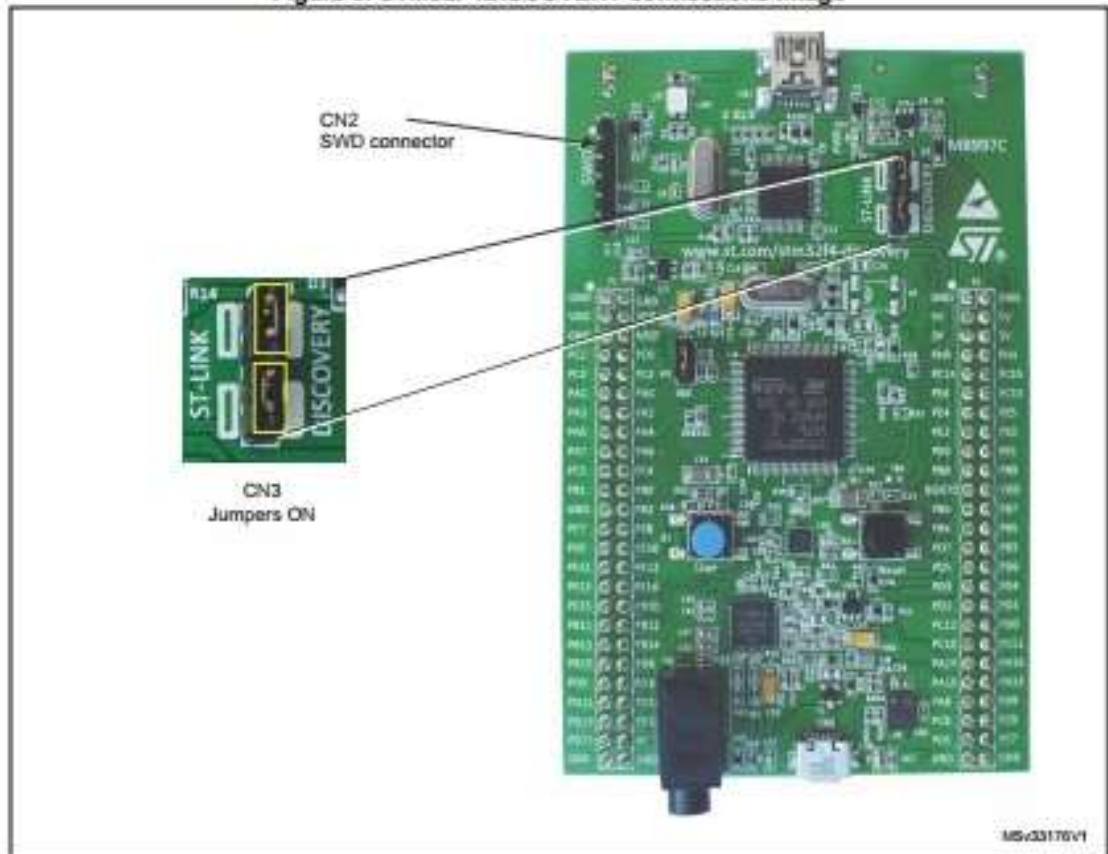
Table 2. Jumper states

Jumper state	Description
Both CN3 jumpers ON	ST-LINK/V2 functions enabled for on board programming (default)
Both CN3 jumpers OFF	ST-LINK/V2 functions enabled for application through external CN2 connector (SWD supported)

4.2.1 Using ST-LINK/V2 to program/debug the STM32F4 on board

To program the STM32F4 on board, simply plug in the two jumpers on CN3, as shown in *Figure 8* in red, but do not use the CN2 connector as that could disturb communication with the STM32F407VGT6 of the STM32F4DISCOVERY.

Figure 8. STM32F4DISCOVERY connections image



4.2.2 Using ST-LINK/V2 to program/debug an external STM32 application

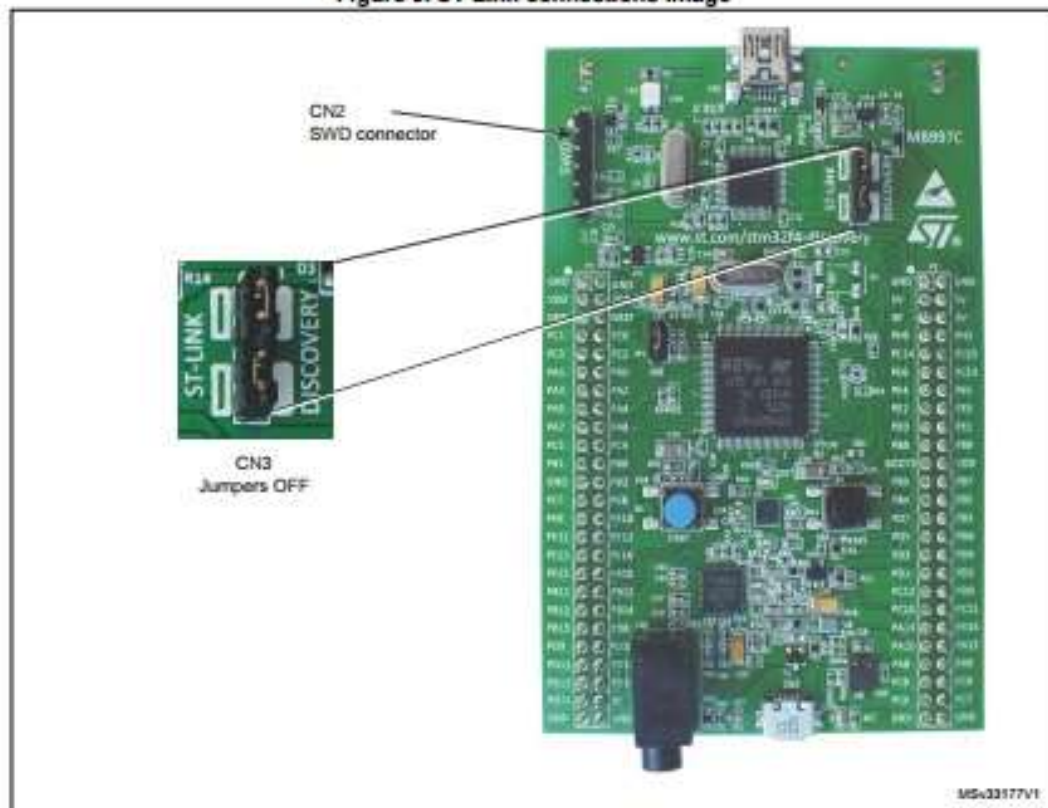
It is very easy to use the ST-LINK/V2 to program the STM32 on an external application. Simply remove the 2 jumpers from CN3 as shown in *Figure 9*, and connect your application to the CN2 debug connector according to *Table 3*.

