

**SISTEM PENGENDALIAN KECEPATAN ALIRAN UDARA PADA
WIND TUNNEL DENGAN UMPAN BALIK KECEPATAN ALIRAN
UDARA MENGGUNAKAN KONTROL LOGIKA FUZZY**

SKRIPSI

TEKNIK ELEKTRO KONSENTRASI TEKNIK KONTROL

Diajukan untuk Memenuhi Persyaratan
Memperoleh Gelar Sarjana Teknik



Disusun oleh:

AZWAN MAHADIN KUSUMA

NIM. 105060307111028 - 63

UNIVERSITAS BRAWIJAYA

FAKULTAS TEKNIK

MALANG

2016

LEMBAR PENGESAHAN

**SISTEM PENGENDALIAN KECEPATAN ALIRAN UDARA PADA
WIND TUNNEL DENGAN UMPAN BALIK KECEPATAN ALIRAN
UDARA MENGGUNAKAN KONTROL LOGIKA FUZZY**

SKRIPSI

TEKNIK ELEKTRO KONSENTRASI TEKNIK KONTROL

Diajukan Untuk Memenuhi Persyaratan

Memperoleh Gelar Sarjana Teknik



AZWAN MAHADIN KUSUMA
NIM. 105060307111028

Skripsi ini telah direvisi dan disetujui oleh dosen pembimbing
Pada tanggal 18 Agustus 2016

Dosen Pembimbing I

Dr. Ir. Bambang Siswoyo, MT.

NIP. 19621211 198802 1 001

Dosen Pembimbing II

Ir. Purwanto, MT

NIP. 19540424 198601 1 001

Mengetahui,

Ketua Jurusan Teknik Elektro

M. Aziz Muslim, ST., MT., Ph.D

NIP. 19741203 200012 1 001

JUDUL SKRIPSI:

SISTEM PENGENDALIAN KECEPATAN ALIRAN UDARA PADA *WIND TUNNEL*
DENGAN UMPAN BALIK KECEPATAN ALIRAN UDARA MENGGUNAKAN
KONTROL LOGIKA FUZZY

Nama Mahasiswa : AZWAN MAHADIN KUSUMA

NIM : 105060307111028

Program Studi : Teknik Elektro

Konsentrasi : Teknik Kontrol

Komisi Pembimbing :

Ketua : Dr. Ir. Bambang Siswoyo, MT

Anggota : Ir. Purwanto, MT



TIM DOSEN PENGUJI :

Dosen Penguji 1 : M. Aziz Muslim, ST., MT., Ph.D

Dosen Penguji 2 : Goegoes Dwi Nusantoro, ST., MT

Dosen Penguji 3 : Ir. Retnowati, MT



Tanggal Ujian : 18 Agustus 2016

SK Penguji : No.1028 /UN10.6/SK/2016

PERNYATAAN ORISINALITAS SKRIPSI

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya dan berdasarkan hasil penelusuran berbagai karya ilmiah, gagasan dan masalah ilmiah yang diteliti dan diulas di dalam Naskah Skripsi ini adalah asli dari pemikiran saya. Tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu Perguruan Tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata di dalam naskah Skripsi ini dapat dibuktikan terdapat unsur-unsur jiplakan, saya bersedia Skripsi dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, pasal 25 ayat 2 dan pasal 70).

Malang, 18 Agustus 2016

Mahasiswa,

(Materai Rp 6.000,-)

(Tanda Tangan)

Azwan Mahadin Kusuma

NIM. 105060307111028



KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat ALLAH SWT karena berkat Rahmat dan Karunia-NYA penulis dapat menyelesaikan penulisan skripsi ini. Shalawat beserta salam semoga senantiasa terlimpah curahkan kepada Nabi Muhammad SAW, kepada keluarganya, para sahabatnya, hingga kepada umatnya hingga akhir zaman, amin.

Dalam penyusunan skripsi ini tidak lepas dari bantuan, bimbingan serta dukungan dari berbagai pihak. Pada kesempatan ini penulis ingin menyampaikan rasa terima kasih kepada :

- Keluarga tercinta, Bapak Safi'i, Ibu Rustinah dan Kakak Fitria Rachmawati yang telah merawat, menjaga, memberikan kasih sayang, membesarkan, membimbing dan mendoakan hingga sampai pada kesuksesan saat ini.
- Bapak M. Aziz Muslim, S.T., M.T., Ph.D selaku Ketua Jurusan Teknik Elektro Universitas Brawijaya.
- Bapak Hadi Suyono, S.T., M.T., Ph.D selaku Sekretaris Jurusan Teknik Elektro Universitas Brawijaya.
- Bapak Ir. Purwanto, M.T selaku KKDK Teknik Kontrol dan sebagai Dosen Pembimbing II yang telah memberi masukan, ide, saran dan telah meluangkan waktunya untuk bimbingan.
- Bapak Dr. Ir. Bambang Siswoyo, MT sebagai Dosen Pembimbing I yang telah memberikan segala ilmu, bimbingan, nasihat, gagasan, ide, saran, motivasi, waktu dan bantuan yang telah diberikan.
- Bapak Ibu Dosen, karyawan, staff recording dan RBTE atas segala bantuan dan kemudahan.
- Teman-teman seperjuangan Kibro, Rainer, Randy, Dandy, Dugal, Afin yang selalu memberikan dukungan-dukungan yang membangun selama perkuliahan dan pengerjaan skripsi.
- Saudari tercinta Risna Yulianti yang telah mensupport, memberi do'a, dukungan, semangat dan masukan yang banyak kepada penulis.
- Teman-teman "Game", Delief, Gigih, Dikma, Willi, Bayu yang telah menghibur dan memberi support yang besar.

- Keluarga besar “Magnet” yang telah mendukung dan memberi do’a kepada penulis.
- Semua pihak yang telah banyak memberi dukungan dan masukan kepada penulis.

Semoga ALLAH SWT memberi balasan yang berlipat ganda kepada semuanya. Sehingga pada akhirnya skripsi ini dapat bermanfaat bagi kita semua dan yang terpenting turut serta memajukan ilmu pengetahuan, Amin Ya Robbal Alamin.

Malang, Agustus 2016

Penulis

UNIVERSITAS



RINGKASAN

Azwan Mahadin Kusuma, Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya, Agustus 2016, Sistem Pengendalian Kecepatan Aliran Udara Pada *Wind Tunnel* Dengan Umpan Balik Kecepatan Aliran Udara Menggunakan Kontrol Logika Fuzzy, Dosen Pembimbing: Dr. Ir. Bambang Siswoyo, MT., Ir. Purwanto, MT.

Wind tunnel adalah alat yang digunakan dalam penelitian aerodinamika untuk mempelajari karakteristik aliran udara. *Wind tunnel* digunakan untuk mensimulasikan keadaan sebenarnya pada suatu benda yang berada dalam pengaruh gaya-gaya aerodinamis untuk menganalisis kinerja mekanika terbang dari suatu benda terbang. Untuk membangkitkan aliran udara pada *wind tunnel*, yaitu dengan cara mengatur putaran *propeller* atau baling-baling yang di kendalikan melalui motor penggerak (motor induksi 3 fasa) dan menggunakan sensor *air flow* untuk mengetahui kecepatan aliran udaranya. Motor penggerak tidak sepenuhnya sebagai pengontrolan kecepatan aliran udara, namun spesifikasi *propeller* juga sangatlah berpengaruh pada kecepatan aliran udara. Dengan itu untuk keperluan uji spesimen, perlu distabilkan kecepatan aliran udaranya dengan diperlukannya sistem kontrol kecepatan aliran udara berdasarkan *experience of operator* (metode berdasarkan pengalaman operator) dengan melihat respon *output* nya.

Penelitian ini menggunakan Kontrol Logika Fuzzy (KLF) metode Mamdani dengan *output single tone*. Pada penelitian ini diberi *setpoint* 10 m/s dan menghasilkan respon *steady state* 5,6 detik dan ketika sistem diberi gangguan menghasilkan respon *recovery time* sebesar 1600 ms.

Kata kunci: *Wind Tunnel*, Kontrol Logika Fuzzy, Aliran Udara, *Propeller*.

SUMMARY

Azwan Mahadin Kusuma, Departement of Electrical Engineering, Faculty of Engineering University of Brawijaya, August 2016, *Air Flow Speed Control System On Wind Tunnel with Feedback Air Flow Speed Using Fuzzy Logic Control*, Academic Supervisor: Dr. Ir. Bambang Siswoyo, MT., and Ir. Purwanto, MT.

Wind tunnel is a tool used in aerodynamic research to study the characteristics of the airflow. Wind tunnel used to simulate the real situation on an object under the influence of aerodynamic forces to analyze the mechanics of performance flew from a flying object. To generate airflow in the wind tunnel, namely by adjusting the rotation propeller or propellers are controlled via the motor (3 phase induction motors) and using a water flow sensor to determine the speed of air flow. The driving motors are not entirely as controlling air flow velocity, but the specification propeller also very influential on the speed of air flow. With that for the purposes of the test specimens, need to be stabilized air flow velocity of the need for air flow speed control system based on the experience of the operator (the method based on operator experience) to see its output response.

This study uses Fuzzy Logic Control (KLF) Mamdani method with single tone output. In this experiment given setpoint 10 m/s and produces a steady state response and 5.6 seconds when the system is generating a response by interruption recovery time of 1600 ms

Keywords: Wind Tunnel, Fuzzy Logic Control, Air Flow, Propeller.

DAFTAR ISI

KATA PENGANTAR	i
RINGKASAN	iii
SUMMARY	iv
DAFTAR ISI	v
DAFTAR GAMBAR	viii
DAFTAR TABEL	ix
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan Penulisan	3
1.5 Sistematika Pembahasan	3
BAB II TINJAUAN PUSTAKA	5
2.1 Terowongan Angin (<i>Wind Tunnel</i>)	5
2.2 Kontroler	5
2.3 Kontrol Logika Fuzzy	6
2.3.1 Fungsi Keanggotaan Kontrol Logika Fuzzy	6
2.3.2 <i>Fuzzifikasi</i>	7
2.3.3 <i>Aturan Fuzzy (Fuzzy Rule)</i>	8
2.3.4 Metode Inferensi <i>MAX-MIN</i>	8
2.3.5 Metode <i>Defuzzifikasi Weighted Average</i>	9
2.4 Mikrokontroler Arduino Mega2560	10
2.4.1 Penyimpanan Data (<i>Memory</i>)	11
2.4.2 <i>Input dan Output</i>	12
2.4.3 Sumber Daya	13
2.4.4 Komunikasi	13
2.5 VFD (<i>Variable Frequency Drive</i>)	13
2.6 PWM (<i>Pulse Width Modulation</i>)	14
2.7 Sensor <i>Air Flow</i> (Tabung Pitot)	15
2.8 Motor Induksi 3 Fasa	16

BAB III METODOLOGI PENELITIAN	19
3.1 Spesifikasi Alat	19
3.2 Perancangan Pembuatan Sistem	19
3.2.1 Perancangan Perangkat Keras	19
3.2.2 Perancangan Perangkat Lunak.....	20
3.3 Pengujian Perangkat Keseluruhan.....	20
3.4 Pengambilan Kesimpulan.....	20
BAB IV PERANCANGAN DAN PEMBUATAN ALAT	21
4.1 Diagram Blok Sistem	21
4.2 Perancangan Sistem.....	22
4.3 Perancangan Perangkat Keras	22
4.3.1 Baling - baling (<i>Propeller</i>)	22
4.3.2 Arduino Mega2560	22
4.3.3 Motor Induksi 3 Fasa	23
4.3.4 Tabung Pitot (<i>Sensor Air Flow</i>)	24
4.3.5 VFD (<i>Variable Frequency Drive</i>)	24
4.4 Perancangan Kontrol Logika Fuzzy (KLF).....	25
4.4.1 Nilai <i>Input</i> dan <i>Output</i> KLF	25
4.4.2 Fungsi Keanggotaan	25
4.4.3 Fungsi Keanggotaan Keluaran.....	26
4.4.4 Perancangan Aturan <i>Fuzzy</i>	27
4.4.5 Metode Inferensi <i>Max-Min</i>	27
4.4.6 Metode Defuzifikasi	29
4.5 Perancangan Perangkat Lunak.....	30
BAB V PENGUJIAN DAN ANALISIS	33
5.1 Pengujian Sensor <i>Air Flow</i>	33
5.2 Pengujian Respon Aktuator (Motor Induksi 3 Fasa)	36
5.3 Pengujian Keseluruhan Sistem	39
BAB VI PENUTUP	43
6.1 Kesimpulan.....	43
6.2 Saran.....	43
DAFTAR PUSTAKA	45
LAMPIRAN	47

DAFTAR GAMBAR

Gambar 2.1 *Wind Tunnel* 5

Gambar 2.2 Fungsi Keanggotaan Bentuk Triangular 7

Gambar 2.3 Fungsi Keanggotaan Bentuk Trapesium 7

Gambar 2.4 Inferensi *Fuzzy* dengan Metode *MAX-MIN*..... 9

Gambar 2.5 Bentuk Fisik Arduino Mega2560 11

Gambar 2.6 Skema Rangkaian Arduino Mega2560 11

Gambar 2.7 Bentuk Sinyal PWM..... 14

Gambar 2.8 Prinsip Kerja Sensor *Air Flow* 16

Gambar 2.9 Kontruksi Motor Induksi 3 Fasa..... 17

Gambar 4.1 Diagram Blok Sistem 21

Gambar 4.2 Bentuk Fisik Propeller..... 22

Gambar 4.3 Tampak depan Arduino Mega2560..... 23

Gambar 4.4 Bentuk Fisik Motor Induksi 3 Fasa..... 24

Gambar 4.5 Bentuk Fisik Sensor *Air Flow* 24

Gambar 4.6 Bentuk Fisik VFD 25

Gambar 4.7 Fungsi Keanggotaan *Input Error* 26

Gambar 4.8 Fungsi Keanggotaan *Input Change Error* 26

Gambar 4.9 Fungsi Keanggotaan Keluaran PWM..... 27

Gambar 4.10 Fungsi Keanggotaan Keluaran (MP=0,3)..... 28

Gambar 4.11 Fungsi Keanggotaan Keluaran (Z=0,25; P=0,25; MP=0,75)..... 29

Gambar 4.12 *Flowchart* Perangkat Lunak 31

Gambar 5.1 Diagram Blok Pengujian Sensor *Air Flow* 34

Gambar 5.2 Grafik Hubungan Kecepatan Putaran Motor dengan Tegangan Keluaran Sensor *Air Flow* 36

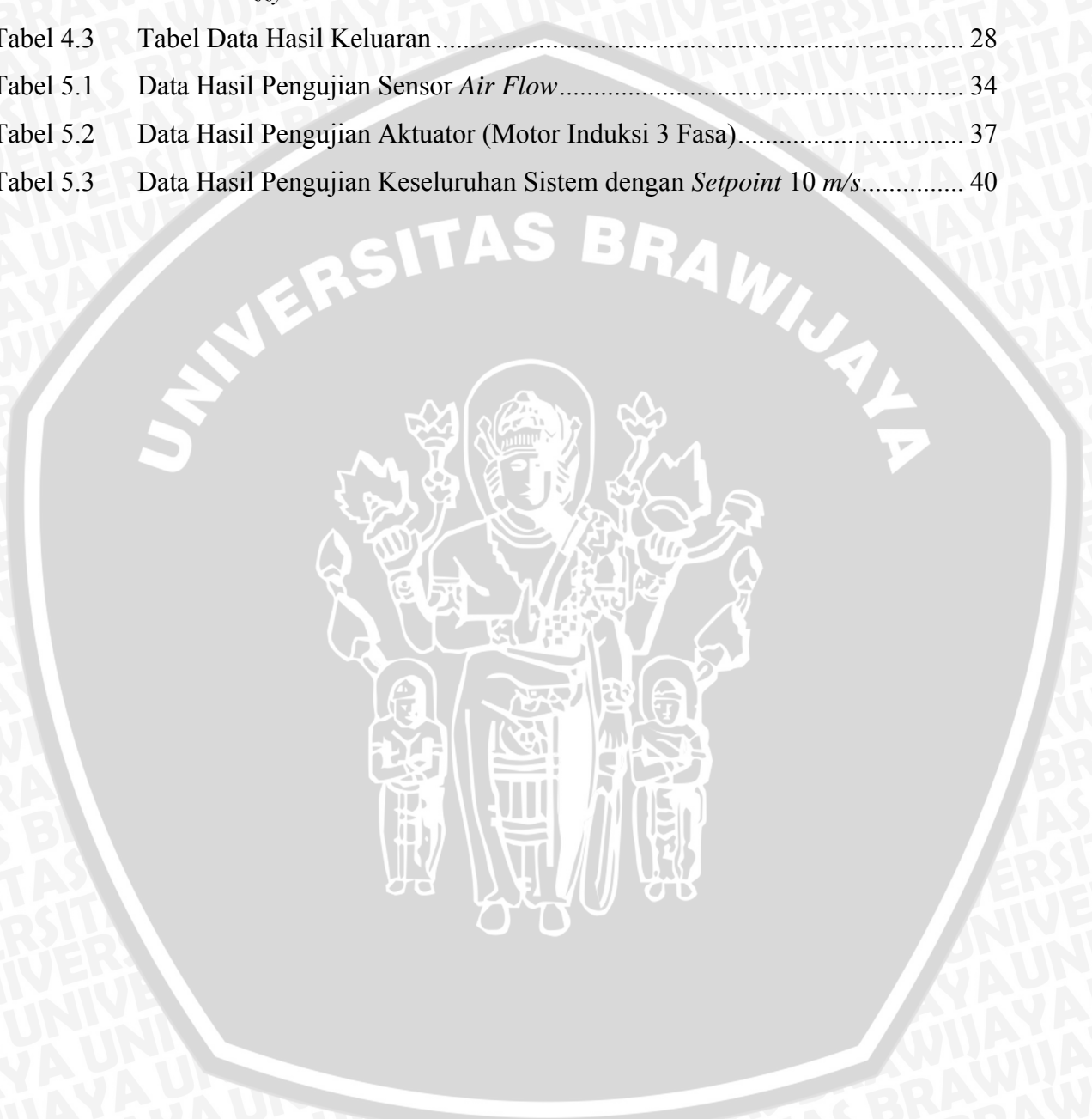
Gambar 5.3 Grafik Hubungan Tegangan Keluaran Motor Terhadap Waktu..... 38

Gambar 5.4 Pengujian Sistem dengan *Setpoint 10 m/s*..... 42



DAFTAR TABEL

Tabel 4.1	Fungsi Pin Arduino Mega2560	23
Tabel 4.2	Aturan <i>Fuzzy Rule</i>	27
Tabel 4.3	Tabel Data Hasil Keluaran	28
Tabel 5.1	Data Hasil Pengujian Sensor <i>Air Flow</i>	34
Tabel 5.2	Data Hasil Pengujian Aktuator (Motor Induksi 3 Fasa).....	37
Tabel 5.3	Data Hasil Pengujian Keseluruhan Sistem dengan <i>Setpoint 10 m/s</i>	40



BAB I PENDAHULUAN

1.1 Latar Belakang Masalah

Adanya minat atau ketertarikan yang cukup besar pada bidang industri terkait dengan penggunaan *Unmanned Aerial Vehicle* (UAV) baru-baru ini. Hal ini dikarenakan pesawat tanpa awak yang bisa digunakan untuk membawa muatan, baik peralatan elektronik maupun muatan lainnya yang mampu dikendalikan secara jarak jauh oleh pilot atau mengendalikan diri sendiri secara otomatis. Pesawat ini memiliki prinsip terbang menggunakan hukum aerodinamika melalui sayap untuk mengangkat dirinya secara otomatis. Seringkali penggunaan UAV dimanfaatkan untuk penelusuran tempat-tempat yang susah/sulit dijangkau dengan manusia, seperti pegunungan, lembah dan lokasi penambangan.

Adanya pengelompokan pesawat udara aerodinamis yaitu tipe menggunakan motor dan tidak menggunakan motor. Untuk tipe menggunakan motor atau disebut sayap putar (*rotary wing*) dan tidak menggunakan motor sering disebut bersayap tetap (*fixed wing*). Untuk penulisan laporan ini dipilih pengukuran gaya aerodinamis pada tipe yang tidak menggunakan motor (*fixed wing*), hal ini dikarenakan pada tipe ini pesawat memiliki sayap yang lebar dan lebih stabil saat penerbangan. Pesawat sayap tetap (*fixed wing*) yang terbang bukan dikarenakan gerakan pada sayap, melainkan pesawat sayap tetap ini terbang melaju melalui udara, pergerakan udara pada sayap menghasilkan gaya angkat yang mampu mengangkat pesawat. Pesawat bersayap tetap memanfaatkan prinsip *bernaulli*, yaitu fluida yang bergerak lebih cepat memiliki tekanan yang lebih rendah dibandingkan dengan fluida yang bergerak lebih lambat. Hal ini dikarenakan sayap bagian atas lebih panjang dibanding sayap bagian bawah, karena adanya model kelengkungan model permukaan sayap bagian atas, maka dari itu sayap ini akan mengalirkan aliran udara lebih cepat dibagian atas dibandingkan dengan bagian bawah. Adanya perbedaan kecepatan udara yang menyebabkan pesawat dapat terbang, pesawat harus bergerak pada kecepatan tertentu agar ada udara yang mengalir di sayap.

Dalam mendesain UAV, dibutuhkan rangka pesawat (*frame*) yang terdiri dari badan pesawat (*fuselage*), sayap (*wing*), *stabilizer horizontal* dan *vertikal*. Kerangka ini memiliki karakteristik berupa gaya angkat yang perlu adanya alat bantu uji berupa *wind tunnel*.

Wind tunnel atau terowongan angin digunakan untuk mensimulasikan keadaan sebenarnya pada suatu benda yang berada dalam pengaruh gaya-gaya *aerodinamis*. *Wind tunnel* dikembangkan dalam menganalisis efek angin yang bergerak atau disekitar objek padat. Adanya beberapa bagian penting pada *Wind Tunnel* sebagai alat uji aerodinamika, yaitu *fan*, *diffuser*, *test section*, *contraction*, dan *honeycomb*. *Fan* atau *propeller* adalah bagian utama penghasil udara. Untuk dapat menggerakkan *fan* maka dibutuhkan suatu penggerak *fan*. *Fan* dihubungkan dengan poros motor penggerak supaya *fan* dapat berputar dan menghasilkan udara. *Diffuser* berfungsi sebagai komponen yang menyebabkan kenaikan tekanan pada udara. Ketika udara melalui *diffuser* maka tekanannya akan naik, tetapi kecepatannya menurun, sesuai dengan Prinsip *Bernoulli* menyatakan bahwa pada suatu aliran fluida, peningkatan kecepatan pada fluida akan menimbulkan penurunan tekanan pada aliran tersebut, atau sebaliknya. *Test section* adalah bagian dari terowongan angin untuk meletakkan model yang akan diuji. Dimensi *test section* disesuaikan dengan dimensi model yang akan diuji, kemudian dimensi tersebut ditentukan dengan dimensi model dan diberi toleransi ukuran. *Contraction* adalah ruang yang memiliki luas perlahan-lahan mengecil. *Contraction* memiliki fungsi yang berlawanan dari *diffuser*. *Contraction* merupakan komponen yang dapat menyebabkan kenaikan kecepatan udara dan penurunan tekanan udara. Dan *Honeycomb* berbentuk seperti sarang lebah dan bersaluran lurus. *Honeycomb* berfungsi untuk mengurangi turbulensi aliran. Ketika aliran udara *turbulen* masuk dan melalui *honeycomb*, maka aliran tersebut akan menjadi aliran *laminar*.

Pada kecepatan aliran udara, spesifikasi *propeller* sangat berpengaruh, dimana beberapa spesifikasi *propeller* diantaranya yaitu bobot *propeller*, diameter *propeller*, *pitch propeller*, jumlah *propeller*, dan luas *propeller*. Maka dari itu untuk keperluan uji spesimen model pesawat, perlu distabilkan kecepatan aliran udaranya. Untuk itu diperlukan sistem kontrol kecepatan aliran udara dengan berdasarkan *experience of operator* (metode berdasarkan pengalaman operator) dengan melihat respon outputnya, maka dipilih metode Kontrol Logika Fuzzy (KLF).

1.2 Rumusan Masalah

Rumusan masalah yang didapat dari latar belakang adalah sebagai berikut :

1. Bagaimana merancang sistem kontrol aliran udara pada *wind tunnel* saat terjadi gangguan menggunakan Kontrol Logika Fuzzy?
2. Bagaimana merancang *hardware* dan *software* kontrol aliran udara pada *wind tunnel* menggunakan Kontrol Logika Fuzzy?