Note: SB11 must be OFF if you use CN2 pin 5 in your external application.

Table 3. Debug connector CN2 (SWD)

Pin	CN2	Designation
1	VDD_TARGET	VDD from application
2	SWCLK	SWD clock
3	GND	Ground
4	SWDIO	SWD data input/output
5	NRST	RESET of target MCU
6	SWO	Reserved

Figure 9. ST-Link connections image



MS-03177V1

4.3 Power supply and power selection

The power supply is provided either by the host PC through the USB cable, or by an external 5V power supply.

The D1 and D2 diodes protect the 5V and 3V pins from external power supplies:

- 5V and 3V can be used as output power supplies when another application board is connected to pins P1 and P2.
In this case, the 5V and 3V pins deliver a 5V or 3V power supply and power consumption must be lower than 100 mA.
- 5V can also be used as input power supplies e.g. when the USB connector is not connected to the PC.
In this case, the STM32F4DISCOVERY board must be powered by a power supply unit or by auxiliary equipment complying with standard EN-60950-1: 2006+A11/2009, and must be Safety Extra Low Voltage (SELV) with limited power capability.

4.4 LEDs

- LD1 COM: LD1 default status is red. LD1 turns to green to indicate that communications are in progress between the PC and the ST-LINK/V2.
- LD2 PWR: red LED indicates that the board is powered.
- User LD3: orange LED is a user LED connected to the I/O PD13 of the STM32F407VGT6.
- User LD4: green LED is a user LED connected to the I/O PD12 of the STM32F407VGT6.
- User LD5: red LED is a user LED connected to the I/O PD14 of the STM32F407VGT6.
- User LD6: blue LED is a user LED connected to the I/O PD15 of the STM32F407VGT6.
- USB LD7: green LED indicates when VBUS is present on CN5 and is connected to PA9 of the STM32F407VGT6.
- USB LD8: red LED indicates an overcurrent from VBUS of CN5 and is connected to the I/O PD5 of the STM32F407VGT6.

4.5 Pushbuttons

- B1 USER: User and Wake-Up button connected to the I/O PA0 of the STM32F407VGT6.
- B2 RESET: Pushbutton connected to NRST is used to RESET the STM32F407VGT6.

4.6 On board audio capability

The STM32F4 uses an audio DAC (CS43L22) to output sounds through the audio mini jack connector.

The STM32F4 controls the audio DAC through the I2C interface and processes digital signals through I2S connection or analog input signal.

- The sound can come independently from different inputs:
 - ST MEMS microphone (MP45DT02): digital using PDM protocol or analog when using the low pass filter.
 - USB connector: from external mass storage such as a USB key, USB HDD, and so on.
 - Internal memory of the STM32F4.
- The sound can be output in different ways through audio DAC:
 - Using I2S protocol
 - Using the STM32F4 DAC to analog input AIN1x of the CS43L22
 - Using the microphone output directly via a low pass filter to analog input AIN4x of the CS43L22

4.7 USB OTG supported

The STM32F4 is used to drive only USB OTG full speed on this board. The USB micro-AB connector (CN5) allows the user to connect a host or device component, such as a USB key, mouse, and so on.

Two LEDs are dedicated to this module:

- LD7 (green LED) indicates when VBUS is active
- LD8 (red LED) indicates an overcurrent from connected device

4.8 Motion sensor (ST MEMS LIS302DL or LIS3DSH)

Two different versions of motion sensor (U5 in schematic) are available on the board depending the PCB version. The LIS302DL is present on board MB997B (PCB revision B) and the LIS3DSH is present on board MB997C (PCB rev C).

The LIS302DL or LIS3DSH are both an ultra compact low-power three-axis linear accelerometer.

It includes a sensing element and an IC interface able to provide the measured acceleration to the external world through I2C/SPI serial interface.

The LIS302DL has dynamically user selectable full scales of $\pm 2g/\pm 8g$ and it is capable of measuring acceleration with an output rate of 100Hz to 400Hz.

The LIS3DSH has $\pm 2g/\pm 4g/\pm 6g/\pm 8g/\pm 16g$ dynamically selectable full-scale and it is capable of measuring acceleration with an output data rate of 3.125 Hz to 1.6 kHz.

4.9 JP1 (Idd)

Jumper JP1, labeled Idd, allows the consumption of STM32F407VGT6 to be measured by removing the jumper and connecting an ammeter.

- Jumper on: STM32F407VGT6 is powered (default).
- Jumper off: an ammeter must be connected to measure the STM32F407VGT6 current, (if there is no ammeter, the STM32F407VGT6 is not powered).

4.10 OSC clock

4.10.1 OSC clock supply

If PH0 and PH1 are only used as GPIOs instead of as a clock, then SB13 and SB14 are closed and R24, R25 and R68 are removed.

- **MCO from ST-LINK.** From MCO of the STM32F103. This frequency cannot be changed, it is fixed at 8 MHz and connected to PH0-OSC_IN of the STM32F407VGT6. Configuration needed:
 - SB13, SB14 OPEN
 - R25^(a) removed
 - R68^(a) soldered
- **Oscillator onboard.** From X2 crystal. For typical frequencies and its capacitors and resistors, please refer to the STM32F407VGT6 Datasheet. Configuration needed:
 - SB13, SB14 OPEN
 - R25^(a) soldered
 - R68^(a) removed
- **Oscillator from external PH0.** From external oscillator through pin 7 of the P2 connector. Configuration needed:
 - SB13 closed
 - SB14 closed
 - R25 and R68 removed

4.10.2 OSC 32 KHz clock supply

If PC14 and PC15 are only used as GPIOs instead of as a clock, then SB15 and SB16 are closed, and R21 and R22 are removed.

- **Oscillator onboard.** From X1 Crystal (not provided). Configuration needed:
 - SB15, SB16 OPEN
 - C16, C27, R21 and R22 soldered.
- **Oscillator from external PC14.** From external oscillator through the pin 9 of P2 connector. Configuration needed:
 - SB16 closed
 - SB15 closed
 - R21 and R22 removed

4.11 Solder bridges

Table 4. Solder bridges

Bridge	State ⁽¹⁾	Description
SB13,14 (X2 crystal) ⁽²⁾	OFF	X2, C14, C15, R24 and R25 provide a clock. PH0, PH1 are disconnected from P2.
	ON	PH0, PH1 are connected to P2 (R24, R25 and R68 must not be fitted).
SB3,5,7,9 (Default)	ON	Reserved, do not modify.
SB2,4,6,8 (Reserved)	OFF	Reserved, do not modify.
SB15,16 (X3 crystal)	OFF	X3, C16, C27, R21 and R22 deliver a 32 KHz clock. PC14, PC15 are not connected to P2.
	ON	PC14, PC15 are only connected to P2. Remove only R21, R22
SB1 (B2-RESET)	ON	B2 pushbutton is connected to the NRST pin of the STM32F407VGT6 MCU.
	OFF	B2 pushbutton is not connected the NRST pin of the STM32F407VGT6 MCU.
SB20 (B1-USER)	ON	B1 pushbutton is connected to PA0.
	OFF	B1 pushbutton is not connected to PA0.
SB17 (VDD powered from 3V)	OFF	VDD is not powered from 3V, depends on JP1 jumper.
	ON	VDD is permanently powered from 3V, JP1 jumper has no effect.
SB11 (NRST)	ON	NRST signal of the CN2 connector is connected to the NRST pin of the STM32F407VGT6 MCU.
	OFF	NRST signal of the CN2 connector is not connected to the NRST pin of the STM32F407VGT6 MCU.
SB12 (SWO)	ON	SWO signal of the CN2 connector is connected to PB3.
	OFF	SWO signal is not connected.
SB10 (STM_RST)	OFF	No incidence on STM32F103C8T6 (ST-LINK/V2) NRST signal.
	ON	STM32F103C8T6 (ST-LINK/V2) NRST signal is connected to GND.
SB18 (BOOT0)	ON	BOOT0 signal of the STM32F407VGT6 MCU is held low through a 510 ohm pull-down resistor.
	OFF	BOOT0 signal of the STM32F407VGT6 MCU is held high through a 10 Kohm pull-up resistor.
SB19 (BOOT1)	OFF	The BOOT1 signal of the STM32F407VGT6 MCU is held high through a 10 Kohm pull-up resistor.
	ON	The BOOT1 signal of the STM32F407VGT6 MCU is held low through a 510 ohm pull-down resistor.

1. Default SBx state is shown in bold.

2. SB13 and SB14 are OFF to allow the user to choose between MCO and X2 crystal for clock source.

Table 5. MCU pin description versus board function (continued)

MCU pin		Board function															
Main function	Alternate functions	LOFP100	CS43122	MP45DT02	LIS302DL or LIS3DSH	Pushbutton	LED	SWD	USB	OSC	From I/O	Power supply	ONS	ON2	P1	P2	
PE4	TRACED1/ FSMC_A20/ DCMI_D4	3	*	*	*	*	*	*	*	*	*	*	*	*	*	*	13
PE5	TRACED2/ FSMC_A21/ TIM9_CH1/ DCMI_D6	4	*	*	*	*	*	*	*	*	*	*	*	*	*	*	14
PE6	TRACED3/ FSMC_A22/ TIM9_CH2/ DCMI_D7	5	*	*	*	*	*	*	*	*	*	*	*	*	*	*	11
PE7	FSMC_D4/ TIM1_ETR	38	*	*	*	*	*	*	*	*	*	*	*	*	*	25	*
PE8	FSMC_D5/ TIM1_CH1N	39	*	*	*	*	*	*	*	*	*	*	*	*	*	26	*
PE9	FSMC_D6/ TIM1_CH1	40	*	*	*	*	*	*	*	*	*	*	*	*	*	27	*
PE10	FSMC_D7/ TIM1_CH2N	41	*	*	*	*	*	*	*	*	*	*	*	*	*	28	*
PE11	FSMC_D8/ TIM1_CH2	42	*	*	*	*	*	*	*	*	*	*	*	*	*	29	*
PE12	FSMC_D9/ TIM1_CH3N	43	*	*	*	*	*	*	*	*	*	*	*	*	*	30	*
PE13	FSMC_D10/ TIM1_CH3	44	*	*	*	*	*	*	*	*	*	*	*	*	*	31	*
PE14	FSMC_D11/ TIM1_CH4	45	*	*	*	*	*	*	*	*	*	*	*	*	*	32	*
PE15	FSMC_D12/ TIM1_BKIN	46	*	*	*	*	*	*	*	*	*	*	*	*	*	33	*
PH0	OSC_IN	12	*	*	*	*	*	*	*	OSC_IN	*	*	*	*	*	*	7

Table 5. MCU pin description versus board function (continued)

MCU pin		Board function														
Main function	Alternate functions	LOFP100	CS43L22	MP45DT02	LIS302DL or LIS3DSH	Pushbutton	LED	SWD	USB	OSC	Free I/O	Power supply	GN5	GN2	P1	P2
PH1	OSC_OUT	13	*	*	*	*	*	*	*	OSC_OUT	*	*	*	*	*	8
*	*	*	*	*	*	*	*	*	*	*	*	5V	*	*	*	3
*	*	*	*	*	*	*	*	*	*	*	*	5V	*	*	*	4
*	*	*	*	*	*	*	*	*	*	*	*	3V	*	*	*	5
*	*	*	*	*	*	*	*	*	*	*	*	3V	*	*	*	6
*	*	*	*	*	*	*	*	*	*	*	*	VDD	*	*	3	*
*	*	*	*	*	*	*	*	*	*	*	*	VDD	*	*	4	*
*	*	*	*	*	*	*	*	*	*	*	*	VDD	*	*	*	22
*	*	*	*	*	*	*	*	GND	GND	*	*	GND	5	3	1	*
*	*	*	*	*	*	*	*	*	*	*	*	GND	*	*	2	*
*	*	*	*	*	*	*	*	*	*	*	*	GND	*	*	5	*
*	*	*	*	*	*	*	*	*	*	*	*	GND	*	*	23	*
*	*	*	*	*	*	*	*	*	*	*	*	GND	*	*	49	*
*	*	*	*	*	*	*	*	*	*	*	*	GND	*	*	50	*
*	*	*	*	*	*	*	*	*	*	*	*	GND	*	*	*	1
*	*	*	*	*	*	*	*	*	*	*	*	GND	*	*	*	2