1.3 Batasan Masalah

Mengacu pada permasalahan laporan ini, maka akan dibatasi sebagai berikut:

1. Model alat uji kontrol aliran udara yang dibuat adalah miniatur terowongan angin.
2. Pembahasan sistem pengontrolan menggunakan aplikasi KLF.
3. Rangkaian elektrik dan kinerja *driver* tidak dibahas mendalam.
4. Gangguan diberikan melalui perubahan aliran udara pada terowongan angin.

1.4 Tujuan Penulisan

Merancang sistem kontrol aliran udara untuk mengatur aliran udara pada *wind tunnel* menggunakan Kontrol Logika Fuzzy.

1.5 Sistematika Pembahasan

Adapun sistematika laporan penelitian ini adalah sebagai berikut :

1. BAB I Pendahuluan

Dalam bab ini membahas latar belakang masalah, rumusan masalah, batasan masalah, tujuan penulisan dan sistematika pembahasan.

2. BAB II Tinjauan Pustaka

Dalam bab ini membahas beberapa teori-teori untuk mendukung dalam perancangan dan pembuatan alat.

3. BAB III Metode Penelitian

Membahas spesifikasi alat, perancangan pembuatan sistem, pengujian perangkat keseluruhan, dan pengambilan kesimpulan.

4. BAB IV Perancangan dan Pembuatan Alat

Membahas perancangan alat yang meliputi spesifikasi, perencanaan blok diagram, prinsip kerja, dan pembuatan alat.

5. BAB V Pengujian dan Analisis

Berisikan hasil pengujian sistem yang telah dibuat dan analisis hasil yang diperoleh.

6. BAB VI Penutup

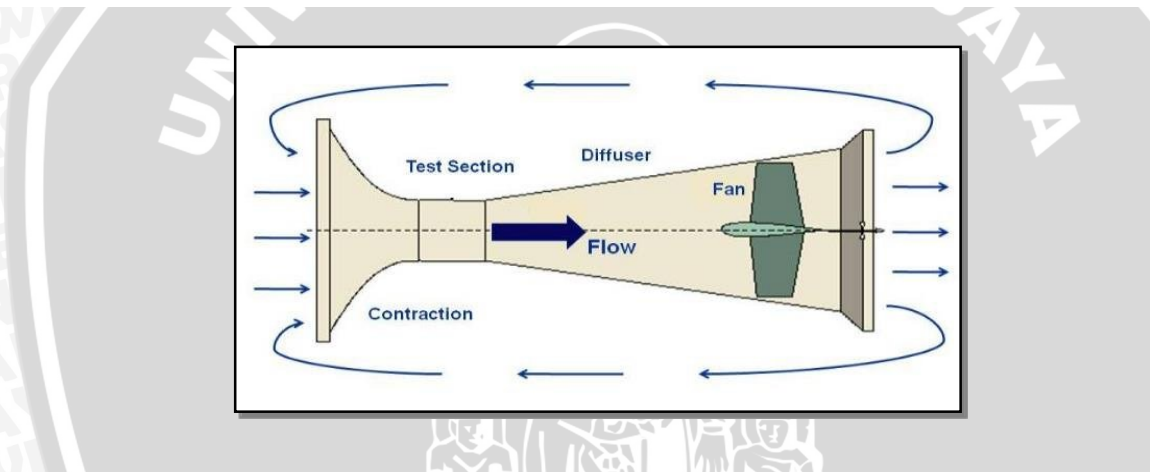
Membahas kesimpulan penelitian dan saran-saran yang diperlukan untuk pengembangan penelitian selanjutnya.

BAB II

TINJAUAN PUSTAKA

2.1 Terowongan Angin (*Wind Tunnel*)

Wind tunnel adalah alat yang digunakan dalam penelitian aerodinamika untuk mempelajari karakteristik aliran udara. *Wind tunnel* digunakan untuk mensimulasikan keadaan sebenarnya pada suatu benda yang berada dalam pengaruh gaya-gaya aerodinamis untuk menganalisis kinerja mekanika terbang dari suatu benda terbang. *Wind tunnel* juga banyak digunakan dalam pengujian berbagai kondisi benda dalam aliran udara seperti konstruksi gedung pencakar langit, lingkungan perkotaan dan lain-lain. Bentuk *wind tunnel* dapat dilihat pada Gambar 2.1.



Gambar 2.1 *Wind Tunnel*
Sumber : www.grc.nasa.gov

2.2 Kontroler

Kontroler disebut juga kompensator, pengendali atau *filter*. Kontroler adalah suatu sistem dinamis yang ditambahkan untuk mendapatkan karakteristik sistem yang diinginkan. Fungsi kontroler yaitu sebagai pembanding nilai masukan dan keluaran sistem secara keseluruhan, menentukan *error*, menghasilkan sinyal kontrol untuk mengurangi *error* menjadi nilai nol/ nilai yang sangat kecil.

Prinsip kerja kontroler adalah membandingkan nilai aktual keluaran *plant* dengan nilai referensi, kemudian menentukan nilai kesalahan dan akhirnya menghasilkan sinyal kontrol untuk meminimalkan kesalahan. (Ogata, K. 1997).

2.3 Kontrol Logika Fuzzy

Kontrol logika *fuzzy* adalah sistem berbasis aturan (*rule based system*) yang didalamnya terdapat himpunan aturan *fuzzy* yang mempresentasikan mekanisme pengambilan keputusan. Aturan yang dibuat digunakan untuk memetakan variabel *input* ke variabel *output* dengan pernyataan *If – Then*.

Kontrol ini akan menggunakan data tertentu (*crisp*) dari sejumlah sensor kemudian mengubahnya menjadi bentuk linguistik atau fungsi keanggotaan melalui proses *fuzzifikasi*. Lalu dengan aturan *fuzzy*, *inference engine* yang akan menentukan hasil keluaran *fuzzy*. Setelah itu hasil ini akan diubah kembali menjadi bentuk numerik melalui proses *defuzzifikasi*. Struktur dasar kontrol logika *fuzzy* ditunjukkan sebagai berikut :

2.3.1. Fungsi Keanggotaan Kontrol Logika Fuzzy

Fungsi keanggotaan menotasikan nilai kebenaran anggota-anggota himpunan *fuzzy*. Interval nilai yang digunakan untuk menentukan fungsi keanggotaan, yaitu nol dan satu. Tiap fungsi keanggotaan memetakan elemen himpunan *crisp* ke semesta himpunan *fuzzy*.

Suatu himpunan *fuzzy* A dalam semesta pembicaraan U dinyatakan dengan fungsi keanggotaan μ_A yang harganya berada dalam interval [0,1]. Secara matematika hal ini dinyatakan dengan :

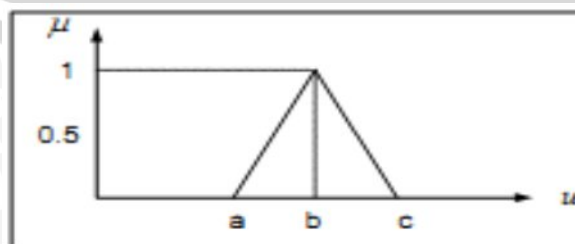
$$\mu_A : U \rightarrow [0,1] \quad (2.1)$$

a) Fungsi Keanggotaan Bentuk Triangular

Definisi fungsi triangular sebagai berikut :

$$\mu_A(x) = \begin{cases} 0 & x < a \\ \frac{x-a}{b-a} & a \leq x \leq b \\ \frac{c-x}{c-b} & b \leq x \leq c \\ 0 & x > c \end{cases} \quad (2.2)$$

Berikut dapat dilihat grafik fungsi keanggotaan bentuk triangular ditunjukkan pada Gambar 2.2.



Gambar 2.2 Fungsi keanggotaan bentuk triangular

Sumber : Yan, Ryan dan Power, 1994

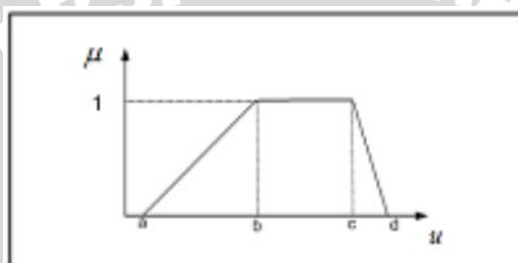
Fungsi keanggotaan bentuk triangular ini digunakan bila diinginkan himpunan *fuzzy* mempunyai nilai proporsional terhadap 0 maupun 1.

b) Fungsi Keanggotaan Bentuk Trapesium

Definisi fungsi trapesium sebagai berikut :

$$\mu_A(x) = \begin{cases} 0 & x < a \\ \frac{x-a}{b-a} & a \leq x \leq b \\ 1 & b \leq x \leq c \\ \frac{c-x}{d-c} & c \leq x \leq d \\ 0 & x > d \end{cases} \quad (2.3)$$

Berikut dapat dilihat grafik fungsi keanggotaan bentuk trapesium ditunjukkan pada Gambar 2.3.



Gambar 2.3 Fungsi keanggotaan bentuk trapesium

Sumber : Yan, Ryan dan Power, 1994

2.3.2. Fuzzifikasi

Proses *fuzzifikasi* merupakan proses untuk mengubah variabel non *fuzzy* (variabel numerik) menjadi variabel *fuzzy* (variabel linguistik). Nilai masukan-masukan yang masih dalam bentuk variabel numerik yang telah dikuantisasi sebelum diolah oleh pengendali logika *fuzzy* harus diubah terlebih dahulu ke dalam variabel *fuzzy*. Melalui fungsi keanggotaan yang telah disusun, maka dari nilai-nilai masukan tersebut menjadi informasi *fuzzy* yang berguna nantinya untuk proses pengolahan secara *fuzzy* pula. Proses ini disebut *fuzzifikasi*. (Yan, Ryan dan Power, 1994). Proses *fuzzifikasi* diekspresikan sebagai berikut:

$$\mu_A(x) = \text{operator fuzzifikasi yang memetakan himpunan } x \text{ ke himpunan } A \quad (2.4)$$

Dengan:

- x = nilai *crisp* variabel masukan
- A = himpunan *fuzzy* variabel yang terdefinisi
- $\mu_A(x)$ = operator fuzzifikasi yang memetakan himpunan *crisp* ke himpunan *fuzzy*



2.3.3 Aturan Fuzzy (Fuzzy Rule)

Fuzzy rule adalah bagian yang menggambarkan dinamika suatu sistem terhadap masukan yang dikarakteristikan oleh sekumpulan variabel-variabel linguistik dan berbasis pengetahuan seorang operator. Pernyataan tersebut umumnya dinyatakan oleh suatu pernyataan bersyarat.

Dalam pengendali berbasis *fuzzy*, aturan pengendalian *fuzzy* berbentuk aturan “ IF – THEN”. Untuk sebuah sistem *Multi Input Single Output* (MISO) basis aturan pengendalian *fuzzy* berbentuk seperti berikut :

$$\text{Rule 1 } x_1 \text{ is } A_1 \text{ and } x_2 \text{ is } B_1 \text{ then } z \text{ is } C_1$$

$$\text{Rule 2 } x_1 \text{ is } A_2 \text{ and } x_2 \text{ is } B_2 \text{ then } z \text{ is } C_2$$

$$\text{Rule n } x_1 \text{ is } A_n \text{ and } x_2 \text{ is } B_n \text{ then } z \text{ is } C_n$$

Dengan X, Y, Z merupakan variabel linguistik, dimana X dan Y merupakan variabel masukan, dan Z merupakan variabel keluaran sistem. $a_i, b_i,$ dan c_i merupakan nilai linguistik dari $X, Y,$ dan Z .

2.3.4. Metode Inferensi MAX-MIN

Metode inferensi merupakan proses untuk mendapatkan keluaran dari suatu kondisi masukan dengan mengikuti aturan-aturan yang telah ditetapkan. Keputusan yang didapatkan pada proses ini masih dalam bentuk *fuzzy* yaitu derajat keanggotaan keluaran. Pada metode *Max-Min* aturan operasi minimum Mamdani digunakan untuk implikasi *fuzzy*. Persamaan aturan minimum adalah :

$$\mu_{A \cup B}(x) = \min(\mu_A(x), \mu_B(x)) \quad (2.5)$$

Dengan :

$$\mu_{A \cup B}(x) = \min(\mu_A(x), \mu_B(x))$$

Sebagai contoh, terdapat dua basis kaidah aturan *fuzzy*, yaitu:

$$\text{Rule 1 : } x_1 \text{ is } A_1 \text{ and } x_2 \text{ is } B_1 \text{ then } z \text{ is } C_1$$

$$\text{Rule 2 : } x_1 \text{ is } A_2 \text{ and } x_2 \text{ is } B_2 \text{ then } z \text{ is } C_2$$

Pada metode penalaran *MAX-MIN* fungsi keanggotaan konsekuen dinyatakan dengan :

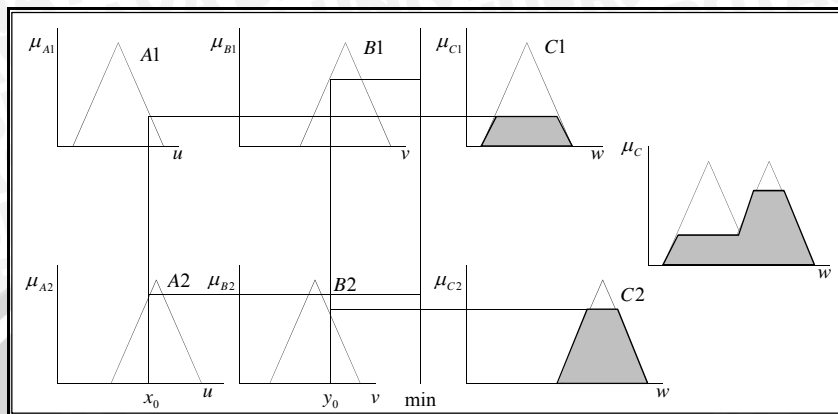
$$\mu_{C_1}(z) = \min(\mu_{A_1}(x_1), \mu_{B_1}(x_2)) \quad \mu_{C_2}(z) = \min(\mu_{A_2}(x_1), \mu_{B_2}(x_2)) \quad (2.6)$$

Dimana :





Agar lebih jelas, metode dapat dilihat pada Gambar 2.4.



Gambar 2.4 Inferensi fuzzy dengan metode MAX-MIN

Sumber : Yan, Ryan dan Power, 1994

2.3.5 Metode Defuzzifikasi Weighted Average

Defuzzifikasi adalah proses untuk mendapatkan nilai numerik dari data fuzzy yang dihasilkan dari proses inferensi (Yan, Ryan dan Power, 1994). Proses defuzzifikasi dinyatakan sebagai berikut :

$$\mu_{A_i}(x) \cdot \mu_{B_j}(y) \cdot \mu_{C_k}(z) \quad (2.7)$$

Dengan :

$\mu_{A_i}(x)$ = aksi kontrol fuzzy

$\mu_{B_j}(y)$ = aksi kontrol crisp

$\mu_{C_k}(z)$ = operator defuzzifikasi

Metode Weighted Average didefinisikan sebagai berikut :

$$z = \frac{\sum_{i=1}^n \mu_{A_i}(x) \cdot \mu_{B_j}(y) \cdot \mu_{C_k}(z)}{\sum_{i=1}^n \mu_{A_i}(x) \cdot \mu_{B_j}(y)} \quad (2.8)$$

Dengan :

z = Keluaran

$\mu_{A_i}(x)$ = Bobot nilai benar $\mu_{B_j}(y)$

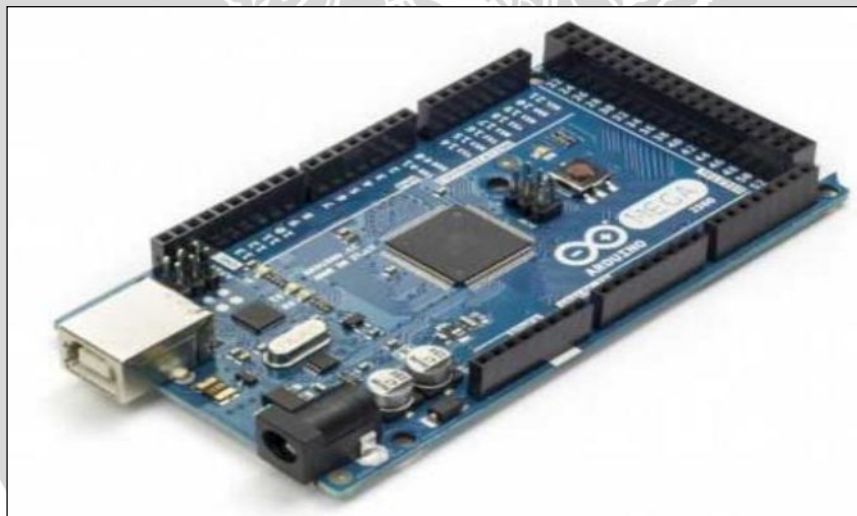
$\mu_{C_k}(z)$ = Nilai linguistik pada fungsi keanggotaan keluaran

n = Banyak derajat keanggotaan



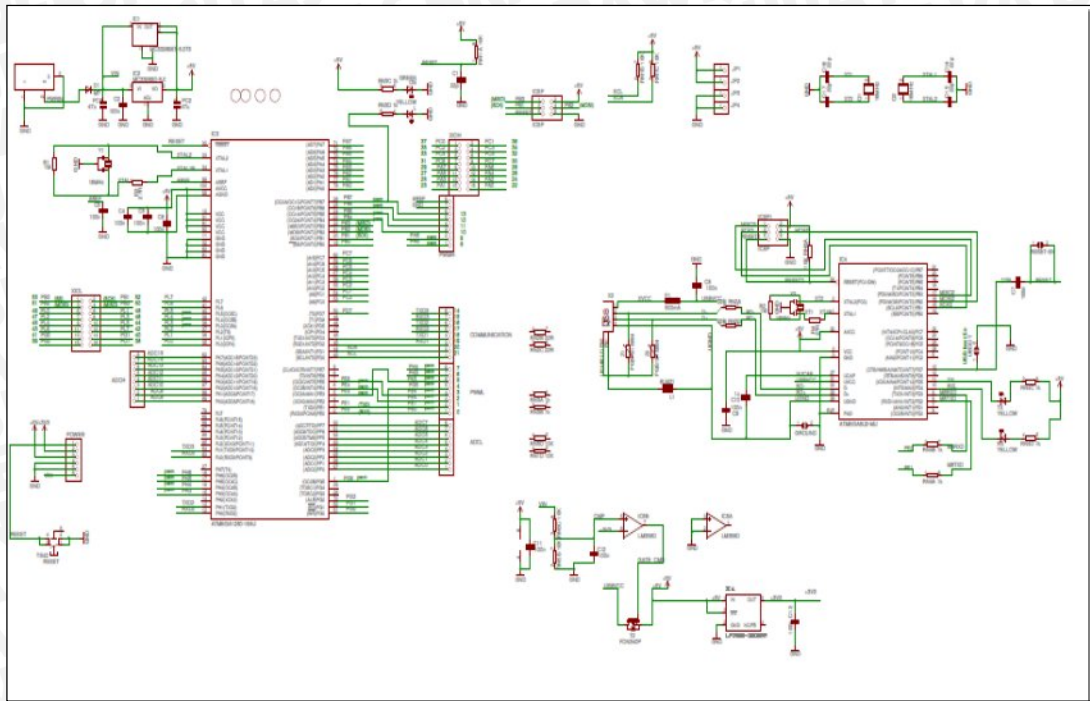
2.4 Mikrokontroler Arduino Mega2560

Mikrokontroler ini berfungsi sebagai pengaturan kerja alat agar dapat bekerja secara sistematis, dengan program yang telah di *input* kedalam mikrokontroler sehingga *output* keluaran akan bekerja sesuai dengan program *assambler* tersebut. Arduino Mega2560 adalah papan mikrokontroler berbasis ATmega2560 (*datasheet ATmega2560*). Arduino Mega2560 memiliki 54 pin digital *input/output*, dimana 15 pin dapat digunakan sebagai *output* PWM, 16 pin sebagai *input analog*, dan 4 pin sebagai UART (port serial hardware), 16 MHz kristal osilator, koneksi USB, jack power, header ICSP, dan tombol *reset*. Cukup dengan menghubungkannya ke komputer melalui kabel USB atau *power* dihubungkan dengan adaptor AC-DC atau baterai untuk mulai mengaktifkannya. Arduino Mega2560 kompatibel dengan sebagian besar *shield* yang dirancang untuk *Arduino Duemilanove* atau *Arduino Diecimila*. Arduino Mega2560 adalah versi terbaru yang menggantikan versi Arduino Mega. Bentuk fisik dan skema rangkaian Arduino Mega2560 ditunjukkan pada Gambar 2.5 dan Gambar 2.6.



Gambar 2.5 Bentuk fisik Arduino Mega2560

Sumber : <http://arduino.cc>



Gambar 2.6 Skema Rangkaian Arduino Mega2560

Sumber : <http://arduino.cc>

2.4.1 Penyimpan Data (*Memory*)

Arduino Mega2560 memiliki 256 KB *flash memory* untuk menyimpan kode (yang 8 KB digunakan untuk *bootloader*), 8 KB SRAM dan 4 KB EEPROM (yang dapat dibaca dan ditulis dengan perpustakaan EEPROM).

2.4.2 *Input dan Output*

Masing-masing dari 54 digital pin pada Arduino Mega2560 dapat digunakan sebagai *input* atau *output* dengan menggunakan fungsi *pinMode()*, *digitalWrite()*, dan *digitalRead()*. Arduino Mega beroperasi pada tegangan 5 volt. Setiap pin dapat memberi atau menerima arus maksimum 40 mA dan memiliki resistor *pull-up internal* (yang terputus secara *default*) sebesar 20-50 kΩ. Arduino Mega2560 juga memiliki pin pada fungsi khusus, yaitu:

- Serial yaitu 0 (RX) dan 1 (TX); Serial 1 : 19 (RX) dan 18 (TX); Serial 2 : 17 (RX) dan 16 (TX); Serial 3 : 15 (RX) dan 14 (TX). Digunakan untuk menerima (RX) dan mengirimkan (TX) data serial TTL.
- *Eksternal Interupsi* yaitu Pin 2 (interrupt 0), pin 3 (interrupt 1), pin 18 (interrupt 5), pin 19 (interrupt 4), pin 20 (interrupt 3), dan pin 21 (interrupt 2). Pin ini dikonfigurasi untuk acuan sebuah interupsi pada nilai rendah, meningkat atau menurun, atau berubah nilai.

- SPI yaitu Pin 50 (MISO), pin 51 (MOSI), pin 52 (SCK), pin 53 (SS). Pin ini sebagai komunikasi SPI.
- LED yaitu Pin 13. Tersedia secara *built-in* pada papan Arduino ATmega2560. LED terhubung ke pin digital 13. Jika pin diberi bernilai *HIGH*, maka *LED* menyala (*ON*), dan ketika pin diberi bernilai *LOW*, maka LED padam (*OFF*).
- TWI yaitu Pin 20 (SDA) dan pin 21 (SCL). Yang mendukung komunikasi TWI menggunakan perpustakaan *Wire*.

Arduino Mega2560 juga mempunyai 16 pin sebagai analog *input* dengan masing-masing pin menyediakan resolusi 10 bit (yaitu 1024 nilai yang berbeda). Secara *default*, pin ini dapat diukur/diatur dari mulai *Ground* sampai dengan 5 *Volt*, juga memungkinkan untuk mengubah titik jangkauan tertinggi atau terendah mereka menggunakan pin AREF dan fungsi *analogReference()*.

Beberapa pin lainnya yang tersedia, antara lain:

- AREF yaitu *input* analog sebagai referensi tegangan. Digunakan dengan fungsi *analogReference()*.
- RESET yaitu Jalur *LOW*, digunakan untuk *me-reset* (menghidupkan ulang) mikrokontroler.

2.4.3 Sumber Daya

Penggunaan adaptor dapat dihubungkan dengan menancapkan *plug jack* dengan pusat-positif sebesar 2,1 *mm* pada *power jack* pada *board*. Pengaktifan Arduino Mega2560 dapat melalui koneksi USB atau Dengan catu daya eksternal (otomatis). Untuk daya eksternal (*non-USB*) dapat berasal dari adaptor AC-to-DC atau baterai. Untuk baterai, ujung kepala baterai dapat dimasukkan pada pin *Gnd* dan *Vin* pada *header* di bagian konektor *POWER*. *Board* Arduino Mega2560 ini dapat beroperasi dengan catu daya eksternal antara 6 *volt* sampai dengan 20 *volt*. Pin 5 *volt* dapat beroperasi tidak stabil jika diberi daya kurang dari 7 *volt*. Jika menggunakan daya lebih dari 12 *volt*, maka regulator tegangan akan panas dan dapat merusak *board* Arduino Mega2560. *Range* daya yang direkomendasikan antara 7 sampai dengan 20 *volt*.

Adapun pin *POWER* pada *board* Arduino Mega2560 adalah:

- *VIN* yaitu Jika menggunakan catu daya eksternal (berbeda dengan 5 *volt* yang berasal dari konektor USB atau sumber tegangan yang telah disesuaikan).
- *GND* yaitu pin *ground*.