Lampiran 6. Data Sheet pH Meter

pH Sensor Test Procedure

The purpose of this test is to determine if a pH electrode is functioning within acceptable limits. The asymmetry potential (AP) and slope (efficiency) can be used as guidelines to judge an electrode's performance. Typically an electrode is replaced when the AP is greater than ± 40 mV and/or the slope drops below 85%. Consideration should also be given to the electrodes speed of response. Please follow this step by step procedure to determine the performance of an electrode. Required test equipment includes 7.00 and 4.01 pH buffer solutions with a pH meter that has a mV readout.

1. Set the pH/mV switch on the pH meter to the mV position.
 2. Connect a shorting plug to the input on the pH meter, or connect a precision mV generator with a 0 mV input. Adjust the standardize/zero control on the pH meter for a reading equal to 0.0 mV.
 3. Disconnect the shorting plug/precision mV generator, and connect the electrode that will be tested.
 4. Rinse the electrode thoroughly with DI water to remove all traces of storage solution, process medium, or previous test solution. Thoroughly rinse the electrode after each buffer test to prevent carry over contamination of the pH buffer solutions. Gently blot the electrode on a soft tissue to remove the excess rinse water. Do not rub the bulb since it can cause a static charge build-up.
 5. Insert the electrode and the automatic temperature compensator (ATC) in 7.00 pH buffer solution. Allow 30 seconds for the electrode/ATC to reach thermal equilibrium with the buffer solution. Record the polarity and the mV reading. This is the asymmetry potential of the electrode. A perfect electrode would have an AP equal to 0 mV, but most electrodes read between ± 30 mV.
- Note: If the meter does not have an ATC then place a thermometer along with the electrode in the 7.00 pH buffer solution. Allow 30 seconds for the pair to reach thermal equilibrium with the buffer. Adjust the temperature setting on the meter to correspond with the thermometer reading. Record the polarity and the mV reading to determine the AP.
6. Repeat step 4, and insert the electrode and the ATC in a 4.01 buffer solution. Allow 30 seconds before recording the mV reading.

Continued

Measurement and Control Products for Science and Industry
19 Thomas, Irvine, California 92618 USA
Phone: 949.829.5555 Toll-Free: 800.288.2833
Fax: 949.829.5960 E-Mail: sales@broadleyjames.com



pH Sensor Test Procedure

7. Determine the mathematical difference between the two mV readings. This is the electrode's span.
8. Divide the electrode's span by the theoretical span of 176.9 mV (at 250C) and multiply by 100. This determines the slope of the electrode.

Note: For best results, the pH buffers solutions should be used at 250C. Otherwise, record the temperature of the buffer and determine the temperature adjusted pH with the temperature coefficient charts printed on the buffer container.

Example:

Reading in a 7.00 pH buffer solution: mV (max/min range ± 40 mV)

Reading in a 4.01 pH buffer solution: mV

$$\text{Span} = \frac{164.6}{\text{(Reading in 4.01 pH)}} \text{ mV} - \frac{-7.4}{\text{(Asymmetry Potential)}} \text{ mV} = \underline{172.0} \text{ mV (min 150.4 mV)}$$

$$\text{Slope} = \frac{\text{Span}}{\text{Theoretical Span}} = \frac{172.0}{176.9} \text{ mV} \times 100 = \underline{97} \% \text{ (min 85\%)}$$

Electrode Test Result Calculator:

Reading in a 7.00 pH buffer solution: mV (max/min range ± 40 mV)

Reading in a 4.01 pH buffer solution: mV

$$\text{Span} = \frac{0.0}{\text{(Reading in 4.01 pH)}} \text{ mV} - \frac{0.0}{\text{(Asymmetry Potential)}} \text{ mV} = \underline{0.0} \text{ mV (min 150.4 mV)}$$

$$\text{Slope} = \frac{\text{Span}}{\text{Theoretical Span}} = \frac{0.0}{176.9} \text{ mV} \times 100 = \underline{0} \% \text{ (min 85\%)}$$

Date Tested: _____ Initial: _____ Identification: _____
(Electrode P/N, Vessel Location/No.)



Measurement and Control Products for Science and Industry

19 Thomas, Irvine, California 92618 USA

Phone: (949) 829-5555 Toll-Free: (800) 288-2833 Fax: (949) 829-5560

E-Mail: sales@broadleyjames.com Website: www.broadleyjames.com

Lampiran 7. Data Sheet Bluetooth HC-05

-Bluetooth to Serial Port Module

Overview



HC-05 module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup.

Serial port Bluetooth module is fully qualified Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps Modulation with complete 2.4GHz radio transceiver and baseband. It uses CSR Bluecore O4-External single chip Bluetooth system with CMOS technology and with AFH(Adaptive Frequency Hopping Feature). It has the footprint as small as 12.7mmx27mm. Hope it will simplify your overall design/development cycle.

Specifications

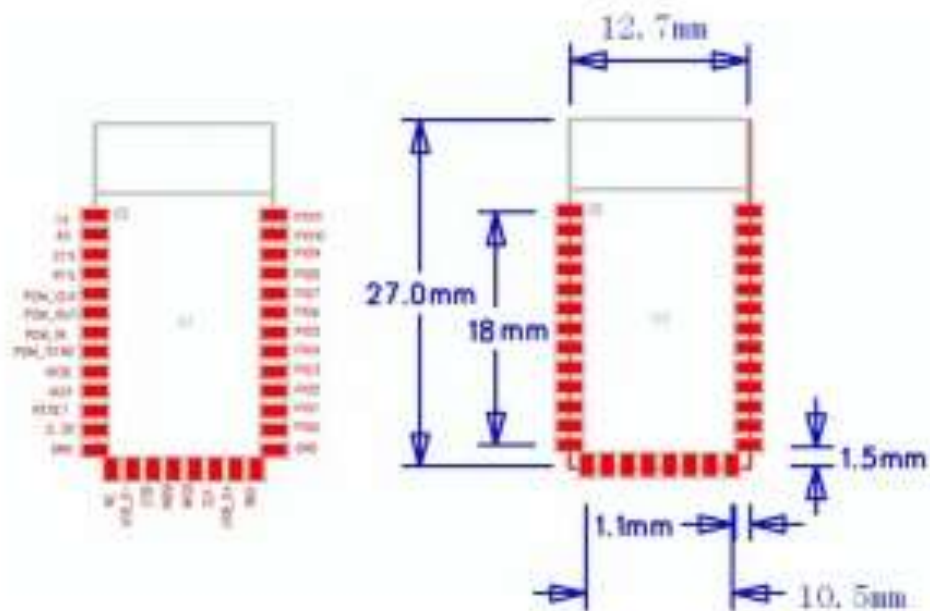
Hardware features

- Typical -80dBm sensitivity
- Up to +4dBm RF transmit power
- Low Power 1.8V Operation ,1.8 to 3.6V I/O
- PIO control
- UART interface with programmable baud rate
- With integrated antenna
- With edge connector

Software features

- Default Baud rate: 38400, Data bits:8, Stop bit:1,Parity:No parity, Data control: has. Supported baud rate: 9600,19200,38400,57600,115200,230400,460800.
- Given a rising pulse in PIO0, device will be disconnected.
- Status instruction port PIO1: low=disconnected, high=connected;
- PIO10 and PIO11 can be connected to red and blue led separately. When master and slave are paired, red and blue led blinks 1time/2s in interval, while disconnected only blue led blinks 2times/s.
- Auto-connect to the last device on power as default.
- Permit pairing device to connect as default.
- Auto-pairing PINCODE:"0000" as default
- Auto-reconnect in 30 min when disconnected as a result of beyond the range of connection.

Hardware



PIN Name	PIN #	Pad type	Description	Note
GND	13 21 22	VSS	Ground pot	
3.3 VCC	12	3.3V	Integrated 3.3V (+) supply with On-chip linear regulator output within 3.15-3.3V	
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	
PIO7	30	Bi-Directional	Programmable input/output line	
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 internal pull-up	Reset if low input debounced so must be low for >5MS to cause a reset	
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 pull-up	Chip select for serial peripheral interface, active low	
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		

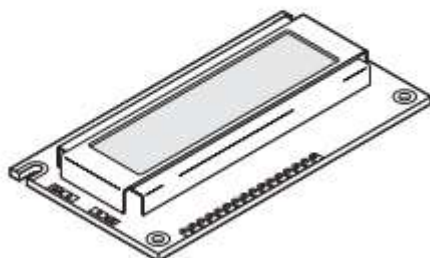
USB_+	20	Bi-Directional		
NC	14			
PCM_CLK	5	Bi-Directional	Synchronous PCM data clock	
PCM_OUT	6	CMOS output	Synchronous PCM data output	
PCM_IN	7	CMOS Input	Synchronous PCM data input	
PCM_SYNC	8	Bi-Directional	Synchronous PCM data strobe	

Lampiran 8. Data Sheet LCD 16x2

ALPHANUMERIC LCD DISPLAY (16 x 2)

Order Code

LED008 16 x 2 Alphanumeric Display
FRM010 Serial LCD Firmware (optional)



Contents

1 x 16x2 Alphanumeric Display
1 x data booklet

Introduction

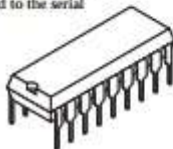
Alphanumeric displays are used in a wide range of applications, including palmtop computers, word processors, photocopiers, point of sale terminals, medical instruments, cellular phones, etc. The 16 x 2 intelligent alphanumeric dot matrix display is capable of displaying 224 different characters and symbols. A full list of the characters and symbols is printed on pages 7/8 (note these symbols can vary between brand of LCD used). This booklet provides all the technical specifications for connecting the unit, which requires a single power supply (+5V).

Further Information

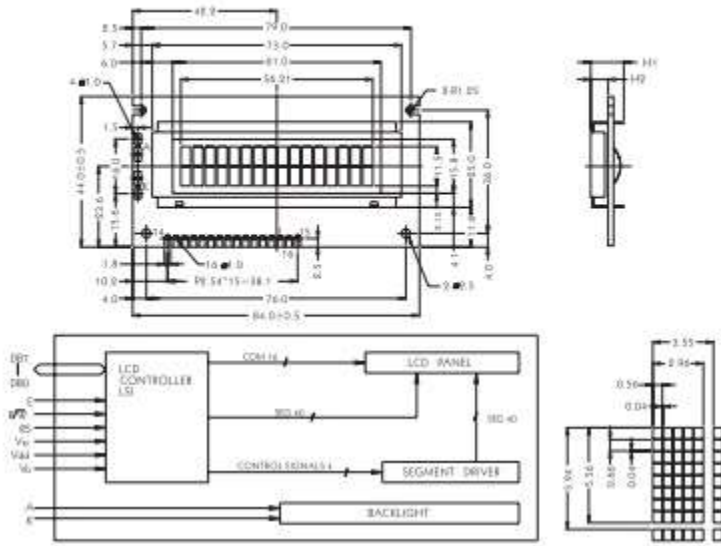
Available as an optional extra is the Serial LCD Firmware, which allows serial control of the display. This option provides much easier connection and use of the LCD module. The firmware enables microcontrollers (and microcontroller based systems such as the PICAXE) to visually output user instructions or readings onto an LCD module. All LCD commands are transmitted serially via a single microcontroller pin. The firmware can also be connected to the serial port of a computer.