2.4.4 Komunikasi

Fasilitas yang istimewa pada Arduino Mega2560 ini yaitu dapat berkomunikasi dengan komputer dan Arduino yang lain. Tersedia empat perangkat UART untuk komunikasi serial TTL (5 volt). ATmega16U2 pada board merupakan saluran untuk USB dan menyediakan portcom virtual untuk software pada komputer. Untuk pengguna Windows diperlukan sebuah file dengan tipe .inf; tetapi tidak pada OSX atau Linux, kedua Operating System ini dapat mengidentifikasi atau mengenali secara otomatis board Arduino sebagai port COM.

2.5 VFD (Variable Frequency Drive)

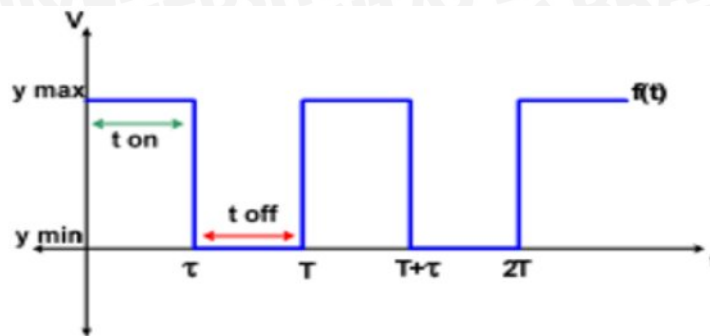
VFD adalah suatu alat yang digunakan untuk mengendalikan kecepatan motor listrik AC (*Alternating Current*) dengan mengontrol frekuensi daya listrik yang dipasang ke motor. VFD semakin populer karena kemampuannya untuk mengontrol kecepatan motor induksi. VFD mengontrol kecepatan motor induksi dengan mengubah frekuensi untuk nilai disesuaikan pada sisi mesin sehingga memungkinkan motor listrik dengan cepat dan mudah menyesuaikan kecepatan dengan nilai yang diinginkan. Dua fungsi utama dari VFD adalah untuk melakukan konversi listrik dari satu frekuensi ke yang lain dan untuk mengontrol frekuensi keluaran. Aplikasi VFD digunakan dari mulai peralatan kecil sampai peralatan besar, yaitu pengaturan pabrik tambang, kompresor dan sistem ventilasi untuk bangunan besar. Selain itu, VFD juga digunakan pada pompa, konveyor dan alat pengendali mesin. Penggunaan VFD pada motor dapat menghemat energi sehingga mengurangi biaya listrik.

2.6 PWM (Pulse Width Modulation)

Pengaturan lebar pulsa modulasi atau PWM merupakan salah satu teknik yang “ampuh” yang digunakan dalam sistem kendali (*control system*) saat ini. Pengaturan lebar modulasi dipergunakan di berbagai bidang yang sangat luas, salah satu diantaranya adalah kendali kecepatan (*speed control*), kendali sistem tenaga (*power control*), pengukuran atau instrumentasi dan telekomunikasi (*measurement and communication*). Modulasi lebar pulsa (PWM) dicapai/diperoleh dengan bantuan sebuah gelombang kotak yang mana siklus kerja (*duty cycle*). (Prabowo, Brilliant Adhi, 2009)

Gelombang dapat diubah-ubah untuk mendapatkan sebuah tegangan keluaran yang bervariasi yang merupakan nilai rata-rata dari gelombang tersebut. PWM pada dasarnya adalah menyalakan (*ON*) dan mematikan (*OFF*) motor dengan cepat. Kuncinya adalah

mengatur berapa lama waktu *ON* dan *OFF*. Berikut merupakan gambar sinyal PWM dapat dilihat pada Gambar 2.7.



Gambar 2.7 Bentuk Sinyal PWM

Sumber : Prabowo, Brilliant Adhi, 2009

Ton adalah waktu di mana tegangan keluaran berada pada posisi tinggi (baca: *high* atau 1) dan, Toff adalah waktu di mana tegangan keluaran berada pada posisi rendah (baca: *low* atau 0). Anggap Ttotal adalah waktu satu siklus atau penjumlahan antara Ton dengan Toff, biasa dikenal dengan istilah “periode satu gelombang”.

$$V_{out} = V_{in} \cdot \frac{t_{on}}{T} \quad (2.9)$$

Tegangan keluaran dapat bervariasi dengan *duty-cycle* dan dapat dirumuskan sebagai berikut :

$$V_{out} = V_{in} \cdot \frac{t_{on}}{T} \quad (2.10)$$

Dengan:

V_{out} = Tegangan keluaran

V_{in} = Tegangan masukan

t_{on} = Periode logika tinggi

T = Periode keseluruhan

T_{ot} = *Duty Cycle*

2.7 Sensor Air Flow (Tabung Pitot)

Tabung pitot merupakan alat ukur yang digunakan untuk mengukur suatu tekanan fluida. Biasanya tabung pitot ini digunakan pada pesawat untuk mengukur kecepatan dari suatu pesawat. Tabung pitot diciptakan oleh insinyur Prancis yang bernama Henri de Pitot (1695-1771) di awal abad 18 dan telah dimodifikasi untuk bentuk modern pada pertengahan abad ke-19 oleh ilmuwan Prancis yaitu Henry Darcy. Dan sampai sekarang

tabung pitot banyak digunakan untuk menentukan kecepatan udara dari sebuah pesawat dan untuk mengukur kecepatan udara dan gas dalam aplikasi industri.

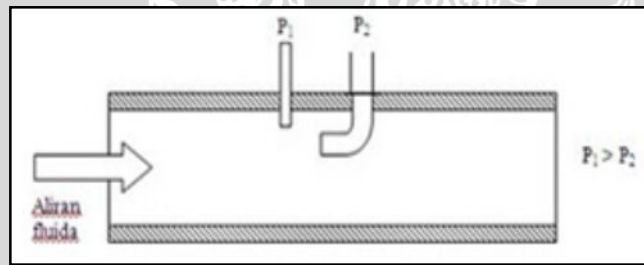
Sensor *air flow* adalah pipa terbuka kecil di mana permukaannya bersentuhan langsung dengan aliran. Pada tabung pitot ini terdiri dari 2 pipa, yaitu :

a. *Static tube* (untuk mengukur tekanan statis)

Pipa ini membuka secara tegak lurus sampai ke aliran sehingga dapat diketahui tekanan statisnya. Tekanan statis (fluida diam) ditinjau ketika fluida yang sedang diam atau berada dalam keadaan setimbang. Fluida statis erat kaitannya dengan hidraustatika dan tekanan. Hidraustatika merupakan ilmu yang mempelajari tentang gaya maupun tekanan di dalam zat cair yang diam.

b. *Dinamic tube* (untuk mengukur tekanan dinamis)

Tekanan dinamis atau fluida ideal merupakan fluida yang mengalir (bergerak). Fluida ideal dikatakan sebagai aliran laminar, artinya kecepatan aliran fluida pada sembarang titik tidak berubah terhadap waktu, baik besar maupun arahnya. Dalam hal ini, aliran fluida ideal memiliki nilai bilangan Reynolds kurang dari 2300. Cara kerja pitot tube dapat dilihat pada Gambar 2.8.



Gambar 2.8 Prinsip Kerja Sensor *Air Flow*

Pipa yang mengukur tekanan statis (P1) terletak secara radial pada batang yang dihubungkan ke skala pengukuran. Tekanan pada ujung pipa di mana fluida masuk merupakan tekanan stagnasi dinamis (P2). Kedua pengukuran tekanan tersebut dimasukkan dalam persamaan *Bernoulli* untuk mengetahui kecepatannya, untuk persamaan perhitungan kecepatannya menggunakan persamaan *bernoulli* adalah sebagai berikut :

$$\rho \left(\frac{v_1^2}{2} + \frac{p_1}{\rho} \right) = \rho \left(\frac{v_2^2}{2} + \frac{p_2}{\rho} \right) \quad (2.11)$$

Di mana :

V = Kecepatan aliran udara (m/s)

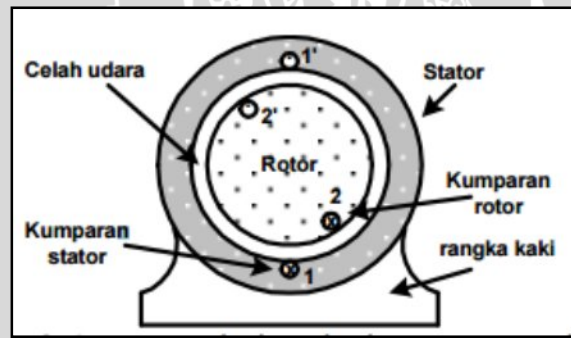
P_1 = Tekanan tabung aliran udara statis (psi)

P_2 = Tekanan tabung aliran udara dinamis (psi)

ρ = Masa jenis udara ($1,2 \text{ kg/m}^3$)

2.8 Motor Induksi 3 Fasa

Motor induksi 3 fasa merupakan motor induksi yang banyak digunakan untuk berbagai keperluan terutama di pabrik-pabrik yang menggunakan motor sebagai penggerakannya. Penamaannya berasal dari kenyataan bahwa arus rotor motor ini bukan diperoleh dari sumber tertentu, tapi merupakan arus yang terinduksi akibat adanya perbedaan antara kecepatan rotasi rotor dan kecepatan rotasi medan putar. Diantara semua jenis motor induksi, motor induksi 3 fasa merupakan motor yang paling stabil dalam kondisi normal bila disuplai dengan sistem 3 fasa yang seimbang. Gambaran sederhana konstruksi motor induksi 3 fasa ini diperlihatkan pada Gambar 2.9.



Gambar 2.9 Kontruksi Motor Induksi 3 Fasa

BAB III

METODE PENELITIAN

Untuk mendapatkan hasil kinerja sistem kontrol yang diharapkan dengan menggunakan Kontrol Logika Fuzzy (KLF), maka jenis penelitian ini bersifat kuantitatif, dikarenakan sistem yang dibuat dalam bentuk perancangan suatu pengendalian sistem kontrol yang dapat dikembangkan.

Adapun tahapan yang perlu dilakukan untuk pembuatan miniatur alat sebagai berikut:

1. Spesifikasi alat
2. Perancangan Pembuatan Sistem
3. Pengujian Perangkat Keseluruhan
4. Pengambilan kesimpulan

3.1 Spesifikasi Alat

Dalam perancangan sistem, hal pertama yang perlu dilakukan adalah menentukan spesifikasi alat. Spesifikasi alat yang harus diperhatikan sebagai berikut:

1. Desain *propeller* berdiameter 147 cm, dengan bentuk alat persegi berukuran 150x150 cm.
2. Motor induksi 3 fasa sebagai penggerak *propeller*.
3. Menggunakan sensor *air flow* sebagai pembaca laju aliran udara.
4. VFD sebagai *inverter*.
5. Mikrokontroler Arduino Mega2560 sebagai pusat proses pengaturan kecepatan motor induksi 3 fasa.
6. Kontroler yang digunakan yaitu Kontrol Logika Fuzzy (KLF).

3.2 Perancangan Pembuatan Sistem

3.2.1 Perancangan Perangkat Keras

- a. Perancangan blok diagram.
- b. Perancangan sensor *air flow*.
- c. Perancangan program KLF menggunakan Arduino Mega2560.
- d. Perancangan pengujian keseluruhan perangkat keras.

3.2.2 Perancangan Perangkat Lunak

Terlebih dahulu dengan pembuatan *flowchart*, kemudian penulisan *listing code* program KLF pada *software* Arduino Mega2560.

3.3 Pengujian Perangkat Keseluruhan

Pengujian perangkat keseluruhan dilakukan untuk mengetahui kinerja alat dan sistem sesuai dengan fungsinya berdasarkan blok diagram yang dibuat. Sistem dengan *overshoot* dan *error steady state* seminimal mungkin adalah harapan yang diinginkan dari sistem pengendalian kecepatan aliran udara ini.

3.4 Pengambilan Kesimpulan

Kesimpulan diperoleh berdasarkan data yang didapat dari hasil pengujian sistem secara keseluruhan. Sistem telah memenuhi harapan jika hasil yang didapat sesuai dengan yang direncanakan dan dapat dikembangkan untuk kedepannya.



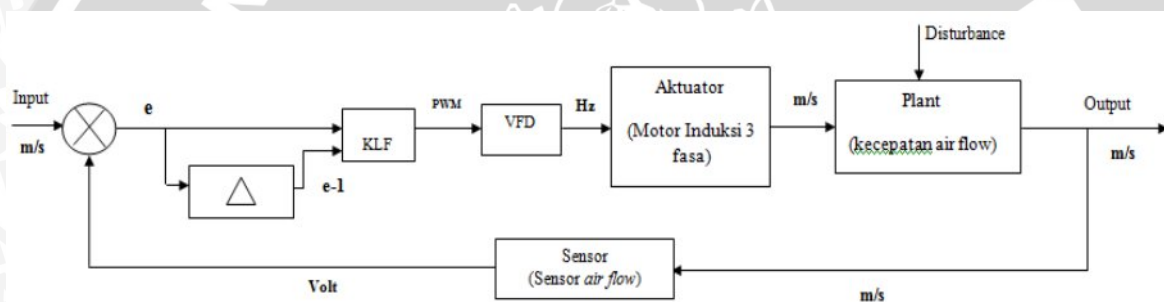
BAB IV

PERANCANGAN DAN PEMBUATAN ALAT

Pada bab ini akan dibahas meliputi perancangan dan pembuatan sistem kontrol aliran udara menggunakan Kontrol Logika Fuzzy (KLF). Untuk mengetahui dinamika aliran fluida, maka perancangan sistem tersebut meliputi perancangan perangkat keras maupun perancangan perangkat lunak.

4.1 Diagram Blok Sistem

Dalam laporan ini dibuat diagram blok agar dalam pengerjaan dapat dilakukan sesuai dengan rancangan sistem. Berikut diagram blok tersebut dapat dilihat pada Gambar 4.1.



Gambar 4. 1 Diagram Blok Sistem

Diagram blok dalam Gambar 4.1 dijelaskan sebagai berikut:

1. *Input* berupa kecepatan aliran udara diberikan melalui *program* pada Arduino Mega2560.
2. Nilai *setpoint* dan *feedback* diolah oleh kontroler untuk memperoleh nilai *crisp error* (e) dan *crisp change error* ($e-1$).
3. Proses pengolahan dimulai dari proses *fuzzifikasi*, *inferensi* dan *defuzzifikasi*. Dan nilai keluaran dari proses *defuzzifikasi* yaitu 0 sampai 255 berupa PWM.
4. Kemudian VFD membaca nilai PWM yang selanjutnya digunakan untuk mengubah frekuensi pada motor induksi AC 3 fasa (0-400 Hz).
5. Selanjutnya motor induksi AC 3 fasa menggerakkan *propeller* dan menghasilkan aliran udara.

6. Kecepatan aliran udara dibaca oleh sensor *air flow* dengan *output* berupa tegangan *analog*.
7. Kemudian sensor *air flow* membaca dan nilai *setpoint* dikurangkan dengan nilai *output* sensor, sehingga mikrokontroler mampu mengolah *Error* yang terjadi.

4.2 Perancangan Sistem

Adapun perancangan miniatur alat ini terdiri dari:

1. Perancangan perangkat keras.
2. Perancangan perangkat lunak (perancangan algoritma KLF).

4.3 Perancangan Perangkat Keras

4.3.1 Baling – Baling (*Propeller*)

Berdasarkan spesifikasi ukuran *wind tunnel* 150cmx150cm, maka *propeller* yang digunakan yaitu berukuran 147cm. Berikut adalah gambar bentuk fisik *propeller* yang digunakan, dan dapat dilihat pada Gambar 4.2.



Gambar 4.2 Bentuk Fisik *Propeller*

4.3.2 Arduino Mega2560

Perancangan ini menggunakan modul Arduino Mega2560 sebagai pusat proses pengaturan kecepatan motor induksi 3 *fasa* untuk menggerakkan *propeller*. Gambar 4.3 di bawah merupakan Arduino Mega2560.



Gambar 4.3 Tampak depan Arduino mega 2560

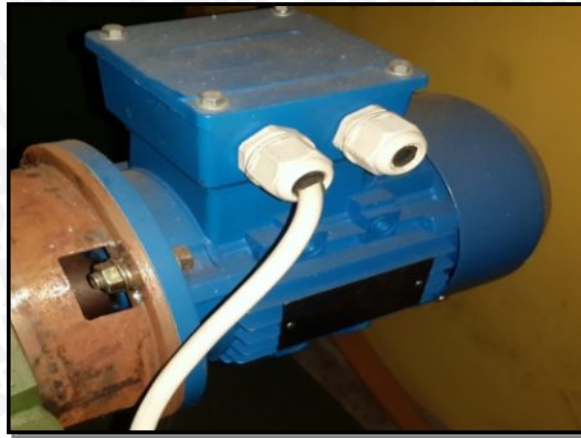
Untuk mendukung mikrokontroler, modul ini memiliki spesifikasi yaitu 54 *pin* digital *input/output*, di mana 15 *pin* dapat digunakan sebagai *output PWM*, 16 *pin* sebagai *input analog*, dan 4 *pin* sebagai *UART (port serial hardware)*, 16 *MHz* kristal osilator, koneksi *USB*, *jack power*, *header ICSP*, dan tombol *reset*. Berikut fungsi *pin* pada Arduino Mega2560 dapat dilihat pada Tabel 4.1.

Tabel 4.1 Fungsi Pin Arduino Mega 2560

No	Pin	Fungsi
1	A0	Masukan <i>Pre-Amplifier</i>
2	8	Masukan motor
3	GND	Jalur masukan GND seluruh sistem
4	Vin	Jalur masukan 5V seluruh sistem

4.3.3 Motor Induksi 3 Fasa

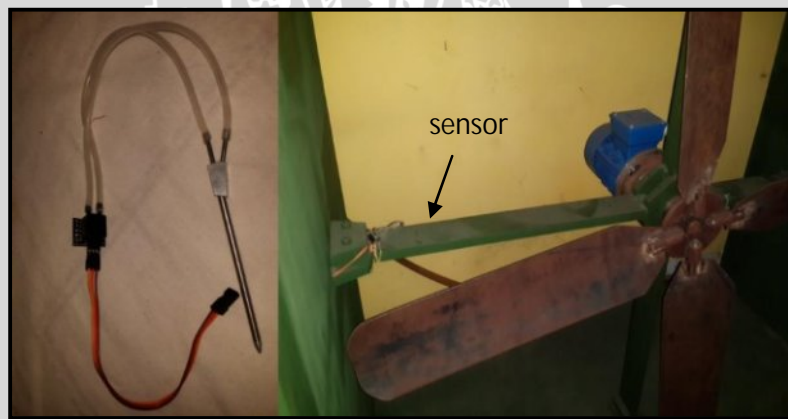
Dengan tegangan maksimum motor induksi 3 fasa adalah 4,8 *volt*, maka motor induksi ini dapat langsung terhubung ke Arduino Mega2560. Motor induksi 3 fasa ini memiliki *range* frekuensi *output* yaitu 0 sampai 400 *Hz*. Fungsi dari motor induksi ini adalah sebagai penggerak *propeller*. Gambar motor induksi 3 fasa diperlihatkan pada Gambar 4.4.



Gambar 4.4 Bentuk Fisik Motor Induksi 3 Fasa

4.3.4 Tabung Pitot (Sensor Air Flow)

Perancangan sistem ini menggunakan sensor *air flow* dengan tujuan untuk pembacaan nilai kecepatan aliran udara. Untuk nilai keluaran sensor sudah berupa tegangan analog. Peletakkan sensor berada pada rangka penyangga motor induksi, berikut diperlihatkan pada Gambar 4.5.



Gambar 4.5 Bentuk Fisik Sensor Air Flow

4.3.5 Variable Frequency Drive (VFD)

VFD merupakan sebuah alat pengatur kecepatan motor dengan mengubah nilai frekuensi pada motor. Pengaturan nilai frekuensi ini dimaksudkan untuk mendapatkan kecepatan putaran motor yang di inginkan. Dalam perancangan ini sangat dibutuhkan perubahan putaran motor induksi 3 fasa, di mana dibutuhkan putaran motor yang fleksibel untuk mengubah kecepatan putaran *propeller*. Pada Gambar 4.6 merupakan bentuk fisik VFD.



Gambar 4.6 Bentuk Fisik VFD

4.4 Perancangan Kontrol Logika Fuzzy (KLF)

4.4.1 Nilai *Input* dan *Output* KLF

Terdapat dua *crisp input* pada perancangan sistem ini, yaitu *crisp input Error* dan *crisp input Change Error* dan terdapat satu *crisp output*. Adapun nilai *Error* dan *Change Error* dapat dijelaskan kedalam persamaan (4.1) dan (4.2) dibawah ini.

$$Error = Sp - fb \quad (4.1)$$

dimana:

Sp = *Setpoint* (nilai *input* atau nilai yang diinginkan)
 fb = *feedback* (nilai pembacaan sensor)

$$Change Error (\Delta) = Error (e) - Error (e-1) \quad (4.2)$$

dimana:

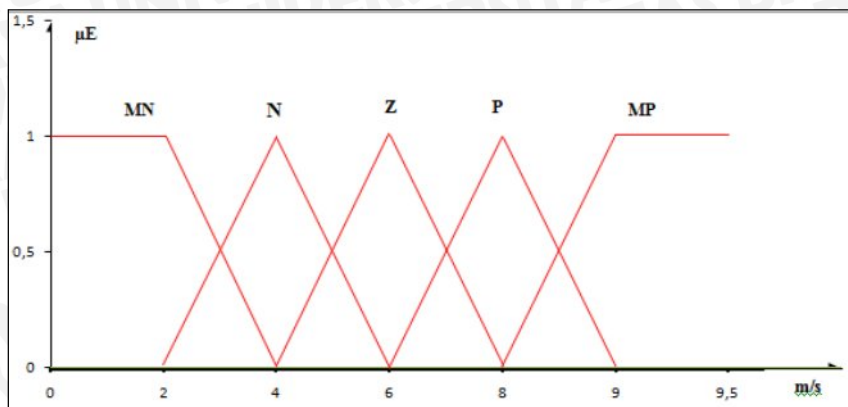
$Error (e)$ = *Error* pada waktu sekarang
 $Error (e-1)$ = *Error* pada waktu sebelumnya

4.4.2 Fungsi Keanggotaan

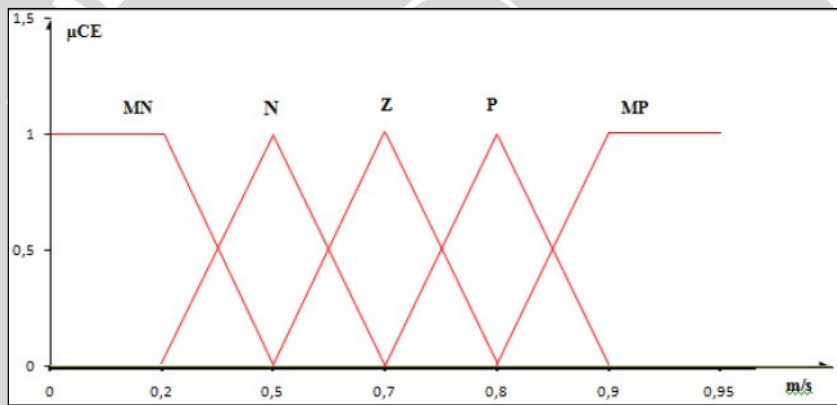
Fungsi keanggotaan merupakan sebuah kurva yang menggambarkan pemetaan dari masukan ke derajat keanggotaan antara 0 dan 1. Melalui fungsi keanggotaan yang telah disusun maka dari nilai-nilai masukan tersebut menjadi informasi fuzzy yang berguna nantinya untuk proses pengolahan secara fuzzy. Banyaknya jumlah fungsi keanggotaan dalam *fuzzy set* menentukan banyaknya aturan yang harus dibuat. Perancangan ini terdapat 5 suku dalam fungsi keanggotaan ini, yaitu *More Negative (MN)*, *Negative (N)*, *Zero (Z)*,

repository.ub.ac.id

Positive (P), dan More Positive (MP). Gambar input dari fungsi keanggotan Error dan Change Error dapat dilihat pada Gambar 4.7 dan Gambar 4.8.



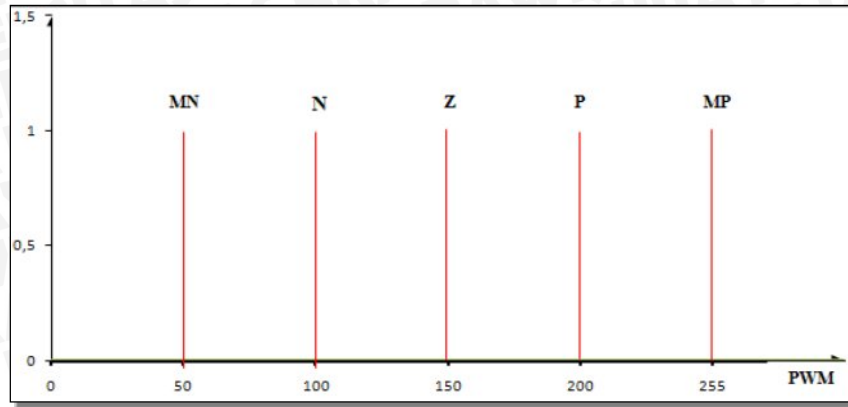
Gambar 4. 7 Fungsi Keanggotaan Input Error



Gambar 4. 8 Fungsi Keanggotaan Input Change Error

4.4.3 Fungsi Keanggotaan Keluaran

Setelah melalui proses pengelompokkan *input crips error*, kemudian didapat fungsi keanggotaan *crisp output*. Fungsi keanggotaan keluaran ini merupakan *input* pada motor induksi AC 3 fasa, dengan *input* yaitu berupa nilai PWM dengan hasil perhitungan *defuzzikasi*. Nilai frekuensi yang diinginkan yaitu sebesar 306 Hz, sesuai dengan *set point* yang diinginkan yaitu 10 m/s. Berikut merupakan fungsi keanggotaan keluaran dapat dilihat pada Gambar 4.9.



Gambar 4. 9 Fungsi Keanggotaan Keluaran PWM

4.4.4 Perancangan Aturan Fuzzy

Fungsi aturan *fuzzy* adalah sebagai penghubung informasi yang diberikan dengan tindakan (*action*). Fungsi ini melakukan penalaran menggunakan *fuzzy input* dan *fuzzy rule* yang telah ditentukan sehingga menghasilkan *fuzzy output* sebagai penentu keluaran. Dari data yang dibuat, terdapat 5 suku fungsi *input* dan memberikan 25 macam *rule*. Berikut merupakan *fuzzy rule* yang dibuat, dapat dilihat pada Tabel 4.2.

Tabel 4.2 Fuzzy Rule

		ERROR				
		MN	N	Z	P	MP
CHANGE ERROR	MN	MP	MP	MP	P	Z
	N	MP	MP	P	Z	N
	Z	P	P	Z	N	MN
	P	P	Z	N	MN	MN
	MP	Z	N	MN	MN	MN

4.4.5 Metode Inferensi Max-Min

Pada *set point* 10 m/s, dilakukan pembuktian untuk beberapa perhitungan dengan mendapatkan *error* dan *change error*, berikut hasil perhitungan dapat dilihat data pada Tabel 4.3.

Tabel 4.3 Tabel Data Hasil Keluaran

Data ke	Crisp E	Crisp CE
1	3,5	0,3
2	4,5	0,55

Perhitungan fuzzifikasi sistem :

1. $E/CE = 3,5/0,3$

$$\mu_E (MN = 3,5) = (4 - 3,5) : 2 = 0,25$$

$$\mu_E (N = 3,5) = (3,5 - 2) : 2 = 0,75$$

$$\mu_{CE} (MN = 0,3) = (0,5 - 0,3) : 0,3 = 0,6$$

$$\mu_{CE} (N = 0,3) = (0,3 - 0,2) : 0,3 = 0,3$$

Perhitungan inferensi pada sistem didapat :

$$MP = 0,25 \quad MP = 0,3 \quad MP = 0 \quad P = 0 \quad Z = 0$$

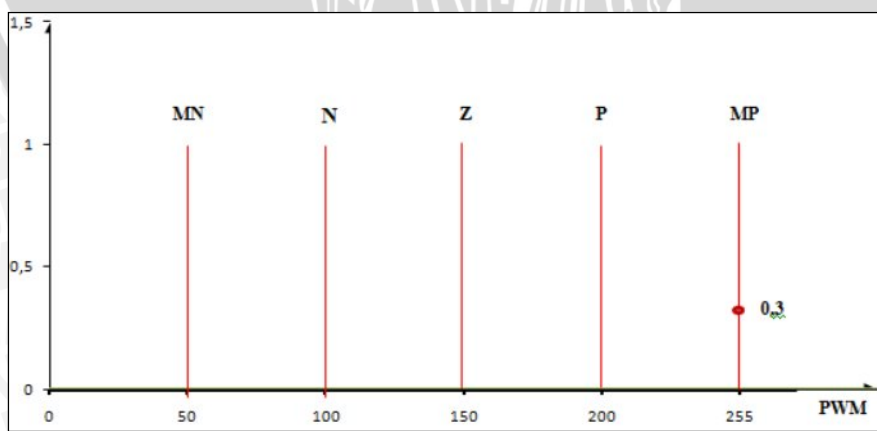
$$MP = 0,25 \quad MP = 0,3 \quad P = 0 \quad Z = 0 \quad N = 0$$

$$MP = 0 \quad P = 0 \quad Z = 0 \quad N = 0 \quad MN = 0$$

$$P = 0 \quad Z = 0 \quad N = 0 \quad MN = 0 \quad MN = 0$$

$$Z = 0 \quad N = 0 \quad MP = 0 \quad MN = 0 \quad MN = 0$$

Fungsi keanggotaan keluaran dapat dilihat pada Gambar 4.10 berikut :



Gambar 4. 10 Gambar Fungsi Keanggotaan Keluaran dengan MP = 0,3

2. $E/Ce = 4,5/0,55$

$\mu_E (Z = 4,5) = (4,5 - 4) : 2 = 0,25$

$\mu_E (N = 4,5) = (6 - 4,5) : 2 = 0,75$

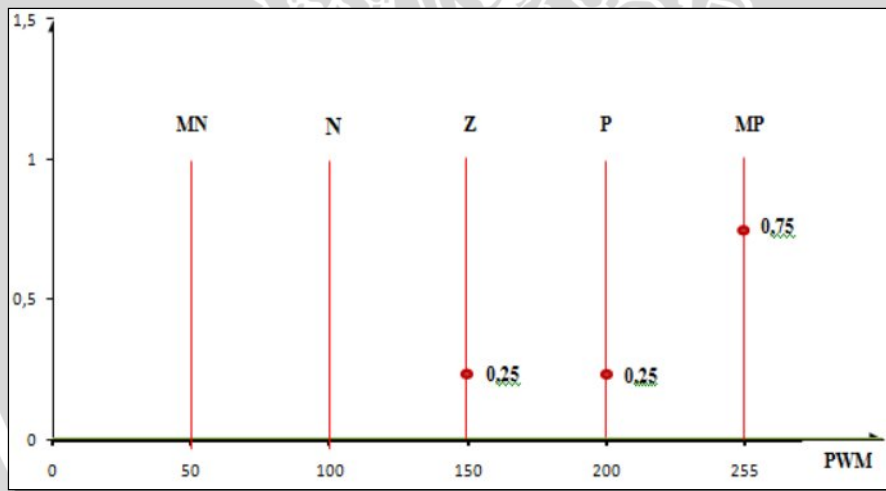
$\mu_{Ce} (Z = 0,55) = (0,55 - 0,5) : 0,2 = 0,25$

$\mu_{Ce} (N = 0,55) = (0,7 - 0,55) : 0,2 = 0,75$

Perhitungan pada sistem didapat :

MP = 0	MP = 0	MP = 0	P = 0	Z = 0
MP = 0	MP = 0,75	P = 0,25	Z = 0	N = 0
MP = 0	P = 0,25	Z = 0,25	N = 0	MN = 0
P = 0	Z = 0	N = 0	MN = 0	MN = 0
Z = 0	N = 0	MP = 0	MN = 0	MN = 0

Fungsi keanggotaan keluaran dapat dilihat pada Gambar 4.11 berikut :



Gambar 4. 11 Gambar Fungsi Keanggotaan Keluaran dengan $Z = 0,25$, $P = 0,25$, $MP = 0,75$

4.4.6 Metode Defuzzifikasi

Defuzzifikasi merupakan transformasi yang menyatakan kembali *output* dari domain fuzzy ke dalam domain *crisp*. *Output* fuzzy diperoleh melalui eksekusi dari beberapa fungsi keanggotaan fuzzy. Kemudian nilai PWM diproses melalui proses defuzzifikasi ini, dan hasilnya akan dieksekusi oleh motor induksi 3 fasa. Untuk metode

yang digunakan yaitu *Weighted Average* (WA), berikut penyelesaian masalah pada metode fuzzifikasi ini.

1. Proses defuzzifikasi untuk data pertama :

$$z^* = \frac{\sum_{i=1}^n \mu_{\bar{z}}(\bar{z}) \cdot \bar{z}}{\sum_{i=1}^n \mu_{\bar{z}}(\bar{z})}$$

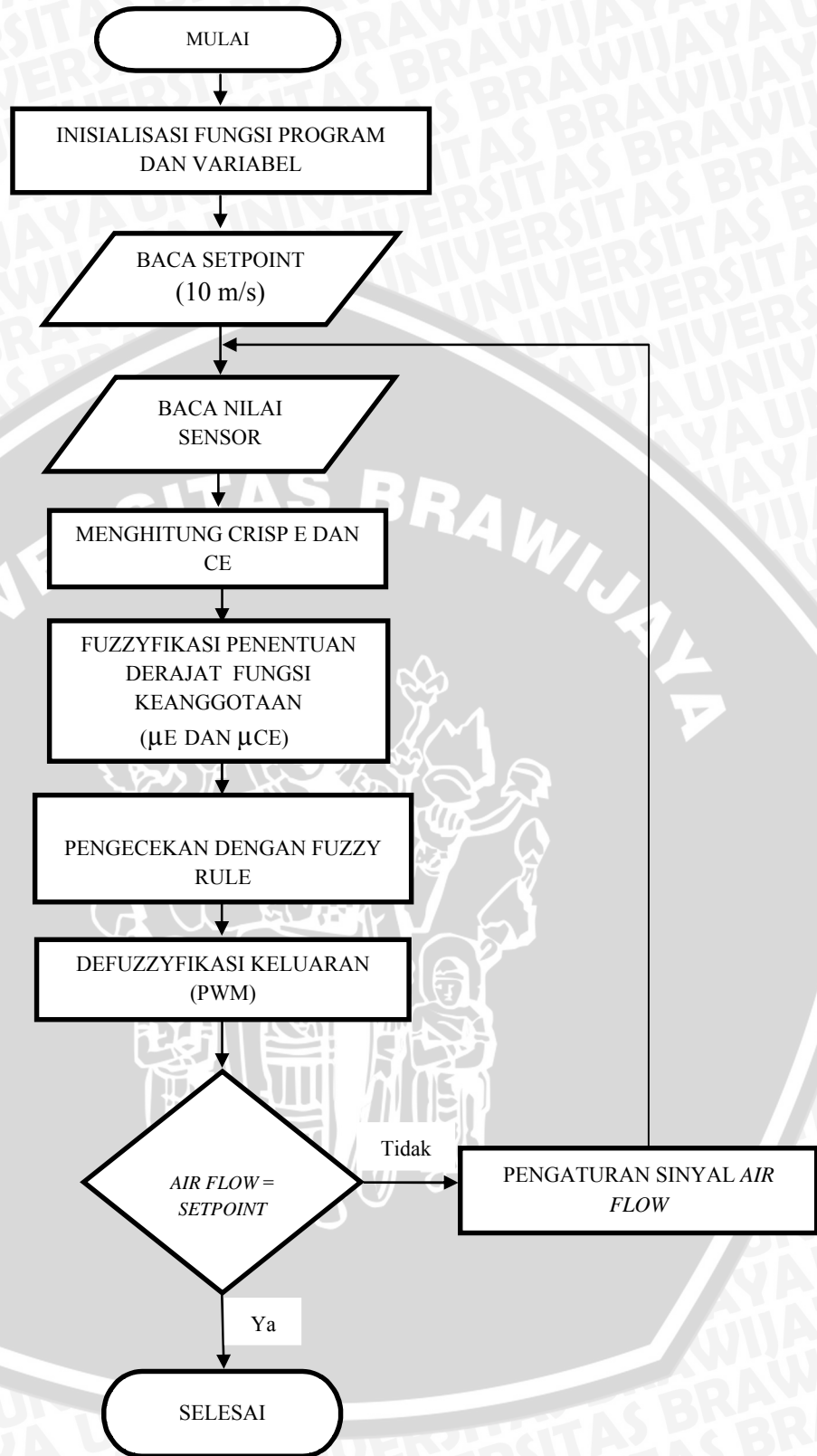
2. Proses defuzzifikasi untuk data kedua :

$$z^* = \frac{\sum_{i=1}^n \mu_{\bar{z}}(\bar{z}) \cdot \bar{z}}{\sum_{i=1}^n \mu_{\bar{z}}(\bar{z})}$$

Sistem dapat dinyatakan mampu apabila sudah melalui proses defuzzifikasi ini. Terkadang hasil dari fuzzifikasi merupakan bilangan yang tidak bulat, maka fungsi *integer* telah membulatkan nilai *output* dari proses fuzzifikasi ini, sehingga *input* dari motor induksi AC 3 fasa menerima bilangan bulat.

4.5 Perancangan Perangkat Lunak

Untuk perancangan perangkat lunak ini menggunakan *software* Arduino Mega2560. Berikut struktur *flowchat* dapat dilihat pada Gambar 4.12.



Gambar 4.12 Flowchart Perangkat Lunak

BAB V

PENGUJIAN DAN ANALISIS

Pengujian ini bertujuan apakah alat yang dirancang berfungsi sesuai dengan yang diharapkan, juga untuk mengetahui tingkat kinerja dan nilai pada masing-masing tiap blok komponen, diantaranya yaitu sensor dan aktuator. Pengujian perancangan alat dapat dilihat melalui blok diagram, tujuannya untuk mempermudah dalam perbaikan ketika terjadi kerusakan atau *error* pada setiap alat (pada setiap blok).

Berikut langkah – langkah pengujian yang dilakukan :

1. Pengujian respon sensor *air flow* tanpa kontroler (pengujian *open loop*)
2. Pengujian respon aktuator tanpa kontroler (pengujian *open loop*)
3. Pengujian keseluruhan sistem meliputi sensor, aktuator serta menggunakan kontroler (pengujian *close loop*)

5.1 Pengujian Sensor *Air Flow*

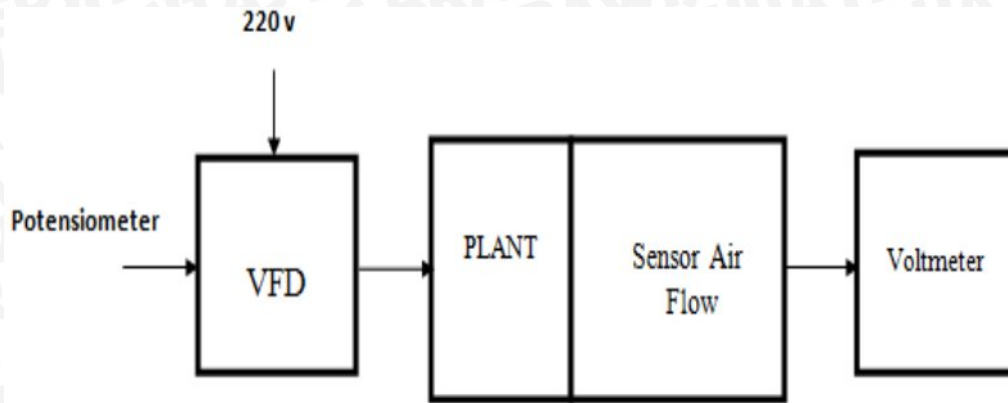
a. Tujuan

Pengujian sensor dilakukan untuk memastikan bahwa sensor telah bekerja dengan baik atau tidak, selain itu untuk mendapatkan hubungan antara kecepatan putaran motor induksi 3 fasa, aliran udara dan tegangan.

b. Peralatan yang digunakan

1. Laptop
2. Sensor *Air Flow*
3. Motor Induksi 3 Fasa, 3/4 HP, 220 volt
4. *Voltmeter*
5. Arduino Mega2560
6. Potensiometer
7. Peralatan alat uji *wind tunnel*
8. Rangkaian *VFD*
9. Program dan *software* Arduino Mega2560

Untuk diagram blok pengujian sensor *air flow* dapat dilihat pada Gambar 5.1.



Gambar 5.1 Diagram Blok Pengujian Sensor Air Flow

c. Langkah Pengujian

1. Hubungkan *plant* dengan sensor *air flow*, rangkaian VFD, dan Arduino Mega2560.
2. Hidupkan motor induksi 3 fasa dengan batas minimal mesin berputar (jangan sampai mesin *off*).
3. Perlahan naikan putaran motor induksi 3 fasa hingga batas maksimal putaran motor.
4. Pengujian ini telah ditetapkan nilai putaran maksimal motor induksi 3 fasa yaitu 11.040 *rpm*.
5. Selanjutnya sensor akan membaca sesuai dengan kecepatan putaran motor induksi 3 fasa.
6. Catat hasil keluaran kecepatan putaran, kecepatan aliran udara dan tegangan keluaran sensor, kemudian buat hubungan antar ketiga nilai.

d. Hasil Pengujian

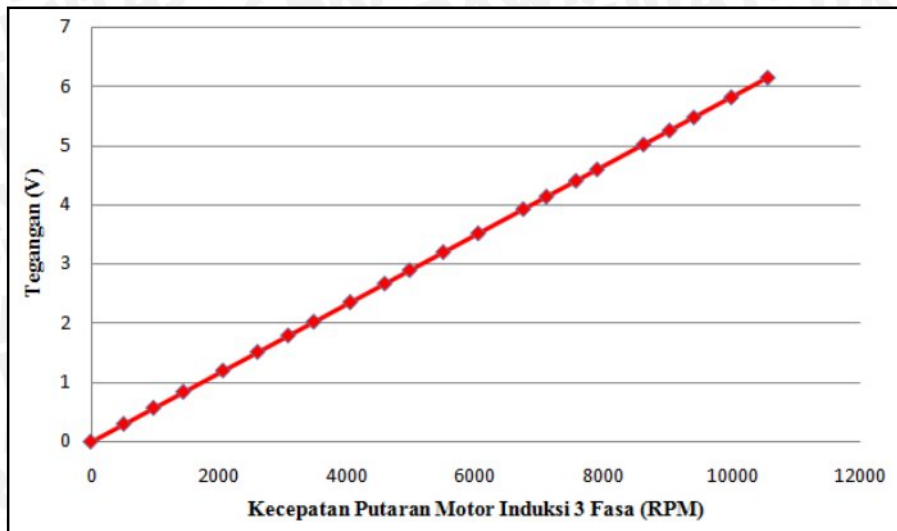
Data hasil pengujian sensor *air flow* dapat dilihat pada Tabel 5.1.

Tabel 5. 1 Hasil Pengujian Sensor Air Flow

No	Kecepatan Putaran (<i>rpm</i>)	Kecepatan Air Flow (<i>m/s</i>)	Tegangan Keluaran (<i>Volt</i>)
1	0	0	0
2	515	0,60	0,30

3	979	1,14	0,57
4	1443	1,68	0,84
5	2063	2,41	1,20
6	2601	3,03	1,52
7	3078	3,59	1,79
8	3476	4,05	2,03
9	4044	4,72	2,36
10	4581	5,34	2,67
11	4970	5,80	2,90
12	5495	6,41	3,20
13	6038	7,04	3,52
14	6742	7,86	3,93
15	7101	8,28	4,14
16	7562	8,82	4,41
17	7891	9,20	4,60
18	8614	10,04	5,02
19	9019	10,52	5,26
20	9398	10,96	5,48
21	9982	11,64	5,82
22	10549	12,30	6,15
23	11040	13,04	6,25

Setelah melakukan pengujian sensor *air flow*, didapat grafik antara kecepatan putaran motor induksi 3 fasa terhadap tegangan keluaran, grafik dapat dilihat pada Gambar 5.2.



Gambar 5. 2 Grafik Hubungan Kecepatan Putaran Motor dengan Tegangan Keluaran Sensor *Air Flow*

5.2. Pengujian Respon Aktuator (motor induksi 3 fasa)

a. Tujuan

Pengujian ini dilakukan untuk mengetahui apakah motor induksi 3 fasa ini bekerja sesuai fungsinya, yaitu sebagai penggerak *propeller*. Selain itu juga untuk mendapatkan hubungan tegangan keluaran terhadap waktu, sebagai dasar untuk menentukan *fuzzy set* dan pengambilan keputusan pada *rule-rule*.

b. Peralatan yang digunakan

1. Sensor *Air Flow*
2. Motor Induksi 3 Fasa, 3/4 HP, 220 volt
3. Peralatan alat uji *wind tunnel*
4. Catu daya 12 V DC
5. Rangkaian *VFD*
6. *Voltmeter*
7. *Oscilloscope* digital

c. Langkah pengujian

1. Hubungkan *VFD* dengan motor induksi 3 fasa.
2. Atur putaran motor induksi 3 fasa pada 0 rpm.
3. Naikkan putaran motor sampai putaran maksimum.

4. Kemudian catat tegangan keluaran sensor *air flow* yang ditampilkan pada *oscilloscope* digital.

d. Hasil Pengujian

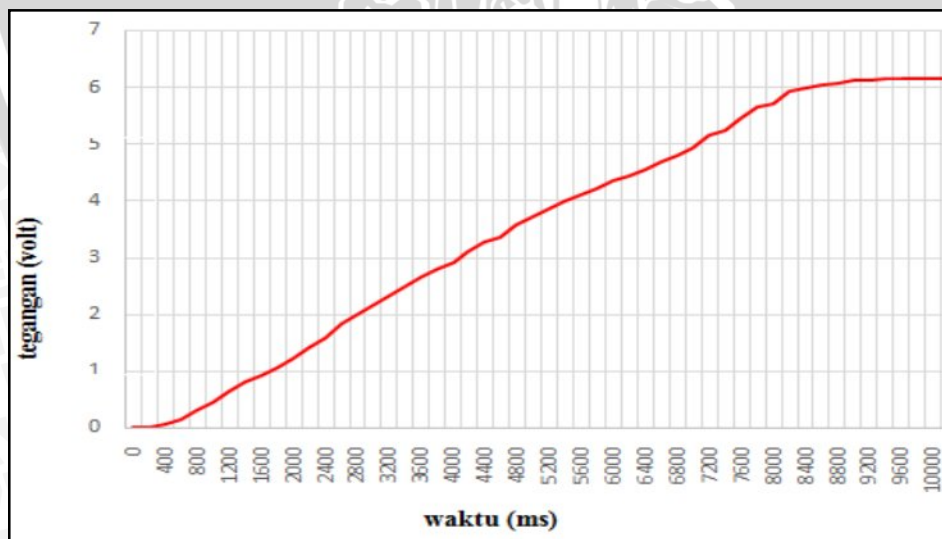
Adapun hasil pengujian aktuator juga didapat dilihat pada Tabel 5.2.

Tabel 5.2 Hasil Pengujian Aktuator (motor induksi 3 fasa)

t (ms)	V out (volt)
0	0
200	0,001
400	0,05
600	0,15
800	0,31
1000	0,45
1200	0,65
1400	0,8
1600	0,9
1800	1,05
2000	1,23
2200	1,42
2400	1,57
2600	1,82
2800	1,98
3000	2,15
3200	2,32
3400	2,49
3600	2,65
3800	2,79
4000	2,92
4200	3,1
4400	3,28
4600	3,35
4800	3,57
5000	3,72
5200	3,84
5400	3,98
5600	4,1
5800	4,22
6000	4,35
6200	4,42
6400	4,55

6600	4,69
6800	4,78
7000	4,92
7200	5,15
7400	5,23
7600	5,46
7800	5,64
8000	5,71
8200	5,92
8400	5,98
8600	6,04
8800	6,05
9000	6,12
9200	6,13
9400	6,14
9600	6,15
9800	6,15
10000	6,15
10200	6,15
10400	6,15

Berikut grafik keluaran tegangan motor induksi AC 3 fasa terhadap waktu, grafik dapat dilihat pada Gambar 5.3.



Gambar 5.3 Grafik Hubungan Tegangan Keluaran Motor Terhadap Waktu

5.3 Pengujian Keseluruhan Sistem

Pengujian keseluruhan sistem pada percobaan ini adalah pengujian sistem menggunakan kontroler (*close loop*).

a. Tujuan

Tujuan pengujian sistem ini adalah untuk mengetahui bagaimana respon sistem secara *close loop* jika diberi *setpoint*.

b. Peralatan yang digunakan

1. Laptop
2. Peralatan alat uji *wind tunnel*
3. Arduino Mega2560
4. Sensor *Air Flow*
5. Motor Induksi 3 Fasa, 3/4 HP, 220 volt
6. *VFD*
7. Catu daya 5 volt
8. Program dan *software* Arduino Mega2560
9. *Oscilloscope* digital

c. Langkah Pengujian

1. Unduh program dengan *setpoint* kecepatan aliran udara yaitu 10 m/s beserta program KLF pada *software* Arduino Mega2560.
2. Catat respon yang ditampilkan pada *oscilloscope* digital.
3. Memberikan gangguan dengan cara menghalangi aliran udara.
4. Mengamati dan catat hasil keluaran nilai kecepatan aliran udara, kemudian membuat grafik kecepatan aliran udara terhadap waktu.

d. Hasil Pengujian

Setelah melakukan prosedur sesuai pengujian diatas, hasil pengujian diperlihatkan pada Tabel 5.3.