An example PICAXE instruction to print the text 'Hello' using the `serout` command is as follows:

```
serout 7,T2400, ("Hello")
```



Outline Dimension and Block Diagram



The tolerance unless classified: ±0.3mm

MECHANICAL SPECIFICATION			
Covered Size	84.0*44.0	Module	H0.7/H1
Visible Area	61.0*15.8	W/L/381	3.1/3.1
Dot Size	0.36*0.36	DL/BL	3.1/3.1
Dot Pitch	0.80*0.70	LED/BL	3.4/14.0

PIN ASSIGNMENT		
Pin no.	Symbol	Function
1	V _{cc}	Power supply (0.5V)
2	V _{dd}	Power supply (+5V)
3	V _e	Control Pulse
4	RS	Register select (panel)
5	R/W	Data read/Write
6	↑	Enable signal
7	DB0	Data bus line
8	DB1	Data bus line
9	DB2	Data bus line
10	DB3	Data bus line
11	DB4	Data bus line
12	DB5	Data bus line
13	DB6	Data bus line
14	DB7	Data bus line
15	A	Power supply for LED BL (1.4)
16	E	Power supply for LED BL (1.4)

ABSOLUTE MAXIMUM RATING					
Item	Symbol	Conditions	Min.	Max.	Unit
Power Supply Voltage	V _{cc} -V _e	-	0	7	V
LCD Drive Supply Voltage	V _{dd} -V _e	-	0	13	V
Input Voltage	V _{in}	-	+0.5	V _{cc} +0.2	V
Operating Temperature	T _{op}	Typ	0	50	°C
Storage Temperature	T _{stg}	Typ	-20	+70	°C

ELECTRICAL CHARACTERISTICS (V _{dd} = +5V, T _a = 25°C)						
Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Logic Supply Voltage	V _{cc}	-	4.5	5	5.5	V
V _H Input Voltage	V _H	-	2.2	-	-	V
V _L Input Voltage	V _L	-	-	-	0.5	V
V _H Output Voltage	V _H	-	2.4	-	-	V
V _L Output Voltage	V _L	-	-	-	0.4	V
Supply Current	I _{cc}	-	2	-	-	mA
LCD Drive Voltage	V _{cc} -V _e	4.3	-	4.5	V	

Electrical Characteristics

V_{DD} = 5V±5%
V_{SS} = 0V

Item	Symbol	Condition	Standard value			Unit	Applicable terminal
			Min.	Typ.	Max.		
Power voltage	V _{DD}		4.5	5.00	5.5	V	V _{DD}
Input H - level voltage	V _{IH}		2.8	—	V _{DD}	V	RS, R/W, E DB0—DB7
Input L - level voltage	V _{IL}		-0.3	—	0.6	V	
Output H - level voltage	V _{OH}	I _{OH} = 0.205mA	2.4	—	—	V	DB0—DB7
Output L - level voltage	V _{OL}	I _{OL} = 1.8mA	—	—	0.4	V	
I/O leakage current	I _{IL}	V _{IE} = 0 - V _{DD}	-1	—	1.0	μA	RS, R/W, E DB0—DB7
Supply current	I _{DD}	V _{DD} = 5V	2	—	—	mA	V _{DD}
LCD operating voltage	V _{LCD}	V _{DD} = V _D	3.0	—	11.0	V	V _D

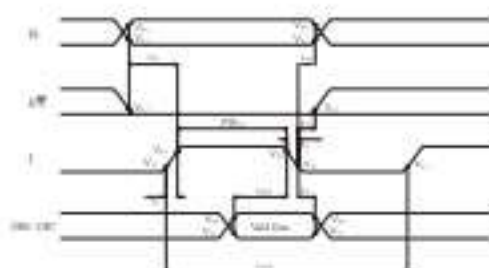
Timing Characteristics

V_{DD} = 5V±5%
V_{SS} = 0V

Item	Symbol	Min.	Max.	Unit
Enable cycle time	T _{CEX}	500	—	ns
Enable pulse width	P _{WH} ("High" level)	200	—	ns
Enable rise/fall time	T _{ra} , T _{rf}	—	25	ns
Set-up time	T _{su} (RS, R/W, E)	40	—	ns
Address hold time	T _{ah}	10	—	ns
Data set-up time	T _{dsu}	60	—	ns
Data delay time	T _{dd}	60	190	ns
Data hold time (writing)	T _{wh}	10	—	ns
Data hold time (reading)	T _{whr}	20	—	ns
Clock oscillating frequency	T _{osc}	270 (T _{WH})		KHz

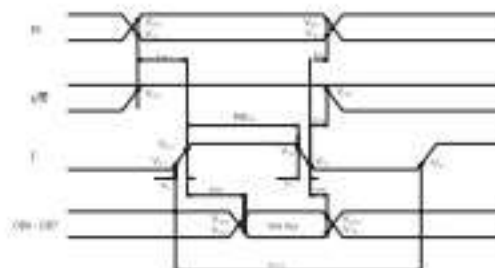
Timing Chart

◆ FIG. 1 WRITE OPERATION



(Note: See Fig. 100 for I/O pin locations.)

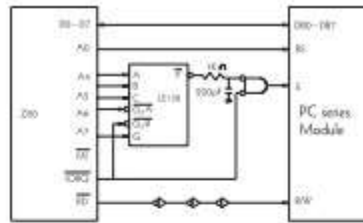
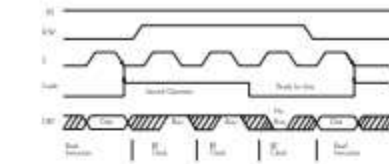
◆ FIG. 2 READ OPERATION



(Note: See Fig. 100 for I/O pin locations.)

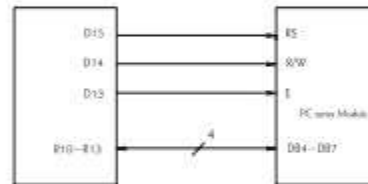
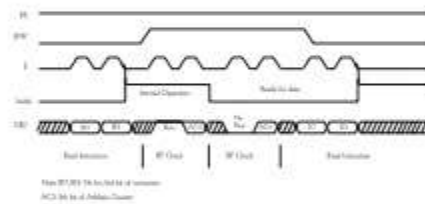
Interface with MPU

◆ **Example of interface with 8-bit MPU (Z80)**



◆ **Example of interface with 4-bit MPU**

Interface with 4-bit MPU can be made through I/O port of 4-bit MPU. If there are enough I/O ports, data can be transferred by 8-bit, however, if there are not data transfer can be done by 4-bit in twice (select interface is 4-bit long), and timing sequence will be complicated in this case. Please take into account that 2 cycles of BF check is necessary, while 2 cycles of data transfer are also necessary.