Tabel 5.3 Hasil Pengujian Keseluruhan Sistem dengan *Setpoint* 10 m/s.

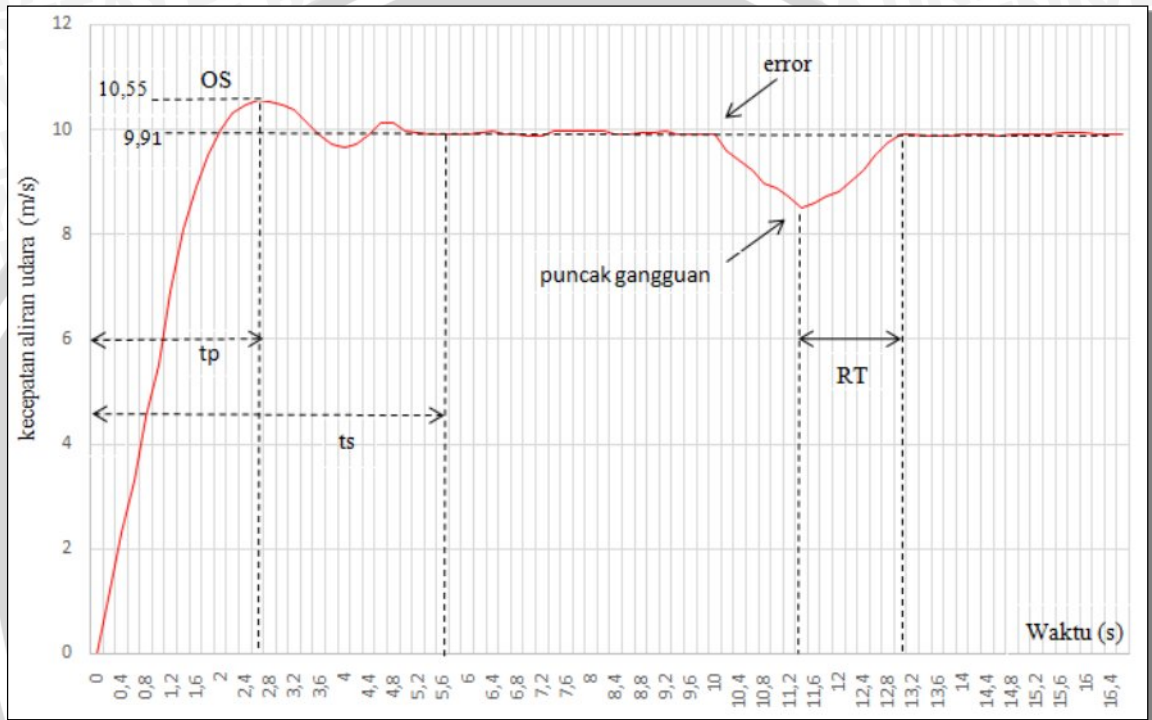
Waktu (s)	Kecepatan Aliran Udara (m/s)
0	0
0,2	1,2
0,4	2,3
0,6	3,3
0,8	4,5
1	5,5
1,2	6,9
1,4	8,1
1,6	8,9
1,8	9,5
2	10
2,2	10,3
2,4	10,45
2,6	10,55
2,8	10,52
3	10,45
3,2	10,35
3,4	10,1
3,6	9,9
3,8	9,7
4	9,65
4,2	9,7
4,4	9,9
4,6	10,1
4,8	10,1
5	9,95
5,2	9,93
5,4	9,9
5,6	9,91
5,8	9,91
6	9,91
6,2	9,94
6,4	9,95
6,6	9,91
6,8	9,91
7	9,86
7,2	9,97
7,4	9,96



7,6	9,97
7,8	9,97
8	9,96
8,2	9,95
8,4	9,91
8,6	9,91
8,8	9,92
9	9,94
9,2	9,95
9,4	9,91
9,6	9,91
9,8	9,91
10	9,9
10,2	9,6
10,4	9,4
10,6	9,2
10,8	8,97
11	8,87
11,2	8,7
11,4	8,5
11,6	8,6
11,8	8,7
12	8,8
12,2	9
12,4	9,2
12,6	9,5
12,8	9,75
13	9,91
13,2	9,91
13,4	9,86
13,6	9,86
13,8	9,87
14	9,91
14,2	9,91
14,4	9,91
14,6	9,98
14,8	9,9
15	9,9
15,2	9,9
15,4	9,91
15,6	9,92
15,8	9,93

16	9,92
16,2	9,91
16,4	9,91
16,6	9,91

Untuk grafik hasil pengujian secara *closed loop* keseluruhan sistem dapat dilihat pada Gambar 5.4.



Gambar 5. 4 Pengujian Sistem dengan *Setpoint* 10 m/s

Dari pengujian keseluruhan sistem, didapat respon sistem sebagai berikut :

1. *Peak Time* 2,6 terjadi pada detik 2,6
2. *Steady State* 5,6 terjadi pada detik 5,6
3. *Maximum Overshoot* sebesar
$$= \frac{10,55 - 9,91}{9,91} \times 100\%$$

= 6,46%
4. *Error Steady State (ESS)* sebesar
$$= \frac{10 - 9,91}{10} \times 100\%$$

= 0,9%
5. *Recovery Time* sebesar
$$= \frac{10 - 8,5}{10} \times 100\%$$



BAB VI PENUTUP

6.1 Kesimpulan

Dapat disimpulkan dari keseluruhan pengujian sistem yaitu :

1. Dengan penggunaan metode KLF sebagai kontrol kecepatan aliran udara dan diberi *setpoint* 10 m/s, sistem memberikan respon *Steady State* di detik 5,6, *Peak Time* pada detik 2,6, *Maximum Overshoot* (MO) 5,5%, *Error Steady State* (ESS) 0,45% .
2. Perubahan pada aliran udara terjadi karena sistem diberi gangguan yaitu dengan cara menghalangi aliran udara, dan menghasilkan respon *recovery time* (RT) sebesar 1600 ms.

6.2 Saran

Dari percobaan yang telah dilakukan, terdapat beberapa saran yang dapat dikembangkan, yaitu :

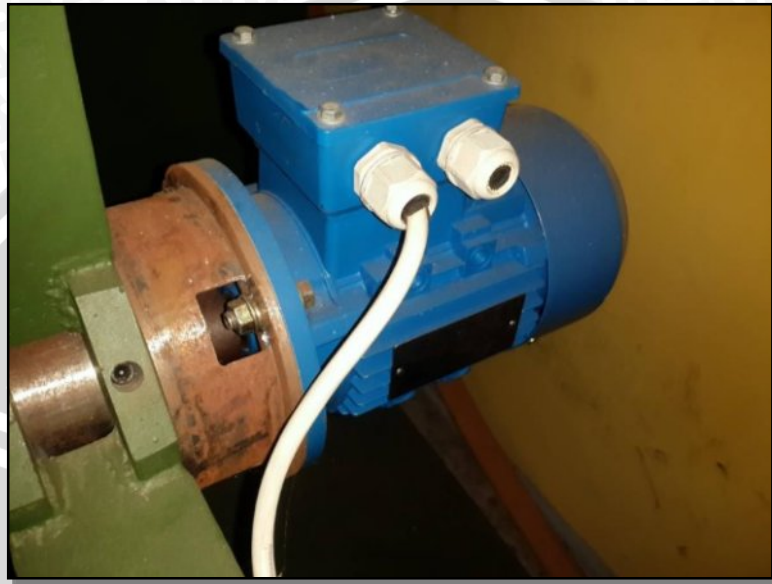
1. Perlu dilakukan beberapa analisis lebih lanjut dan perubahan pada rancangan *wind tunnel* agar didapatkan performa yang lebih optimal.
2. Dimensi modul seksi uji yang besar dapat dimanfaatkan untuk pengujian benda uji yang berdimensi besar juga.

DAFTAR PUSTAKA

- Yan, Jun., Ryan, Michael., Power, James. 1994. *Using Fuzzy Logic*. UK: Prentice Hall International.
- Prabowo, Brilliant Adhi, 2009. *Pemodelan Sistem Kontrol Motor DC dengan Temperatur Udara sebagai Pemicu*. Pusat Penelitian Informatika. LIPI.
- Ogata, K. 1997. *Teknik Kontrol Automatik Jilid 1*. Jakarta. Erlangga.
- Ogata, K. 1997. *Teknik Kontrol Automatik Jilid 2*. Jakarta. Erlangga.
- Muhammad, R. 2015. *Sistem Pengendalian Kecepatan Aliran Udara Pada Wind Tunnel Dengan Umpan Balik Kecepatan Aliran Udara Menggunakan Kontroler PID*. Malang. Teknik Elektro Universitas Brawijaya.
- Anthony, Z. 2011. *Pengaruh perubahan frekuensi dalam sistem pengendalian kecepatan motor induksi 3-fasa terhadap efisiensi motor*. Jurnal Teknik Elektro. ITP.
- Prof. Dr. Zulhal M.Sc.EE, Ir.Zhanggischan. 2004. *Prinsip Dasar Elektroteknik*. Jakarta. Gramedia Pustaka Utama.
- Munson, Donald. 2003. *Mekanika Fluida Edisi ke-4*. Jakarta. Erlangga.
- Arduino. "Arduino Mega2560 datasheet". 1 Mei 2016.
<https://www.arduino.cc/en/Main/ArduinoBoardMega2560.shtml>
- Wiki. "tabung pitot". 1 Mei 2016.
https://id.wikipedia.org/wiki/Tabung_pitot.shtml
- Nasa. "grc". 1 Mei 2016.
<http://www.nasa.gov/centers/glenn/home/index.html>

LAMPIRAN

Foto Alat



Motor Induksi 3 Fasa



Arduino Mega2560





Variable Frequency Drive (VFD)



Air Flow Sensor (Tabung Pitot)



Tampak depan wind tunnel

Datasheet



Freescale Semiconductor
Data Sheet: Technical Data

Document Number: MPXV7002
Rev 3, 01/2015

MPXV7002 Integrated Silicon Pressure Sensor On-Chip Signal Conditioned, Temperature Compensated and Calibrated

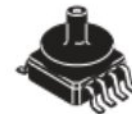
The MPXV7002 series piezoresistive transducers are state-of-the-art monolithic silicon pressure sensors designed for a wide range of applications, but particularly those employing a microcontroller or microprocessor with A/D inputs. This transducer combines advanced micromachining techniques, thin-film metallization, and bipolar processing to provide an accurate, high level analog output signal that is proportional to the applied pressure.

Features

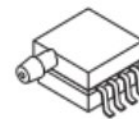
- 2.5% Typical Error over +10°C to +60°C with Auto Zero
- 6.25% Maximum Error over +10°C to +60°C without Auto Zero
- Ideally Suited for Microprocessor or Microcontroller-Based Systems
- Thermoplastic (PPS) Surface Mount Package
- Temperature Compensated over +10° to +60°C
- Patented Silicon Shear Stress Strain Gauge
- Available in Differential and Gauge Configurations

MPXV7002

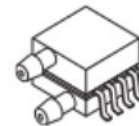
-2 to +2 kPa (-0.3 to +0.3 psi)
0.5 to 4.5 V Output



MPXV7002GC6U/C6T1
CASE 482A



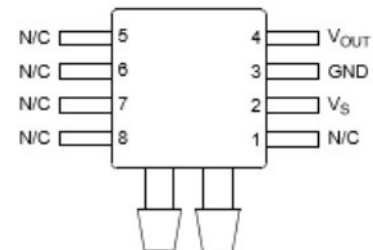
MPXV7002GP
CASE 1369



MPXV7002DP
CASE 1351

Small Outline Package

Top view



Pinout
(Style 2, case number 98A SA99255D)

ORDERING INFORMATION

Device Name	Package Options	Case No.	# of Ports			Pressure Type			Device Marking
			None	Single	Dual	Gauge	Differential	Absolute	
Small Outline Package (MPXV7002 Series)									
MPXV7002GC6U	Rails	482A		•		•			MPXV7002G
MPXV7002GC6T1	Tape & Reel	482A		•		•			MPXV7002G
MPXV7002GP	Trays	1369		•		•			MPXV7002G
MPXV7002DP	Trays	1351			•		•		MPXV7002DP



1 Operating Characteristics

Table 1. Operating Characteristics ($V_S = 5.0$ Vdc, $T_A = 25^\circ\text{C}$ unless otherwise noted. Decoupling circuit shown in [Figure 3](#) required to meet specification.)

Characteristic	Symbol	Min	Typ	Max	Unit
Pressure Range ⁽¹⁾	P_{OP}	-2.0	—	2.0	kPa
Supply Voltage ⁽²⁾	V_S	4.75	5.0	5.25	Vdc
Supply Current	I_o	—	—	10	mAdc
Pressure Offset ⁽³⁾ @ $V_S = 5.0$ Volts	V_{off}	2.25	2.5	2.75	Vdc
Full Scale Output ⁽⁴⁾ @ $V_S = 5.0$ Volts	V_{FSO}	4.25	4.5	4.75	Vdc
Full Scale Span ⁽⁵⁾ @ $V_S = 5.0$ Volts	V_{FSS}	3.5	4.0	4.5 V	Vdc
Accuracy ⁽⁶⁾	—	—	$\pm 2.5^{(7)}$	± 6.25	% V_{FSS}
Sensitivity	V/P	—	1.0	—	V/kPa
Response Time ⁽⁸⁾	t_R	—	1.0	—	ms
Output Source Current at Full Scale Output	I_{O+}	—	0.1	—	mAdc
Warm-Up Time ⁽⁹⁾	—	—	20	—	ms

1. 1.0 kPa (kiloPascal) equals 0.145 psi.

2. Device is ratiometric within this specified excitation range.

3. Offset (V_{off}) is defined as the output voltage at the minimum rated pressure.

4. Full Scale Output (V_{FSO}) is defined as the output voltage at the maximum or full rated pressure.

5. Full Scale Span (V_{FSS}) is defined as the algebraic difference between the output voltage at full rated pressure and the output voltage at the minimum rated pressure.

6. Accuracy (error budget) consists of the following:

- Linearity: Output deviation from a straight line relationship with pressure over the specified pressure range.
- Temperature Hysteresis: Output deviation at any temperature within the operating temperature range, after the temperature is cycled to and from the minimum or maximum operating temperature points, with zero differential pressure applied.
- Pressure Hysteresis: Output deviation at any pressure within the specified range, when this pressure is cycled to and from the minimum or maximum rated pressure, at 25°C .
- TcSpan: Output deviation over the temperature range of 10° to 60°C , relative to 25°C .
- TcOffset: Output deviation with minimum rated pressure applied, over the temperature range of 10° to 60°C , relative to 25°C .
- Variation from Nominal: The variation from nominal values, for Offset or Full Scale Span, as a percent of V_{FSS} , at 25°C .

7. Auto Zero at Factory Installation: Due to the sensitivity of the MPXV7002 Series, external mechanical stresses and mounting position can affect the zero pressure output reading. Auto zero is defined as storing the zero pressure output reading and subtracting this from the device's output during normal operations. Reference AN1636 for specific information. The specified accuracy assumes a maximum temperature change of $\pm 5^\circ\text{C}$ between auto zero and measurement.

8. Response Time is defined as the time for the incremental change in the output to go from 10% to 90% of its final value when subjected to a specified step change in pressure.

9. Warm-up Time is defined as the time required for the product to meet the specified output voltage after the Pressure has been stabilized.

MPXV7002

2 Maximum Ratings

Table 2. Maximum Ratings⁽¹⁾

Rating	Symbol	Value	Unit
Maximum Pressure ($P_1 > P_2$)	P_{max}	75	kPa
Storage Temperature	T_{stg}	-30 to +100	°C
Operating Temperature	T_A	10 to 60	°C

1. Exposure beyond the specified limits may cause permanent damage or degradation to the device.

Figure 1 shows a block diagram of the internal circuitry integrated on a pressure sensor chip.

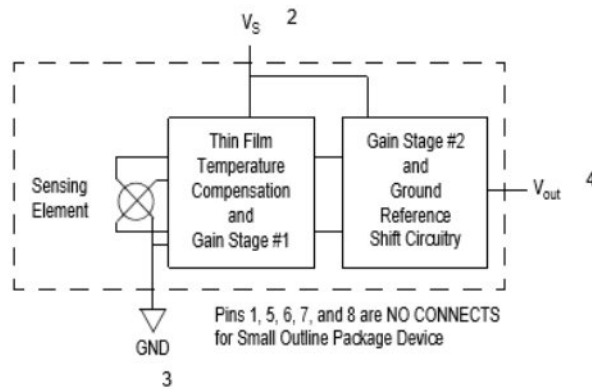


Figure 1. Integrated Pressure Sensor Schematic

3 On-Chip Temperature Compensation, Calibration and Signal Conditioning

The performance over temperature is achieved by integrating the shear-stress strain gauge, temperature compensation, calibration and signal conditioning circuitry onto a single monolithic chip.

Figure 2 illustrates the Differential or Gauge configuration in the basic chip carrier (Case 482). A gel die coat isolates the die surface and wire bonds from the environment, while allowing the pressure signal to be transmitted to the sensor diaphragm.

The MPXV7002 series pressure sensor operating characteristics, and internal reliability and qualification tests are based on use of dry air as the pressure media. Media, other than dry air, may have adverse effects on sensor performance and long-term reliability. Contact the factory for information regarding media compatibility in your application.

Figure 3 shows the recommended decoupling circuit for interfacing the integrated sensor to the A/D input of a microprocessor or microcontroller. Proper decoupling of the power supply is recommended.

Figure 4 shows the sensor output signal relative to pressure input. Typical, minimum, and maximum output curves are shown for operation over a temperature range of 10° to 60°C using the decoupling circuit shown in Figure 3. The output will saturate outside of the specified pressure range.

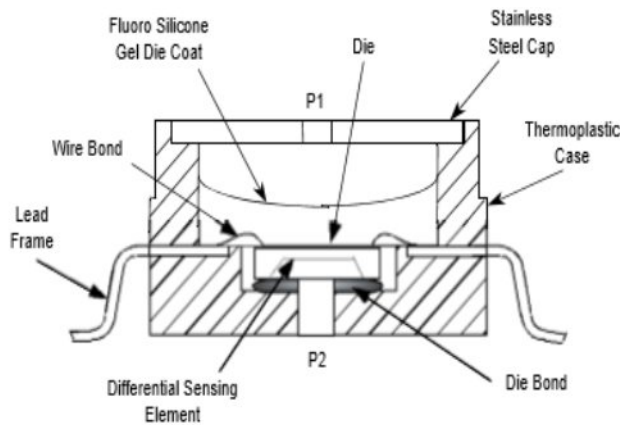


Figure 2. Cross-Sectional Diagram SOP (not to scale)

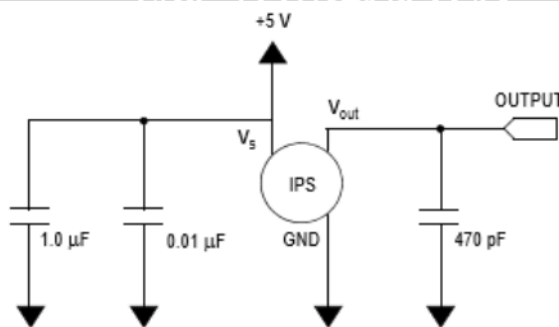


Figure 3. Recommended Power Supply Decoupling and Output Filtering (For additional output filtering, please refer to Application Note AN1646.)

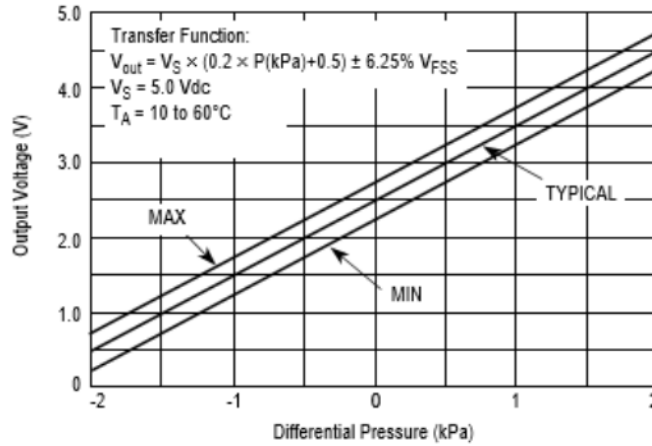


Figure 4. Output versus Pressure Differential

4 Pressure (P1)/Vacuum (P2) Side Identification Table

Freescale designates the two sides of the pressure sensor as the Pressure (P1) side and the Vacuum (P2) side. The Pressure (P1) side is the side containing a gel die coat which protects the die from harsh media.

The Pressure (P1) side may be identified by using the following table:

Part Number	Case Type	Pressure (P1) Side Identifier
MPXV7002GC6U/GC6T1	482A-01	Side with Port Attached
MPXV7002GP	1369-01	Side with Port Attached
MPXV7002DP	1351-01	Side with Part Marking

5 Minimum Recommended Footprint for Surface Mounted Applications

Surface mount board layout is a critical portion of the total design. The footprint for the surface mount packages must be the correct size to ensure proper solder connection interface between the board and the package. With the correct footprint, the packages will self align when subjected to a solder reflow process. It is always recommended to design boards with a solder mask layer to avoid bridging and shorting between solder pads.

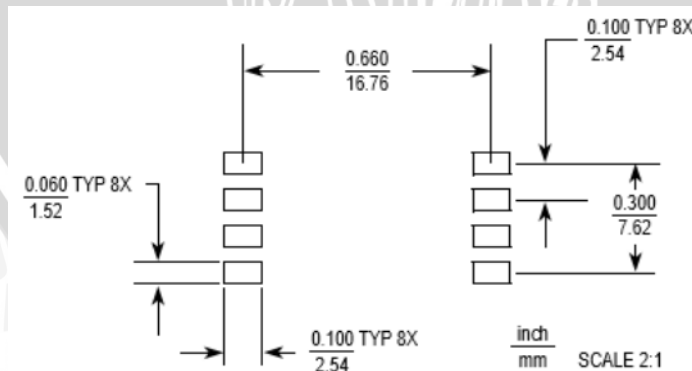


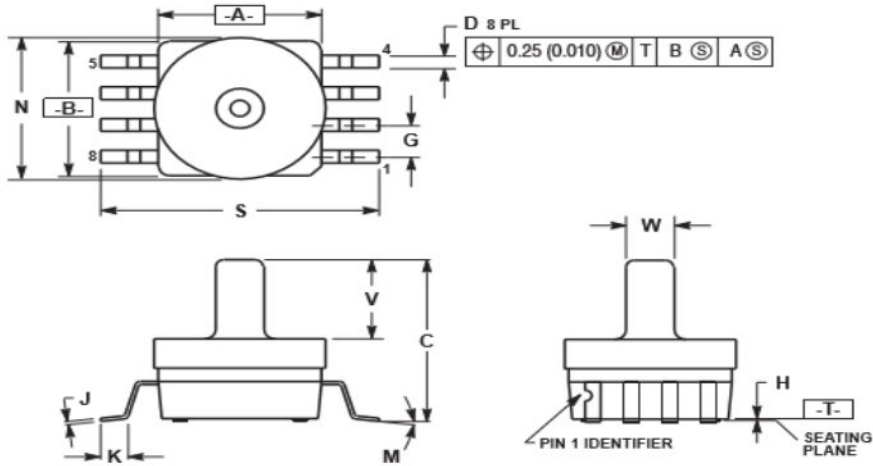
Figure 5. Small Outline Package Footprint

MPXV7002





6 Package Dimensions



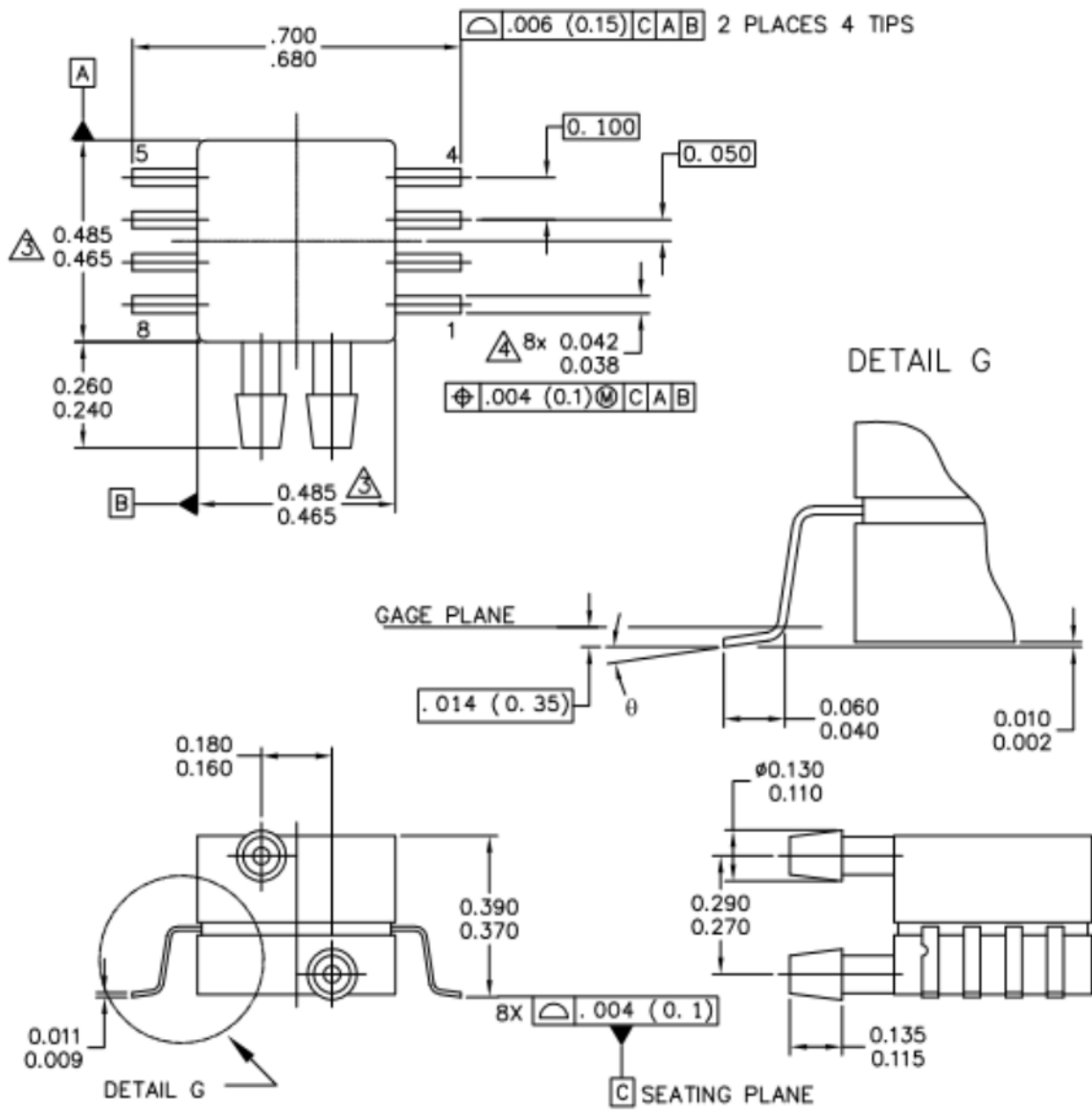
**CASE 482A-01
ISSUE A
SMALL OUTLINE PACKAGE**

NOTES:

1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: INCH.
3. DIMENSION A AND B DO NOT INCLUDE MOLD PROTRUSION.
4. MAXIMUM MOLD PROTRUSION 0.15 (0.006).
5. ALL VERTICAL SURFACES 5° TYPICAL DRAFT.

DIM	INCHES		MILLIMETERS	
	MIN	MAX	MIN	MAX
A	0.415	0.425	10.54	10.79
B	0.415	0.425	10.54	10.79
C	0.500	0.520	12.70	13.21
D	0.038	0.042	0.96	1.07
G	0.100 BSC		2.54 BSC	
H	0.002	0.010	0.05	0.25
J	0.009	0.011	0.23	0.28
K	0.061	0.071	1.55	1.80
M	0°	7°	0°	7°
N	0.444	0.448	11.28	11.38
S	0.709	0.725	18.01	18.41
V	0.245	0.255	6.22	6.48
W	0.115	0.125	2.92	3.17

MPXV7002



© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE	
TITLE: 8 LD SNSR, DUAL PORT	DOCUMENT NO: 98ASA99255D	REV: A	
	CASE NUMBER: 1351-01	27 JUL 2005	
	STANDARD: NON-JEDEC		

**CASE 1351-01
ISSUE A
SMALL OUTLINE PACKAGE**

MPXV7002





NOTES:

1. CONTROLLING DIMENSION: INCH
2. INTERPRET DIMENSIONS AND TOLERANCES PER ASME Y14.5M-1994.
3. DIMENSIONS DO NOT INCLUDE MOLD FLASH OR PROTRUSIONS. MOLD FLASH AND PROTRUSIONS SHALL NOT EXCEED .006 PER SIDE.
4. DIMENSION DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL BE .008 MAXIMUM.

STYLE 1:

PIN 1: GND
 PIN 2: +Vout
 PIN 3: Vs
 PIN 4: -Vout
 PIN 5: N/C
 PIN 6: N/C
 PIN 7: N/C
 PIN 8: N/C

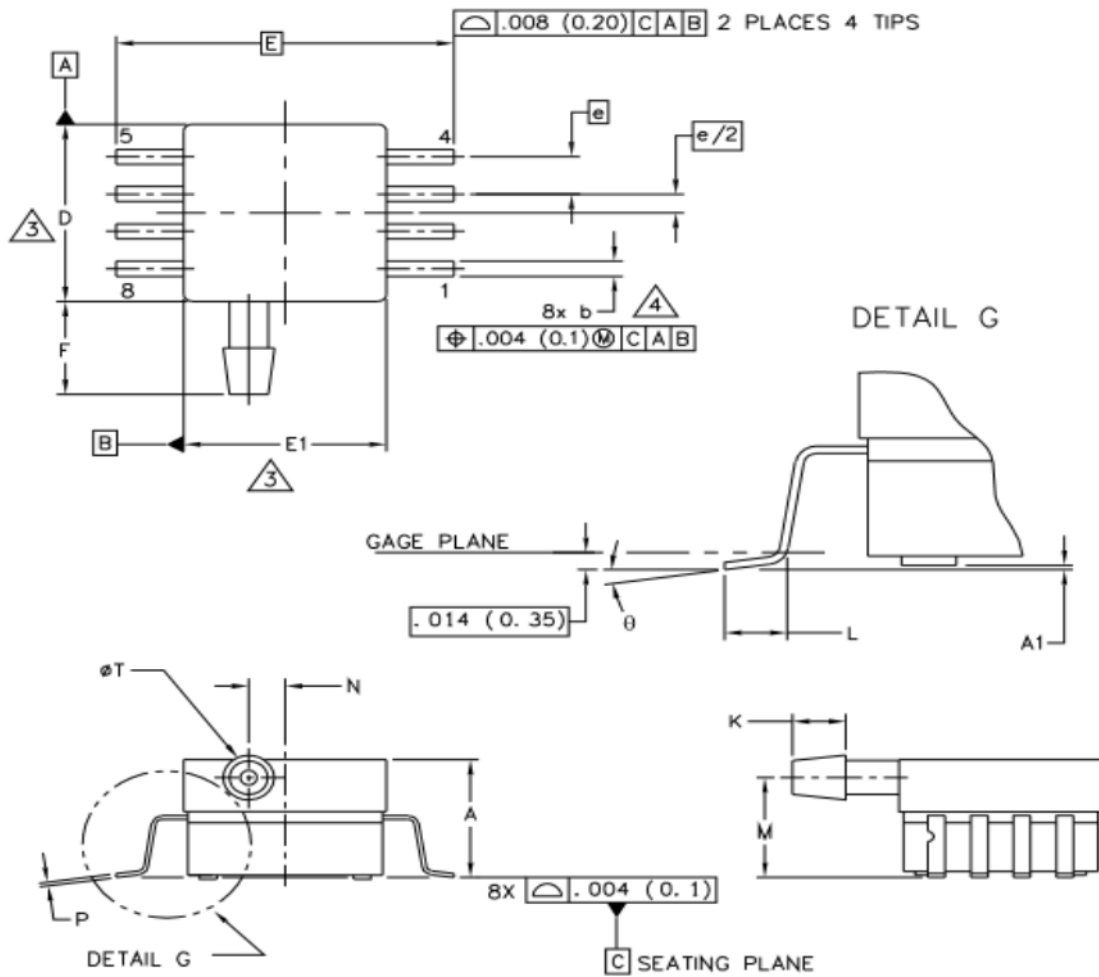
STYLE 2:

PIN 1: N/C
 PIN 2: Vs
 PIN 3: GND
 PIN 4: Vout
 PIN 5: N/C
 PIN 6: N/C
 PIN 7: N/C
 PIN 8: N/C

© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE	
TITLE: 8 LD SNSR, DUAL PORT	DOCUMENT NO: 98ASA99255D	REV: A	
	CASE NUMBER: 1351-01	27 JUL 2005	
	STANDARD: NON-JEDEC		

MPXV7002





© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE
	TITLE: 8 LD SOP, SIDE PORT	DOCUMENT NO: 98ASA99303D CASE NUMBER: 1369-01 STANDARD: NON-JEDEC

**CASE 1369-01
ISSUE B
SMALL OUTLINE PACKAGE**

MPXV7002





Huanyang Inverter

HY Series





I . Introduction


Thank you for purchasing and using the general-purpose inverter of HY series of multi-functions and high performance.


Please read carefully the operation manual before putting the inverter to use so as to correctly install and operate the inverter, give full play to its functions and ensure the safety. Please keep the operation manual handy for future reference, maintenance, inspection and repair.

Due to the inverter of a kind of electrical and electronic product it must be installed, tested and adjusted with parameters by specialized engineering persons of motors.

The marks of  **Danger**  **Caution** and other symbols in the manual remind you of the safety and prevention cautions during the handling, installation, running and inspection. Please follow these instructions to make sure the safe use of the inverter. In case of any doubt please contact our local agent for consultation. Our professional persons are willing and ready to serve you.

The manual is subject to change without notice.

 **Danger** indicates wrong use may kill or injure people.

 **Caution** indicates wrong use may damage the inverter or mechanical system.


Danger

- Be sure to turn off the input power supply before wiring.
- Do not touch any internal electrical circuit or component when the charging lamp is still on after the AC power supply is disconnected, which means the inverter still has high voltage inside and it is very dangerous.
- Do not check components and signals on the circuit boards during the operation.
- Do not disassemble or modify any internal connecting cord, wiring or component of the inverter by yourself.
- Be sure to make correct ground connection of the earth terminal of the inverter.
- Never remodel it or exchange control boards and components by yourself. It may expose you to an electrical shock or explosion, etc.

II. Safety Precautions

1. Before the Power-up

⚠ Caution

- Check to be sure that the voltage of the main circuit AC power supply matches the input voltage of the inverter.
- The symbol, , represents ground terminals. Be sure to make correct ground connection of the earth terminals of the motor and the inverter for safety.
- No contactor should be installed between the power supply and the inverter to be used for starting or stopping of the inverter. Otherwise it will affect the service life of the inverter.

⚡ Danger

- R.S.T terminals are power input terminals, never mixed with U.V.W terminals. Be sure that the wiring of the main circuit is correct. Otherwise it will cause damages of the inverter when the power is applied to it.

III. Standards and Specifications

1. Particular Specifications

Type	Input Voltage	Power (KW)	Inverter Capacity (KVA)	Output Current (A)	Suitable Motor (KW)
HY00D423B	Single & Three Phase 220V 50Hz	0.4	1.0	2.5	0.4
HY0D7523B	Single & Three Phase 220V 50Hz	0.75	2.0	5.0	0.75
HY01D523B	Single & Three Phase 220V 50Hz	1.5	2.8	7.0	1.5
HY02D223B	Single & Three Phase 220V 50Hz	2.2	4.4	11	2.2
HY0D7543B	3 Φ 380V 50Hz	0.75	2.2	2.7	0.75
HY01D543B	3 Φ 380V 50Hz	1.5	3.2	4.0	1.5
HY02D243B	3 Φ 380V 50Hz	2.2	4.0	5.0	2.2
HY03D743B	3 Φ 380V 50Hz	3.7	6.8	8.5	3.7
HY05D543B	3 Φ 380V 50Hz	5.5	10	12.5	5.5
HY07D543B	3 Φ 380V 50Hz	7.5	14	17.5	7.5
HY001143B	3 Φ 380V 50Hz	11	19	24	11
HY001543B	3 Φ 380V 50Hz	15	26	33	15
HY18D543B	3 Φ 380V 50Hz	18.5	32	40	18.5
HY002243B	3 Φ 380V 50Hz	22	37	47	22
HY003043B	3 Φ 380V 50Hz	30	52	65	30
HY003743B	3 Φ 380V 50Hz	37	64	80	37
HY004543B	3 Φ 380V 50Hz	45	72	91	45
HY005543B	3 Φ 380V 50Hz	55	84	110	55

2. General Specifications

Inverter Series	HY-A	
Control Mode	SPWM	
Input Power	330~440Vfor 380V power; 170~240for 220V power	
5-Digits Display & Status Indicator Lamp	Displaying frequency, current, revolution, voltage, counter, temperature, forward or reserve running, and fault, etc.	
Communication Control	RS-485	
Operation Temperature	-10~40°C	
Humidity	0-95% Relative Humidity (without dew)	
Vibration	Below 0.5G	
Frequency Control	Range	0.10~400.00Hz
	Accuracy	Digital: 0.01% (-10~40°C), Analog: 0.1% (25±10°C)
	Setting Resolution	Digital: 0.01Hz, Analog: 1% of Max. Operating Frequency
	Output Resolution	0.01Hz
	Operator Setting Method	Press directly <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> to set.
	Analog Setting Method	External Voltage 0-5V, 0-10V, 4-20mA, 0-20mA.
	Other Functions	Frequency lower limit, starting frequency, stopping frequency, three skip frequencies can be respectively set.
Ramp Control	Ramp Control	Selectable 4-speed steps ramp-up and -down time (0.1-6500s).
	V/F Curve	Set V/F curve at will

	Torque Control	Torque increase is settable by max. 10.0%. The starting torque can reach 150% at 1.0Hz.
	Multi-Inputs	6 multi-function input terminals for 8-speed steps control, program operation, switching of 4-speed Ramp, UP/DOWN function, counter, external emergency stop and other functions.
	Multi-Outputs	5 multi-function output terminals for displaying of running, zero speed, counter, external abnormality, program operation and other information and warnings.
	Other Functions	AVR (auto voltage regulation), Deceleration stop or free-stop, DC brake, auto reset and restart, frequency track, PLC control, traverse function, drawing control, auto energy-savings, carrier adjustable by max. 16KHz, etc.
Protection Functions	Overload Protection	Electronic relay protection motor Drive (for constant torque 150%/1 min. for the kinds of fan 120%/1min.)
	FUSE Protection	FUSE broken, Motor stops.
	Over-voltage	DC Voltage >400V for 220V class DC Voltage >800V for 380V class
	Low Voltage	DC Voltage <200V for 220V class DC Voltage <400V for 380V class
	Instant Stop and Restart	Restarted by frequency track after instantaneous stop.
	Stall Prevention	Anti-stall during Acc/Dec run
	Output End Shorts	Electronic circuit protecting
	Other Functions	Fin over-heat protection, restriction of reverse running, direct start after power on, fault reset, parameter lock PID, one-drive-more, etc.

IV. Storage and Installation

1. Storage

The inverter must be kept in its original package box before installation. Pay attention to the followings when keeping it in storage if the inverter is not used for the time being:

- It must be stored in a dry place without rubbish or dust.
- The suitable temperature for storage is between -20°C and +65°C.
- The relative humidity required is 0-95% without condensation.
- There is no corrosive gas or liquid in the storage ambience.
- It's better to lay the inverter on a rack and keep it in a proper package.
- It is better not to store the inverter for long time. Long time storage of the inverter will lead to the deterioration of electrolytic capacity. If it needs to be stored for a long time make sure to power it up one time within a year and the power-up time should be at least above five hours. When powered up the voltage must be increased slowly with a voltage regulator to the rated

2. Installation Site and Environment

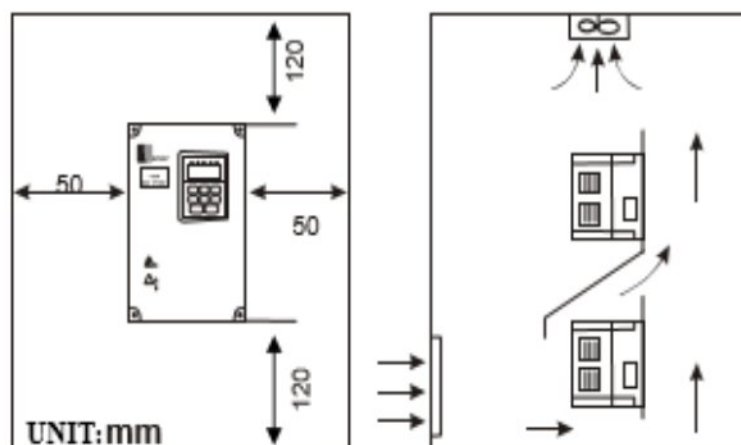
The inverter should be installed at the following location:

- Ambient temperature -5°C to 40°C with good ventilation.
- No water drop and low moisture.
- Free from direct sunshine, high temperature and heavy dust fall.
- Free from corrosive gas or liquid.
- Less dust, oil gas and metallic particles
- Free from vibration and easy for service and inspection.
- Free from the interference of electromagnetic noise.

Attention: The ambient conditions of the inverter will affect its service life.

3. Installation and Direction

- There must be enough space left around the inverter for easy maintenance and cooling. See Diagram .
- The inverter must be installed vertically with the smooth ventilation for effective cooling.
- If there is any instability when installing the inverter, please put a flat board under the inverter bottom base and install it again. If the inverter is installed on a loose surface, stress may cause damage of parts in the main circuit so as to damage the inverter.
- The inverter should be installed on non-combustible materials, such as iron plate.
- If several inverters are installed, upper and lower, together in one cabinet, please add heat dissipation plates and leave enough space between the inverters. See Diagram.



V. Wiring

1. Main Circuit Wiring Schematic Diagram(图标自己加)

Power supply:

- Verify that the inverter's rated voltage coincides with AC power supply voltage to avoid a damage of the inverter.

No fuse breaker:

- Refer to the related list.

Ground fault circuit interrupter:

- Use one of anti-high harmonic.

Electromagnetic contactor:

- Note: Do not use the electromagnetic contactor as the on/off button of power supply for the inverter.

AC reactor:

- It is recommended to install an AC reactor for power factor improvement if the input capacity is more than 1000KVA.

Inverter:

- Be sure to make correct connections of the main circuit wires and control signal wires of the inverter.
- Be sure to make correct setting of parameters for the inverter.



2. Description of Terminal Block

1) Arrangement of Main circuit Terminals (图标自行编制)

HY00D423B-HY01D223B

⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕

HY0D7543B-HY03D743B

E	R	S	T	U	V	W	P+	PR	
⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕

HY03D743B

HY02D223B-HY03D723B

E	R	S	T	U	V	W	P	PR	
⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕

HY05D543B-HY07D543B

E	R	S	T	U	V	W	P1	P	PR

⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕
---	---	---	---	---	---	---	---	---	---

HY001143B-HY003043B

HY05D523B-HY07D523B

R	S	T	E	P	N	U	V	W	
⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕

HY003743B-HY016043B

HY18D523B-HY002223B

R	S	T	E	P	P1	N	U	V	W
⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕



(2) Arrangement of Control Circuit Terminals

UPF	DRV	DCM	SPL	SPM	SPH	RST	REV	FOR	FOR		
	FA	FC	FB	DRV	+10	VI	AI	ACM	AM	RS-	RS+

3) Function Description of Main circuit Terminals

Symbol	Function Description
R.S.T	Input terminal of AC line power. (220V class, for both single/three phase, single phase connected to any two phases)
U.V.W	Output terminal of the inverter
P.Pr	Connector for braking resistor.
P ₁ P	Connector for DC reactor (When using a DC reactor the jumper shall be removed. A05D543B and A07D543B internally jumped)
P、N	Connecting terminal of external braking bank.
E	Ground terminal: the third method of grounding for 220V and special grounding for 380 V of Electrical Engineering Regulations.

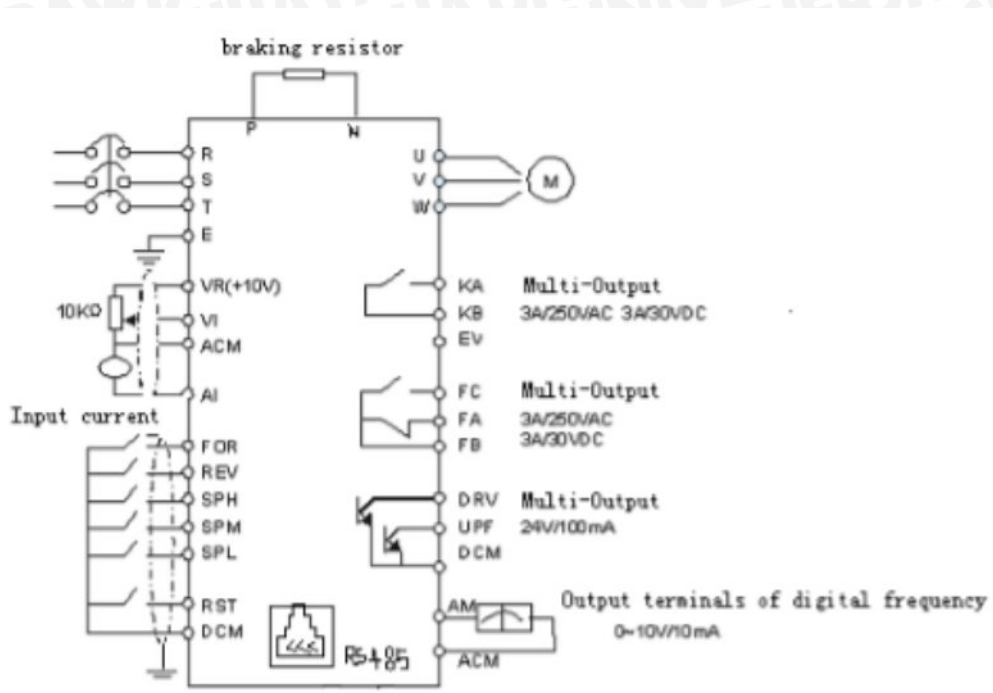
4) Function Description of Control Circuit Terminals

Symbol	Function Description	Factory setting
FOR	Multi-Input 1	Forward run
REV	Multi-Input 2	Reverse run
RST	Multi-Input 3	Reset
SPH	Multi-Input 4	High speed
SPM	Multi-Input 5	Middle Speed
SPL	Multi-Input 6	Low Speed
DCM	Common Terminal of Digital and Control Signals, +12v Power, (EV、IPV、P24) Ground	
EV (IPV)	+12V Power Supply	Max. output current 200mA

P24	+12V Power Supply	Max. output current 200mA
+10	Power Supply for Speed Setting	+10V
VI	Analog Voltage Frequency Reference Input	0~+10V corresponding to the highest operating frequency
AI	Analog Current Frequency Reference Input	4~20mA corresponding to the highest operating frequency
ACM	Common Terminal of Analog and Control Signals	
DRV	Multi-Output 1 (Optical couple output)	DC24V/100mA
UPF	Multi-Output 2 (Optical couple output)	
FA (EFA)、 FB (EFB)、 FC (EFC)	Multi-Output 3 (N/O or N/C)	3A/250VAC
KA (EKA)、 KB (EKB)	Multi-Output 4 (N/O)	3A/250VAC
AM	Output terminals of digital frequency	0~10V
RS+ RS-	RS485 Communication port	

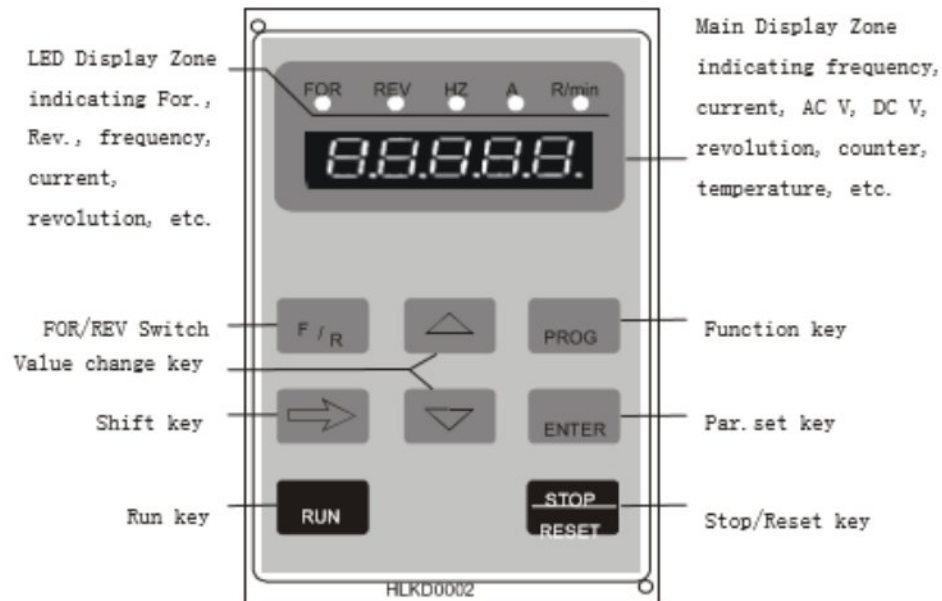
3. Basic Connection Diagram (图表请自行修改)

The wiring of the inverter is divided into two parts, main circuit terminal connections and control circuit terminal connections. The user can see the main circuit terminals, and the control circuit terminals after removing the cover of enclosure. The terminals must be connected correctly as the following wiring circuit diagrams.