Features

- (1) Interface with 8-bit or 4-bit MPU is available.
 - (2) 192 kind of alphabets, numerals, symbols and special characters can be displayed by built-in character generator (ROM).
 - (3) Other preferred characters can be displayed by character generator (RAM).
 - (4) Various functions of instruction are available by programming.
 - Clear display • Cursor at home • On / off cursor
 - Blink character • Shift display • Shift cursor
 - Read / write display data...etc.
 - (5) Compact and light weight design which can be easily mounted in device.
 - (6) Single power supply +5V drive (except for extended temp. type).
 - (7) Low power consumption.
- ◆ Interface between data bus line and 4-bit or 8-bit MPU is available.
Data transfer are made in twice in case of 4-bit MPU, and once in case of 8-bit MPU.

◆ **If interface data is 4-bit long**

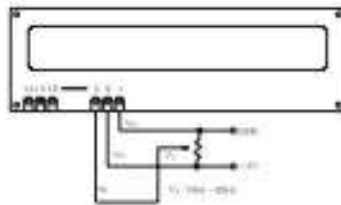
Data transfer are made through 4 bus lines from DB4 to DB7. (while the rest of 4 bus lines from DB0 to DB3 are not used.) Data transfer with MPU are completed when 4-bit data are transferred in twice. (first upper 4-bit data, then lower 4-bit data.)

◆ **If interface data is 8-bit long**

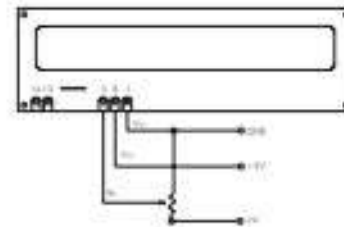
Data transfer are made through all of 8 bus lines from DB0 to DB7.

Example of Power Supply

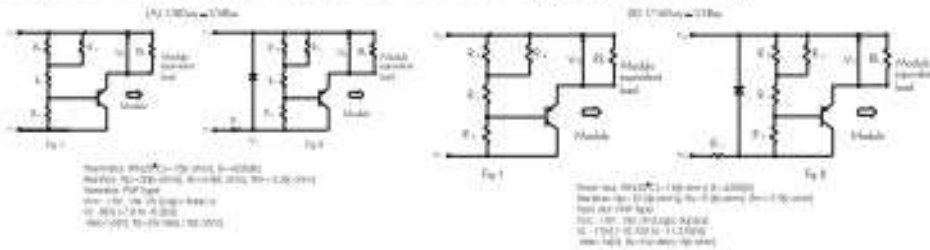
• Normal Temperature Type



• Extended Temperature Type



• Examples of Temperature Compensation Circuits for Extended Temp Type. (Only for reference)



Instructions

Instruction	Code										Description	Executed Time(max.)
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear Display	0	0	0	0	0	0	0	0	0	1	Clears all display and returns the cursor to the home position (Address 0).	1.04μS
Cursor At Home	0	0	0	0	0	0	0	0	0	*	Returns the cursor to the home position (Address 0). Also returns the display being shifted to the original position. DD RAM contents remain unchanged.	1.04μS
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S	Sets the cursor move direction and specifies or not to shift the display. These operations are performed during data write and read.	40μS
Display On/Off Control	0	0	0	0	0	0	1	D	C	B	Sets ON/OFF of all display (D), cursor ON/OFF (C), and blank of cursor position character (B).	40μS
Cursor / Display Shift	0	0	0	0	0	1	S/C	R/L	*	*	Moves the cursor and shifts the display without changing DD RAM contents.	40μS
Function Set	0	0	0	0	1	DL	N	F	*	*	Sets interface data length (DL), number of display lines (L), and character font (F).	40μS
CG RAM Address Set	0	0	0	1	ACG					Sets the CG RAM address. CG RAM data is sent and received after this setting.	40μS	
DD RAM Address Set	0	0	1	ADD					Sets the DD RAM address. DD RAM data is sent and received after this setting.	40μS		
Busy Flag / Address Read	0	1	BF	AC					Reads Busy flag (BF) indicating internal operation is being performed and reads address counter count.	0μS		
CG RAM / DD RAM Data Write	1	0	WRITE DATA					Writes data into DD RAM or CG RAM.	40μS			
CG RAM / DD RAM Data Read	1	1	READ DATA					Reads data from DD RAM or CG RAM.	40μS			

Code		Description	Executed Time (max)
I/D = 1: Increment	DL = 0: 4-bit	DD RAM: Display Data RAM	f _{clk} or f _{osc} = 250KHz. However, when frequency changes, execution time also changes. Example: if f _{clk} or f _{osc} is 270KHz, 70μS × 910 / 270 = 23μS
I/D = 0: Decrement	N = 1: 2lines	CG RAM: Character Generator RAM	
S = 1: Walk display shift	N = 0: 1line	ACG: CG RAM Address	
S/C = 0: cursor movement	F = 1.5×10dots	ADD: DD RAM Address Corresponds to cursor address.	
R/L = 1: Shift to the right	F = 0.5×7dots	AC: Address Counter, used for both DD RAM and CG RAM	
R/L = 0: Shift to the left	BF = 1: Internal operation is being performed.		*: Inverted
DL = 1: 8-bit	BF = 0: Instruction acceptable		

Power Supply Reset

The internal reset circuit will be operated properly when the following power supply conditions are satisfied. If it is not operated properly, please perform initial setting along with the instruction.

Item	Symbol	Measuring Condition	Standard Value			Unit
			Min.	Typ.	Max.	
Power Supply RISE Time	t _{rise}	—	0.1	—	1.0	μs
Power Supply CFF Time	t _{off}	—	1	—	—	μs