VI. Instruction of the Digital Operator

1. Description of the Digital Operator (图标自行修改)



2. Description of Indicator Lamp Status

1) Description of Indicator Lamp Status

Indicator lamp	Status	Description
FOR	on	The motor is in forward rotation.
REV	on	The motor is in reverse rotation.
HZ	on	Displaying set frequency or output frequency.
A	on	Displaying output current.
ROTT	on	Displaying rated motor revolution
A ROTT	on	Displaying AC or DC voltage.
HZ ROTT	on	Displaying counting value.
HZ A ROTT	on	Displaying internal temperature of the inverter.

2) Description of Display Items

Display	Indic. lamp	on	Meaning
750.00	HZ A R/min	⊗ ○ ○	Present output frequency is 50.00HZ
F50.00	HZ A R/min	⊗ ○ ○	Present set frequency is 50.00HZ
R003.0	HZ A R/min	○ ⊗ ○	Present output current is 3.0A
01440	HZ A R/min	○ ○ ⊗	Present output revolution is 1440r/min
510.1	HZ A R/min	○ ⊗ ⊗	Present DC voltage is 510.1V
380.0	HZ A R/min	○ ⊗ ⊗	Present AC voltage is 380.0V
t035.0	HZ A R/min	⊗ ⊗ ⊗	Present inverter's temperature is 35.0°C
00105	HZ A R/min	⊗ ○ ⊗	Present counter's value is 105
50.0	HZ A R/min	○ ○ ○	Present target value of PID is 50.0%
48.0	HZ A R/min	○ ○ ○	Present feedback value of PID is 48.0%
00012	HZ A R/min	⊗ ⊗ ○	Present time of power-on is 12 hours
00108	HZ A R/min	○ ⊗ ⊗	Total run time of inverter is 108 hours



3. Description of Operation Examples

Procedures	Display	Indicator Lamp	Explanation
Power up, Operation of power ↓	Dsp2.0 flash → Vr2.00 → 000.00	FOR e HZ <input type="checkbox"/> <input type="checkbox"/>	Self detect when power-up, display version no. (Flashing) and finally set frequency.
PROG ↓	PD000	FOR e HZ <input type="checkbox"/> <input type="checkbox"/>	Enter programming Display the function of PD000
ENYER	000.00	FOR e HZ <input type="checkbox"/> <input type="checkbox"/>	Display the contents of PD000
▲ ↓ ENTER ↓ PROG	50.00 END → 50.00 PD001 050.00	FOR e HZ <input type="checkbox"/> <input type="checkbox"/> FOR e HZ <input type="checkbox"/> <input type="checkbox"/>	Change the content of PD000 Confirm changed value. Display END 50.00 PD001 Back from programming
↓ RUN	50.00	FOR HZ <input type="checkbox"/> <input type="checkbox"/>	Display running and operating frequency
↓ ← DISP	50.00 F 0.00 → 5 0.00	FOR HZ <input type="checkbox"/> <input type="checkbox"/>	Monitor screen switching, display output current
↓ PROG	005.0	FOR HZ <input type="checkbox"/> <input type="checkbox"/>	Monitor screen switching, display output current
↓ ← DISP	01440	FOR A <input type="checkbox"/> <input type="checkbox"/>	Monitor screen switching, display revolution
↓ PROG	50.00	FOR ROTT <input type="checkbox"/> <input type="checkbox"/>	Switch back to main screen, display set frequency
↓ F/R	50.00	FOR HZ <input type="checkbox"/> <input type="checkbox"/>	Switch of For.Rev. rotation, display the status of Rev rotation
↓ ▲	050.0	FOR HZ <input type="checkbox"/> <input type="checkbox"/>	Switch to adjustable frequency
← ↓ ▲	030.00	FOR HZ <input type="checkbox"/> <input type="checkbox"/>	Adjust set frequency, i.e. the value of PD000
↓ ENTER	030.00	FOR HZ <input type="checkbox"/> <input type="checkbox"/>	Confirm changed value, write to PD000 as value
↓ STOP	030.00	FOR e HZ <input type="checkbox"/> <input type="checkbox"/>	Stop

VII. Commissioning

1. Important Checks before the Commissioning

- If there is any wrong connected wires? Pay special attention to the terminal of U.V.W; Make sure the power supply wires are connected to R.S.T, not U.V.W.
- If there is any metal powder or wires left on the base plate of the inverter or the terminal block, which may cause short circuit.
- If screws are tightly locked and if the connecting parts are loose.
- If there is any short circuit or earth fault at outputs.

2. Commissioning Methods

The procedure of the operator is factory set up for the control mode of HY series. The commissioning can be carried out through the digital operator. Generally, the commissioning can be conducted at 5.00 Hz.

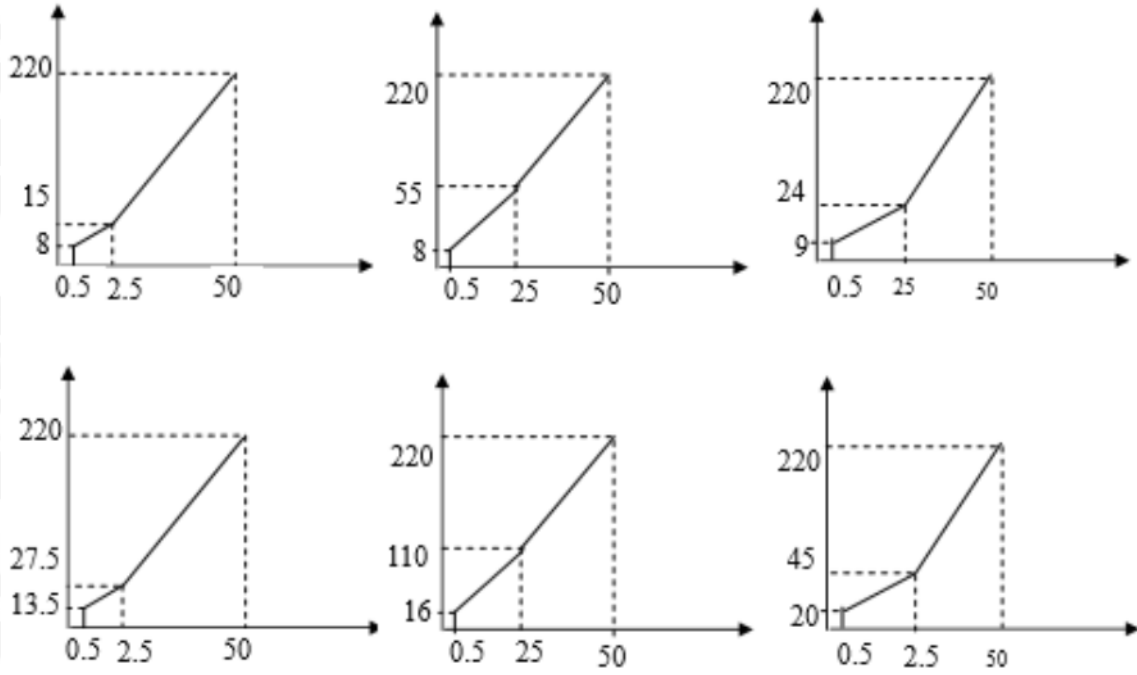
Procedures	Display	Indicator Lamp	Explanation
Power up ↓	dsp1.1→Vr2.0 × 000.0 ∧	FOR Hz ε □	Self detect when power up, display version no. and finally set frequency
△	000.00	FOR Hz ε □	Switch to adjustable frequency on the panel
↓ ←△	050.00	FOR Hz ε □	Change set frequency, i.e. the value of PD003
↓ ENTER	50.00	FOR Hz ε □	Confirm changed value
↓ RUN	50.00	FOR Hz □ □	Run at 50Hz
↓ STOP	50.00	FOR Hz ε □	Stop

Note: □ means indicator lamps is on; ε means indicator lamps flash;

. Descriptions of Functions

This parameter is set for the maximum operating frequency of the inverter.

The following are several curves and set values often used for reference. Specific curves must be set according to concrete characteristics of mechanical load.



Curve of constant torque

Curve of lower torque

Curve of higher torque



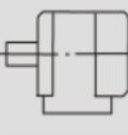



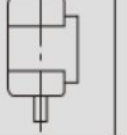
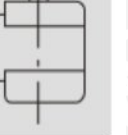
A-Y3A/A-Y3 SERIES MOTOR

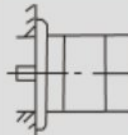
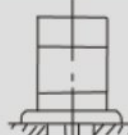
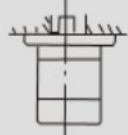
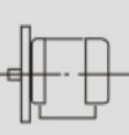
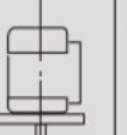
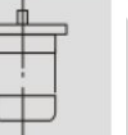


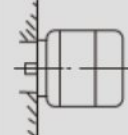
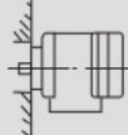
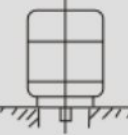
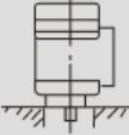
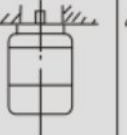
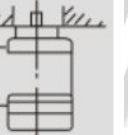
BEARING SIZE

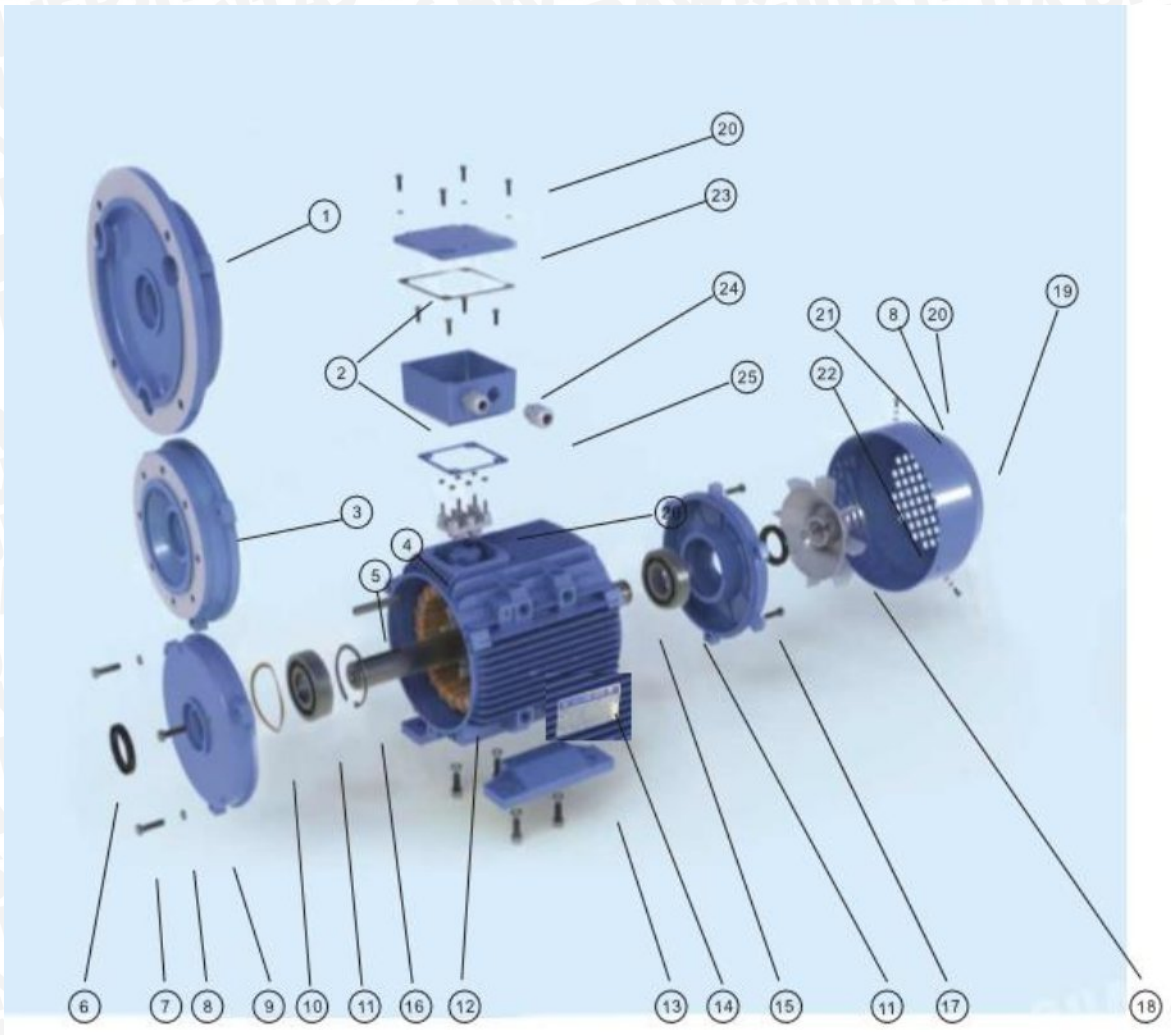
Frame Size	Poles	Drive End	Non-Drive End
		International type	International type
56	2~4	62012Z	62012Z
63	2~4	62012Z	62012Z
71	2~6	62022Z	62022Z
80	2~8	62042Z	62042Z
90	2~8	62052Z	62052Z
100	2~8	62062Z	62062Z
112	2~8	63062Z	63062Z
132	2~8	63082Z	63082Z
160	2~8	63092ZC3	63092ZC3
180	2~8	6311C3	6311C3
200	2~8	6312C3	6312C3
225	2~8	6313C3	6313C3
250	2~8	6314C3	6314C3
280	2	6314C3	6314C3
	4~8	6317C3	6317C3
315	2	6317C3	6317C3
	4~10	NU319C3	6319C3
355	2	6319C3	6319C3
	4~10	NU322C3	6322C3
400	4~10	NU326C3	6326C3

The mounting arrangements of the motors comply with IEC34-7 Recommendation. There are four basic arrangements shown as the following tables and figures.

Fundamental arrangement	B3					
Mounting arrangement	B3	B6	B7	B8	V5	V6
Diagram						
Range of Manufacture (framesize)	56-400	56-160				

Fundamental arrangement	B5			B35		
Mounting arrangement	B5	V1	V3	B35	V15	V36
Diagram						
Range of Manufacture (framesize)	56-280	56-355	56-160	56-400	56-160	

Fundamental arrangement	B14					
Mounting arrangement	B14	B34	V18	V58	V19	V69
Diagram						
Range of Manufacture (framesize)	56-132					



- | | | | |
|----------------------|--------------------|--------------------|-----------------------|
| 1. B5 Flange | 8. Spring Washer | 15. Rotor | 22. Fan clamp |
| 2. Gasket | 9. Front Endshield | 16. Circlip | 23. Terminal box lid |
| 3. B14 Flange | 10. Wave washer | 17. Rear Endshield | 24. Terminal box base |
| 4. Frame | 11. Bearing | 18. Fan | 25. Cable gland |
| 5. Key | 12. Stator | 19. Fan cowl | 26. Terminal board |
| 6. Oil seal (V ring) | 13. Feet | 20. Screw | |
| 7. Bolt | 14. Nameplate | 21. Washer | |

MAIN DATA FOR TERMINAL BOX

Classified number	Frame size	Max.F.Amps	Entry hole size
			International standard
1	H56-80	2.6	2×M20×1.5
2	H90-100	6.8	2×M25×1.5
3	H112-132	15.4	2×M32×1.5
4	H160-180	42.5	2×M40×1.5
5	H200-225	84.2	2×M50×1.5
6	H250-280	166.6	2×M63×1.5
7	H315	358	2×M63×1.5
8	H355	546	2×M63×1.5
9	H400	600	3XM63X1.5

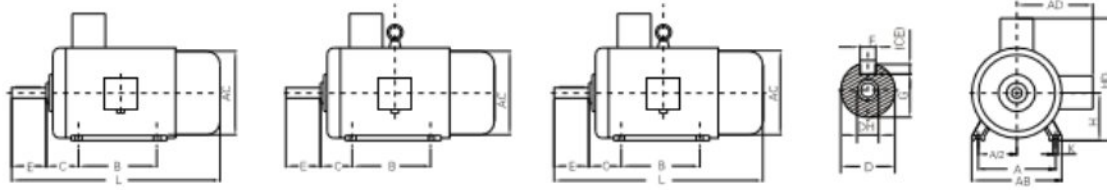
TECHNICAL DATA OF A-Y3 SERIES

NO.	Frame reference and size	Full load current at rated voltage			Rated power		Full load speed in revolutions per minute	Efficiency	Power factor	Direct on line starting torque ratio			Noise level @ 1m on no load	Weight	Rotor inertia Wg ²
		380V	400V	415V	Power kW	HP				LRT	LRA	BDT			
	Type						Speed r/min	EFF. %	P.F. Cosφ	RLT	RLA	RLT	Noise LwdB(A)	Weight kg	J kgm2
1	A-Y3-80M1-2	1.77	1.74	1.68	0.75	1	2840	75.0	0.83	2.2	6.1	2.3	67	16	0.00075
2	A-Y3-80M2-2	2.61	2.48	2.39	1.1	1.5	2840	76.2	0.84	2.2	6.9	2.3	67	17	0.00090
3	A-Y3-90S-2	3.46	3.28	3.16	1.5	2	2850	78.5	0.84	2.2	7.0	2.3	72	20	0.00120
4	A-Y3-90L-2	4.85	4.61	4.45	2.2	3	2855	81.0	0.85	2.2	7.0	2.3	72	23	0.00140
5	A-Y3-100L-2	6.34	6.03	5.81	3	4	2860	82.6	0.87	2.2	7.5	2.3	76	30	0.00290
6	A-Y3-112M-2	8.2	7.79	7.51	4	5.5	2880	84.2	0.88	2.2	7.5	2.3	77	41	0.00550
7	A-Y3-132S1-2	11.1	10.53	10.15	5.5	7.5	2900	85.7	0.88	2.2	7.5	2.3	80	57.5	0.01090
8	A-Y3-132S2-2	14.9	14.1	13.6	7.5	10	2900	87.0	0.88	2.2	7.5	2.3	80	60.5	0.01260
9	A-Y3-160M1-2	21.2	20.2	19.5	11	15	2930	88.4	0.89	2.2	7.5	2.3	86	107	0.03770
10	A-Y3-160M2-2	28.6	27.2	26.2	15	20	2930	89.4	0.89	2.2	7.5	2.3	86	114	0.04990
11	A-Y3-160L-2	34.7	33.0	31.8	18.5	25	2930	90.0	0.90	2.2	7.5	2.3	86	133	0.05500
12	A-Y3-180M-2	41	39.0	37.6	22	30	2940	90.5	0.90	2.0	7.5	2.3	89	165	0.07500
13	A-Y3-200L1-2	55.4	52.6	50.7	30	40	2950	91.4	0.90	2.0	7.5	2.3	92	218	0.12400
14	A-Y3-200L2-2	67.9	64.5	62.2	37	50	2950	92.0	0.90	2.0	7.5	2.3	92	230	0.13900
15	A-Y3-225M-2	82.1	78.0	75.2	45	60	2960	92.5	0.90	2.0	7.5	2.3	92	290	0.23300

16	A-Y3-250M-2	100	94.8	91.4	55	75	2970	93.0	0.90	2.0	7.5	2.3	93	359	0.31200
17	A-Y3-280S-2	135	129	124	75	100	2975	93.6	0.90	2.0	7.0	2.3	94	475	0.57900
18	A-Y3-280M-2	160	152	147	90	125	2975	93.9	0.91	2.0	7.1	2.3	94	510	0.67500
19	A-Y3-315S-2	195	186	179	110	150	2975	94.0	0.91	1.8	7.1	2.2	96	875	1.18000
20	A-Y3-315M-2	233	222	214	132	180	2975	94.5	0.91	1.8	7.1	2.2	96	963	1.82000
21	A-Y3-315L1-2	279	265	256	160	220	2975	94.6	0.92	1.8	7.1	2.2	99	1010	2.08000
22	A-Y3-315L2-2	348	331	319	200	270	2975	94.8	0.92	1.8	7.1	2.2	99	1138	2.38000
23	A-Y3-355M-2	433	412	397	250	340	2980	95.2	0.92	1.6	7.1	2.2	103	1900	3.00000
24	A-Y3-355L-2	545	518	499	315	430	2980	95.4	0.92	1.6	7.1	2.2	103	2300	3.50000



MOUNTING DATA FOR A-Y3



H80-90

H100-132

H160-355

H80-355

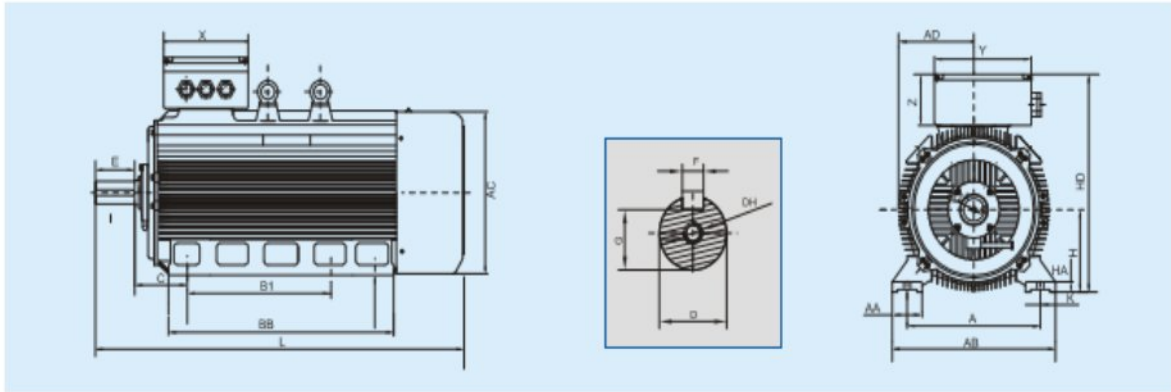
Frame with feet and end-shield without flange(IM B3)

FRAME SIZE	POLES	A	A/2	B	C	D	E	F	G	H	K	AB	AC	AD	HD	L	DH*
80M	2 4 6 8	125	62.5	100	50	19	40	6	15.5	80	10	165	155	145	220	295	M6×16
90S	2 4 6 8	140	70	100	56	24	50	8	20	90	10	180	175	155	250	320	M8×19
90L	2 4 6 8	140	70	125	56	24	50	8	20	90	10	180	175	155	250	345	M8×19
100L	2 4 6 8	160	80	140	63	28	60	8	24	100	12	205	196	180	270	385	M10×22
112M	2 4 6 8	190	95	140	70	28	60	8	24	112	12	230	220	190	300	400	M10×22
132S	2 4 6 8	216	108	140	89	38	80	10	33	132	12	270	259	210	345	470	M12×28
132M	2 4 6 8	216	108	178	89	38	80	10	33	132	12	270	259	210	345	510	M12×28
160M	2 4 6 8	254	127	210	108	42	110	12	37	160	15	320	315	255	420	615	M16×36
160L	2 4 6 8	254	127	254	108	42	110	12	37	160	15	320	315	255	420	660	M16×36
180M	2 4 6 8	279	139.5	241	121	48	110	14	42.5	180	15	355	355	280	455	700	M16×36
180L	2 4 6 8	279	139.5	279	121	48	110	14	42.5	180	15	355	355	280	455	740	M16×36
200L	2 4 6 8	318	159	305	133	55	110	16	49	200	19	395	397	305	505	770	M20×42
225S	4 8	356	178	286	149	60	140	18	53	225	19	435	445	335	560	815	M20×42
225M	2	356	178	311	149	55	110	16	49	225	19	435	445	335	560	820	M20×42
	4 6 8	356	178	311	149	60	140	18	53	225	19	435	445	335	560	845	M20×42
250M	2	406	203	349	168	60	140	18	53	250	24	490	485	370	615	920	M20×42
	4 6 8	406	203	349	168	65	140	18	58	250	24	490	485	370	615	920	M20×42
280S	2	457	228.5	368	190	65	140	18	58	280	24	550	547	410	680	995	M20×42
	4 6 8	457	228.5	368	190	75	140	20	67.5	280	24	550	547	410	680	995	M20×42
280M	2	457	228.5	419	190	65	140	18	58	280	24	550	547	410	680	1045	M20×42
	4 6 8	457	228.5	419	190	75	140	20	67.5	280	24	550	547	410	680	1045	M20×42
315S	2	508	254	406	216	65	140	18	58	315	28	635	620	530	845	1185	M20×42
	4 6 8 10	508	254	406	216	80	170	22	71	315	28	635	620	530	845	1220	M20×42
315M	2	508	254	457	216	65	140	18	58	315	28	635	620	530	845	1290	M20×42
	4 6 8 10	508	254	457	216	80	170	22	71	315	28	635	620	530	845	1325	M20×42
315L	2	508	254	508	216	65	140	18	58	315	28	635	620	530	845	1290	M20×42
	4 6 8 10	508	254	508	216	80	170	22	71	315	28	635	620	530	845	1325	M20×42
355M	2	610	305	560	254	75	140	20	67.5	355	28	730	698	655	1010	1500	M20×42
	4 6 8 10	610	305	560	254	95	170	25	86	355	28	730	698	655	1010	1530	M20×42
355L	2	610	305	630	254	75	140	20	67.5	355	28	730	698	655	1010	1500	M20×42
	4 6 8 10	610	305	630	254	95	170	25	86	355	28	730	698	655	1010	1530	M20×42