Reset function

Initialization Made by Internal Reset Circuit

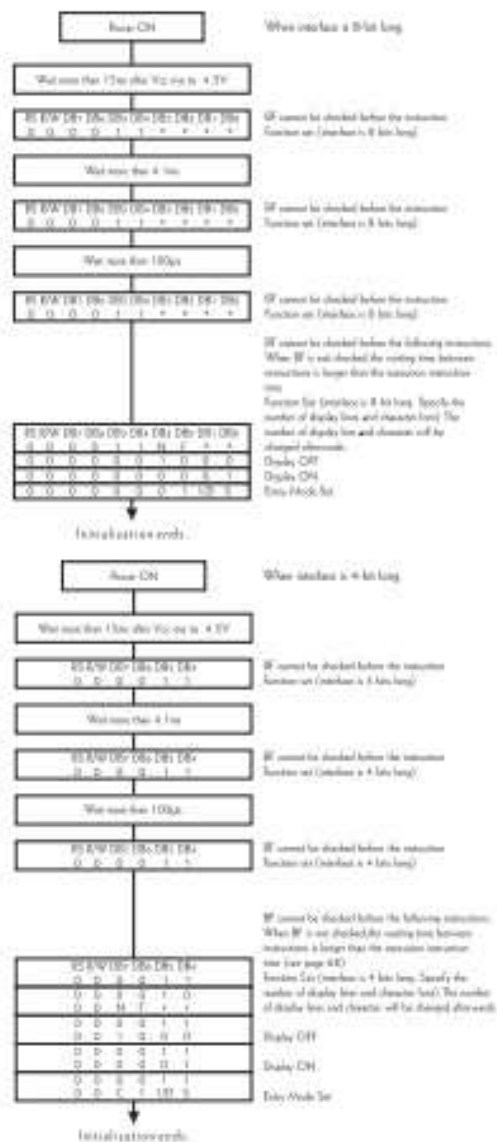
HD44780 automatically initializes (resets) when power is supplied (builtin internal reset circuit). The following instructions are executed in initialization. The busy flag (BF) is kept in busy state until initialization ends. (BF=1) The busy state is 10 ms after V_{dd} reaches to 4.5V.

- (1) Display clear
DL = 1: 8 bit long interface data
DL = 0: 4 bit F= 0.5 x 7dots character font
N = 1: 9 lines
N = 0: 1 line
- (2) Function set
D = 0: Display OFF C = 0: Cursor OFF
B = 0: Blink OFF
- (3) Display ON / OFF control
I / D = 1 + 1 (normal) S = 0: No shift

Note: When conditions stated in power supply conditions using internal reset circuit are not satisfied, the internal reset circuit will not operate properly and initialization will not be performed. Please make initialization using MPU along with instruction.

Initialization along with instruction

If power supply conditions are not satisfied, which for proper operation of internal reset circuit, it is required to make initialization along with instruction. Please make following procedures.



Standard Character Pattern (Powertip Module)

		Higher 4-bit (D4 to Character Code (Hexadecimal))																
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
Lower 4-bit (D0 to D3) of Character Code (Hexadecimal)	0	CG RAM (1)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	1	CG RAM (2)	:	!	@"	#	\$	%	&	'	()	*	+	,	-	.	/
	2	CG RAM (3)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	3	CG RAM (4)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	4	CG RAM (5)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	5	CG RAM (6)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	6	CG RAM (7)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	7	CG RAM (8)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	8	CG RAM (1)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	9	CG RAM (2)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	A	CG RAM (3)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	B	CG RAM (4)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	C	CG RAM (5)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	D	CG RAM (6)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	E	CG RAM (7)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	F	CG RAM (8)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Standard Character Pattern (Elec & Eltek Module)

Upper (200)	Lower (200)	LLLL	LLHL	LLHH	LHLL	LHLH	LHHL	LHHH	HLLL	HLLH	HLHL	HLHH	HHLL	HHLH	HHHL	HHHH	
CG RAM (1)	LLLL		0	1	2	3	4	5					一	夕	三	8	0
(2)	LLHH	!	1	Q	W	9					。	ア	チ	ム	ヨ	9	
(3)	LLHL	"	2	R	5						「	ウ	テ	モ	0	0	
(4)	LLHH	#	3	S	6						」	ウ	テ	モ	0	8	
(5)	LHLL	\$	4	T	7						ノ	工	ト	カ	4	9	
(6)	LHLH	%	5	U	8						。	オ	大	1	0	0	
(7)	LHHL	&	6	V	9						ヲ	カ	ニ	ヨ	0	M	
(8)	LHHH	'	7	W	0						ヲ	キ	又	ラ	9	8	
(1)	HLLL	(8	X	1						イ	ウ	ホ	リ	4	8	
(2)	HLLH)	9	Y	2						ウ	ケ	ル		4	0	
(3)	HLHL	*	:	Z	3						エ	ロ	レ	4	キ		
(4)	HLHH	+	;	[4						オ	サ	エ	0	×	4	
(5)	HHLL	,	<	^	5						カ	シ	フ	ワ	キ	8	
(6)	HHLH	-	=	_	6						ユ	又	へ	シ	キ	+	
(7)	HHHL	.	>	~	7						ヨ	セ	ホ	シ	4		
(8)	HHHH	/	?		8						エ	ソ	マ	。	0	■	

CURRICULUM VITAE PENULIS

A. Biodata Pribadi

1. Nama lengkap : M.Shalakhudin Alyazidi
2. Tempat/ tanggal lahir : Gresik, 12 Mei 1990
3. Jenis kelamin : Laki-laki
4. Kewarganegaraan : Indonesia
5. Tinggi, berat badan : 167 cm, 65 kg
6. Agama : Islam
7. Status : Menikah
8. Alamat : Griya Peganden Asri Blok G2 no.20 Manyar
Gresik, Jawa Timur
9. No. Telp : 085733642445
10. Email : shalakhudin.alyazidi@gmail.com

B. Riwayat Pendidikan

1. SD : SDN 1 Sidorukun 1996-2002
2. SMP : SMP 2 Gresik 2002-2005
3. SMK : SMK Semen Gresik 2005-2008
4. Perguruan Tinggi : Universitas Muhammadiyah Gresik
Program Studi Teknik Elektro S1 2011-2018

SURAT PERNYATAAN

Saya yang bertandatangan dibawah ini:

Nama : M.Shalakhudin Alyazidi

No.Reg : 11.632.007

Alamat : Griya Peganden Asri Blok G2 no.20 Manyar

Gresik, Jawa Timur

Menyatakan dengan sesungguhnya bahwa skripsi yang berjudul:

“ Pembuangan Air Limbah Pupuk Berbasis Mikrokontroler ARM STM32F4 “

Benar-benar merupakan hasil karya yang saya buat sendiri berdasarkan penelitian yang telah saya lakukan (bukan plagiat).

Demikian pernyataan ini saya buat, jika ternyata dikemudian hari pernyataan ini tidak benar maka saya siap menanggung semua resiko berdasarkan hukum dan peraturan yang berlaku.

Gresik, 4 Januari 2018

Hormat saya,

M.Shalakhudin Ayazidi