IM B3 H400-450

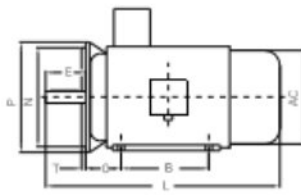
MOUNTING AND OVERALL DIMENSIONS OF IM B3 H400-450



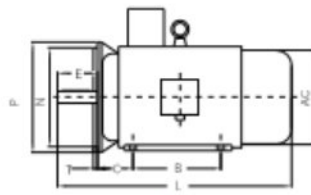
Type	Mounting Dimensions										Overall Dimensions													
	Poles	A	AA	AB	AC	B1	BB	C	D	DH	E	F	G	H	HA	HD	K	L	AD	Eyeboit	X	Y	Z	
400L	4								±110		210	28	100				1925							
400L	6.8.10	686	125	810	855	710	1090	280	±120	M24X54	210	32	109	400	30	1080	±36	1925	430	2xM36	430	540	225	
450L	4								±130		210	32	119				2200							
450L	4.6.8.10	800	190	1000	930	1000	1300	300	±140	M24X54	210	32	129	450	52	1380	±42	2200	480	2xM36	500	595	410	



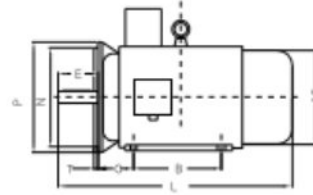
MOUNTING DATA FOR A-Y3



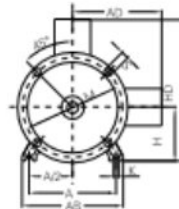
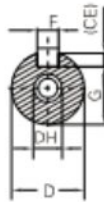
H80-90



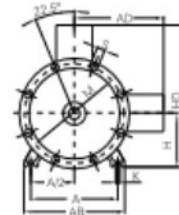
H100-132



H160-355



H80-200



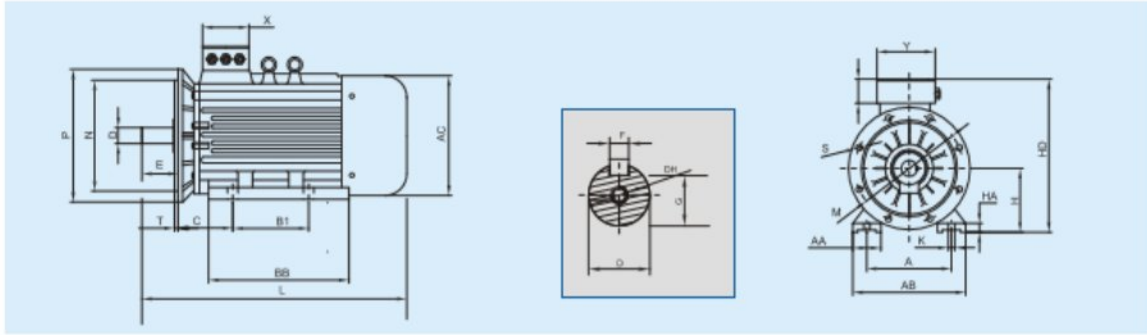
H225-355

FRAME WITH FEET AND END-SHIELD WITH FLANGE(IM B35)

FRAME SIZE	POLES	A	A/2	B	C	D	E	F	G	H	K	M	N	P	S	T	FLANGE HOLES	AB	AC	AD	HD	L	DH*
80M	2 4 6 8	125	62.5	100	50	19	40	6	15.5	80	10	165	130	200	12	3.5	4	165	155	145	220	295	M6 x 16
90S	2 4 6 8	140	70	100	56	24	50	8	20	90	10	165	130	200	12	3.5	4	180	175	155	250	320	M8 x 19
90L	2 4 6 8	140	70	125	56	24	50	8	20	90	10	165	130	200	12	3.5	4	180	175	155	250	345	M8 x 19
100L	2 4 6 8	160	80	140	63	28	60	8	24	100	12	215	180	250	15	4	4	205	196	180	270	385	M10x22
112M	2 4 6 8	190	95	140	70	28	60	8	24	112	12	215	180	250	15	4	4	230	220	190	300	400	M10x22
132S	2 4 6 8	216	108	140	89	38	80	10	33	132	12	265	230	300	15	4	4	270	259	210	345	470	M12x28
132M	2 4 6 8	216	108	178	89	38	80	10	33	132	12	265	230	300	15	4	4	270	259	210	345	510	M12x28
160M	2 4 6 8	254	127	210	108	42	110	12	37	160	15	300	250	350	19	5	4	320	315	255	420	615	M16x36
160L	2 4 6 8	254	127	254	108	42	110	12	37	160	15	300	250	350	19	5	4	320	315	255	420	660	M16x36
180M	2 4 6 8	279	139.5	241	121	48	110	14	42.5	180	15	300	250	350	19	5	4	355	355	280	455	700	M16x36
180L	2 4 6 8	279	139.5	279	121	48	110	14	42.5	180	15	300	250	350	19	5	4	355	355	280	455	740	M16x36
200L	2 4 6 8	318	159	305	133	55	110	16	49	200	19	350	300	400	19	5	4	395	397	305	505	770	M20x42
225S	4 8	356	178	286	149	60	140	18	53	225	19	400	350	450	19	5	8	435	445	335	560	815	M20x42
225M	2	356	178	311	149	55	110	16	49	225	19	400	350	450	19	5	8	435	445	335	560	820	M20x42
	4 6 8	356	178	311	149	60	140	18	53	225	19	400	350	450	19	5	8	435	445	335	560	845	M20x42
250M	2	406	203	349	168	60	140	18	53	250	24	500	450	550	19	5	8	490	485	370	615	920	M20x42
	4 6 8	406	203	349	168	65	140	18	58	250	24	500	450	550	19	5	8	490	485	370	615	920	M20x42
280S	2	457	228.5	368	190	65	140	18	58	280	24	500	450	550	19	5	8	550	547	410	680	995	M20x42
	4 6 8	457	228.5	368	190	75	140	20	67.5	280	24	500	450	550	19	5	8	550	547	410	680	995	M20x42
280M	2	457	228.5	419	190	65	140	18	58	280	24	500	450	550	19	5	8	550	547	410	680	1045	M20x42
	4 6 8	457	228.5	419	190	75	140	20	67.5	280	24	500	450	550	19	5	8	550	547	410	680	1045	M20x42
315S	2	508	254	406	216	65	140	18	58	315	28	600	550	660	24	6	8	635	620	530	845	1185	M20x42
	4 6 8 10	508	254	406	216	80	170	22	71	315	28	600	550	660	24	6	8	635	620	530	845	1220	M20x42
315M	2	508	254	457	216	65	140	18	58	315	28	600	550	660	24	6	8	635	620	530	845	1290	M20x42
	4 6 8 10	508	254	457	216	80	170	22	71	315	28	600	550	660	24	6	8	635	620	530	845	1325	M20x42
315L	2	508	254	508	216	65	140	18	58	315	28	600	550	660	24	6	8	635	620	530	845	1290	M20x42
	4 6 8 10	508	254	508	216	80	170	22	71	315	28	600	550	660	24	6	8	635	620	530	845	1325	M20x42
355M	2	610	305	560	254	75	140	20	67.5	355	28	740	680	800	24	6	8	730	698	655	1010	1500	M20x42
	4 6 8 10	610	305	560	254	95	170	25	86	355	28	740	680	800	24	6	8	730	698	655	1010	1530	M20x42
355L	2	610	305	630	254	75	140	20	67.5	355	28	740	680	800	24	6	8	730	698	655	1010	1500	M20x42
	4 6 8 10	610	305	630	254	95	170	25	86	355	28	740	680	800	24	6	8	730	698	655	1010	1530	M20x42

IM B35 H400-450

MOUNTING AND OVERALL DIMENSIONS OF IM B35 H400-450

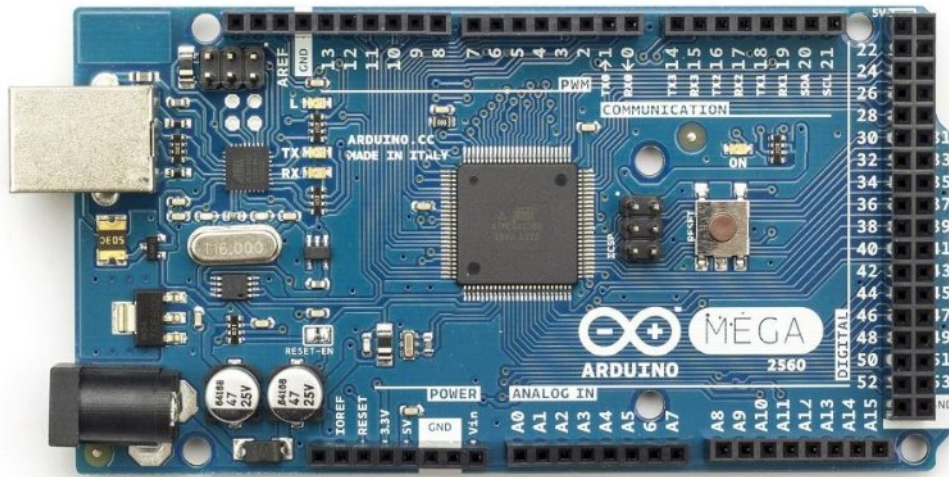


Type	Poles	Mounting Dimensions													Overall Dimensions												
		A	AA	AB	AC	B1	BB	C	D	E	F	H	HA	HD	DH	K	L	M	N	P	S	T	Eyeboit	X	Y	Z	
400L	4								110	210	28					1925											
400L	6/8/10	686	125	810	855	710	1090	280	120	210	32	400	30	1080	M24X54	36	1925	940	880	1000	8xφ28	6	2xM36	430	540	225	
450L	4	800	190	1000	930	1000	1300	300	130	210	32	450	52	1380		42	2200	1080	1000	1150	8xφ28	6	2xM36	500	595	410	
450L	6/8/10								140	210	32				M24X54		2200										



Arduino Mega2560





Technical specs

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
Length	101.52 mm
Width	53.3 mm
Weight	37 g

Programming

The Mega 2560 board can be programmed with the [Arduino Software \(IDE\)](#). For details, see the [reference](#) and [tutorials](#).

The ATmega2560 on the Mega 2560 comes preprogrammed with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header using [Arduino ISP](#) or similar; see these [instructions for details](#).

The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available in the [Arduino repository](#). The ATmega16U2/8U2 is loaded with a DFU bootloader, which can be activated by:

- On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.
- On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode. You can then use [Atmel's FLIP software](#) (Windows) or the [DFU programmer](#) (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See [this user-contributed tutorial](#) for more information.

Power

The Mega 2560 can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the GND and Vin pin headers of the POWER connector.

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

The power pins are as follows:

- Vin. The input voltage to the board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- 5V. This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.
- 3V3. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- GND. Ground pins.
- IOREF. This pin on the board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs for working with the 5V or 3.3V.

Memory

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

Input and Output

See the mapping between Arduino pins and Atmega2560 ports:



Each of the 54 digital pins on the Mega can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive 20 mA as recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50 k ohm. A maximum of 40mA is the value that must not be exceeded to avoid permanent damage to the microcontroller.

In addition, some pins have specialized functions:

- Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX). Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega16U2 USB-to-TTL Serial chip.
- External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2). These pins can be configured to trigger an interrupt on a low level, a rising or falling edge, or a change in level. See the [attachInterrupt\(\)](#) function for details.
- PWM: 2 to 13 and 44 to 46. Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). These pins support SPI communication using the [SPI library](#). The SPI pins are also broken out on the ICSP header, which is physically compatible with the Arduino /Genuino Uno and the old Duemilanove and Diecimila Arduino boards.
- LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- TWI: 20 (SDA) and 21 (SCL). Support TWI communication using the [Wire library](#). Note that these pins are not in the same location as the TWI pins on the old Duemilanove or Diecimila Arduino boards.

See also the mapping Arduino Mega 2560 PIN diagram.

The Mega 2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and [analogReference\(\)](#) function.

There are a couple of other pins on the board:

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

Communication

The Mega 2560 board has a number of facilities for communicating with a computer, another board, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega16U2 (ATmega 8U2 on the revision 1 and revision 2 boards) on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically). The Arduino Software (IDE) includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2/ATmega16U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

Physical Characteristics and Shield Compatibility

The maximum length and width of the Mega 2560 PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

The Mega 2560 is designed to be compatible with most shields designed for the Uno and the older Diecimila or Duemilanove Arduino boards. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Furthermore, the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega 2560 and Duemilanove / Diecimila boards. Please note that I2C is not located on the same pins on the Mega 2560 board (20 and 21) as the Duemilanove / Diecimila boards (analog inputs 4 and 5).

Revisions

The Mega 2560 does not use the FTDI USB-to-serial driver chip used in past designs. Instead, it features the ATmega16U2 (ATmega8U2 in the revision 1 and revision 2 Arduino boards) programmed as a USB-to-serial converter.

Revision 2 of the Mega 2560 board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into [DFU mode](#).

Revision 3 of the Arduino board and the current Genuino Mega 2560 have the following improved features:

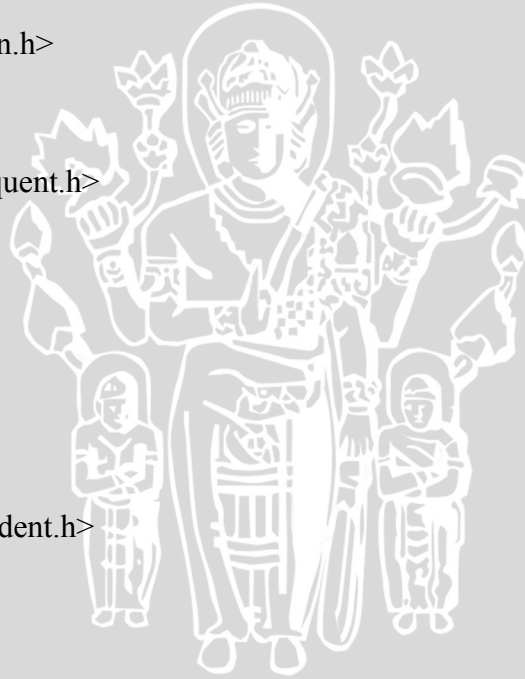
- 1.0 pinout: SDA and SCL pins - near to the AREF pin - and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that use the AVR, which operate with 5V and with the board that uses ATSAM3X8E, that operate with 3.3V. The second one is a not connected pin, that is reserved for future purposes.
- Stronger RESET circuit.
- Atmega 16U2 replace the 8U2.

Listing Program

```
/*  
// NAMA : Azwan Mahadin Kusuma  
// NIM : 105060307111028  
// JUDUL : Sistem Pengendalian Kecepatan Aliran Udara Pada Wind Tunnel  
Dengan Umpan Balik Kecepatan Aliran Udara Menggunakan Kontrol Logika Fuzzy
```

```
/*  
#include <FuzzyRule.h>  
#include <FuzzyComposition.h>  
#include <Fuzzy.h>  
#include <FuzzyRuleConsequent.h>  
#include <FuzzyOutput.h>  
#include <FuzzyInput.h>  
#include <FuzzyIO.h>  
#include <FuzzySet.h>  
#include <FuzzyRuleAntecedent.h>
```

```
int AnFback = A0;  
int AnSpoint = A1;  
int outPWM = 13;  
int d, spoint, flow, e, e_old, ce;  
  
// besarnya flow pada vout = 5volt  
int const fullFlow = 10;
```




```
// class fuzzy
Fuzzy* fuzzy = new Fuzzy();

// fuzzysset error
FuzzySet* e_mn = new FuzzySet(0, 0, 2, 4);
FuzzySet* e_n = new FuzzySet(2, 4, 4, 6);
FuzzySet* e_z = new FuzzySet(4, 6, 6, 8);
FuzzySet* e_p = new FuzzySet(6, 8, 8, 9);
FuzzySet* e_mp = new FuzzySet(8, 9, 9.5, 9.5);

// fuzzysset change error
FuzzySet* ce_mn = new FuzzySet(0, 0, 0.2, 0.5);
FuzzySet* ce_n = new FuzzySet(0.2, 0.5, 0.5, 0.7);
FuzzySet* ce_z = new FuzzySet(0.5, 0.7, 0.7, 0.8);
FuzzySet* ce_p = new FuzzySet(0.7, 0.8, 0.8, 0.9);
FuzzySet* ce_mp = new FuzzySet(0.8, 0.9, 0.95, 0.95);

// fuzzysset singleton output
FuzzySet* o_mn = new FuzzySet(50, 50, 50, 50);
FuzzySet* o_n = new FuzzySet(100, 100, 100, 100);
FuzzySet* o_z = new FuzzySet(150, 150, 150, 150);
FuzzySet* o_p = new FuzzySet(200, 200, 200, 200);
FuzzySet* o_mp = new FuzzySet(255, 255, 255, 255);
```

```
int ReadFlow()
{
```



```
d = analogRead(AnFback);  
flow = (d/1023)*fullFlow;  
return(flow);  
}
```

```
int ReadSpoint()  
{
```

```
    d = analogRead(AnSpoint);  
    spoint = (d/1023)*fullFlow;  
    return(spoint);  
}
```

```
void setup(){
```

```
    e = 0; ce = 0;
```

```
    // fuzzy input error
```

```
    FuzzyInput* error = new FuzzyInput(1);
```

```
    error->addFuzzySet(e_mn);
```

```
    error->addFuzzySet(e_n);
```

```
    error->addFuzzySet(e_z);
```

```
    error->addFuzzySet(e_p);
```

```
    error->addFuzzySet(e_mp);
```

```
    fuzzy->addFuzzyInput(error);
```

```
    // fuzzy input change error
```

```
    FuzzyInput* cerror = new FuzzyInput(2);
```

```
    cerror->addFuzzySet(ce_mn);
```

UNIVERSITAS BRAWIJAYA



```
cerror->addFuzzySet(ce_n);  
cerror->addFuzzySet(ce_z);  
cerror->addFuzzySet(ce_p);  
cerror->addFuzzySet(ce_mp);  
fuzzy->addFuzzyInput(cerror);
```

```
// fuzzy output
```

```
FuzzyOutput* output = new FuzzyOutput(1);
```

```
output->addFuzzySet(o_mn);
```

```
output->addFuzzySet(o_n);
```

```
output->addFuzzySet(o_z);
```

```
output->addFuzzySet(o_p);
```

```
output->addFuzzySet(o_mp);
```

```
fuzzy->addFuzzyOutput(output);
```

```
// consequent
```

```
FuzzyRuleConsequent* thenMN = new FuzzyRuleConsequent();
```

```
thenMN->addOutput(o_mn);
```

```
FuzzyRuleConsequent* thenN = new FuzzyRuleConsequent();
```

```
thenN->addOutput(o_n);
```

```
FuzzyRuleConsequent* thenZ = new FuzzyRuleConsequent();
```

```
thenZ->addOutput(o_z);
```

```
FuzzyRuleConsequent* thenP = new FuzzyRuleConsequent();
```

```
thenP->addOutput(o_p);
```

```
FuzzyRuleConsequent* thenMP = new FuzzyRuleConsequent();
```

```
thenMP->addOutput(o_mp);
```



```
// antecedent baris 1
```

```
FuzzyRuleAntecedent* antecedent1_1 = new FuzzyRuleAntecedent();
```

```
antecedent1_1->joinWithAND(e_mn, ce_mn);
```

```
FuzzyRuleAntecedent* antecedent1_2 = new FuzzyRuleAntecedent();
```

```
antecedent1_2->joinWithAND(e_n, ce_mn);
```

```
FuzzyRuleAntecedent* antecedent1_3 = new FuzzyRuleAntecedent();
```

```
antecedent1_3->joinWithAND(e_z, ce_mn);
```

```
FuzzyRuleAntecedent* antecedent1_4 = new FuzzyRuleAntecedent();
```

```
antecedent1_4->joinWithAND(e_p, ce_mn);
```

```
FuzzyRuleAntecedent* antecedent1_5 = new FuzzyRuleAntecedent();
```

```
antecedent1_5->joinWithAND(e_mp, ce_mn);
```

```
// antecedent baris 2
```

```
FuzzyRuleAntecedent* antecedent2_1 = new FuzzyRuleAntecedent();
```

```
antecedent2_1->joinWithAND(e_mn, ce_n);
```

```
FuzzyRuleAntecedent* antecedent2_2 = new FuzzyRuleAntecedent();
```

```
antecedent2_2->joinWithAND(e_n, ce_n);
```

```
FuzzyRuleAntecedent* antecedent2_3 = new FuzzyRuleAntecedent();
```

```
antecedent2_3->joinWithAND(e_z, ce_n);
```

```
FuzzyRuleAntecedent* antecedent2_4 = new FuzzyRuleAntecedent();
```

```
antecedent2_4->joinWithAND(e_p, ce_n);
```

```
FuzzyRuleAntecedent* antecedent2_5 = new FuzzyRuleAntecedent();
```

```
antecedent2_5->joinWithAND(e_mp, ce_n);
```

```
// antecedent baris 3
```

```
FuzzyRuleAntecedent* antecedent3_1 = new FuzzyRuleAntecedent();
```

```
antecedent3_1->joinWithAND(e_mn, ce_z);
```

```
FuzzyRuleAntecedent* antecedent3_2 = new FuzzyRuleAntecedent();
```

```
antecedent3_2->joinWithAND(e_n, ce_z);
```



```
FuzzyRuleAntecedent* antecedent3_3 = new FuzzyRuleAntecedent();
antecedent3_3->joinWithAND(e_z, ce_z);

FuzzyRuleAntecedent* antecedent3_4 = new FuzzyRuleAntecedent();
antecedent3_4->joinWithAND(e_p, ce_z);

FuzzyRuleAntecedent* antecedent3_5 = new FuzzyRuleAntecedent();
antecedent3_5->joinWithAND(e_mp, ce_z);

// antecedent baris 4

FuzzyRuleAntecedent* antecedent4_1 = new FuzzyRuleAntecedent();
antecedent4_1->joinWithAND(e_mn, ce_p);

FuzzyRuleAntecedent* antecedent4_2 = new FuzzyRuleAntecedent();
antecedent4_2->joinWithAND(e_n, ce_p);

FuzzyRuleAntecedent* antecedent4_3 = new FuzzyRuleAntecedent();
antecedent4_3->joinWithAND(e_z, ce_p);

FuzzyRuleAntecedent* antecedent4_4 = new FuzzyRuleAntecedent();
antecedent4_4->joinWithAND(e_p, ce_p);

FuzzyRuleAntecedent* antecedent4_5 = new FuzzyRuleAntecedent();
antecedent4_5->joinWithAND(e_mp, ce_p);

// antecedent baris 5

FuzzyRuleAntecedent* antecedent5_1 = new FuzzyRuleAntecedent();
antecedent5_1->joinWithAND(e_mn, ce_mp);

FuzzyRuleAntecedent* antecedent5_2 = new FuzzyRuleAntecedent();
antecedent5_2->joinWithAND(e_n, ce_mp);

FuzzyRuleAntecedent* antecedent5_3 = new FuzzyRuleAntecedent();
antecedent5_3->joinWithAND(e_z, ce_mp);

FuzzyRuleAntecedent* antecedent5_4 = new FuzzyRuleAntecedent();
antecedent5_4->joinWithAND(e_p, ce_mp);

FuzzyRuleAntecedent* antecedent5_5 = new FuzzyRuleAntecedent();
```

```
antecedent5_5->joinWithAND(e_mp, ce_mp);
```

```
// rule
```

```
FuzzyRule* rule1_1 = new FuzzyRule(1, antecedent1_1, thenMP);
```

```
fuzzy->addFuzzyRule(rule1_1);
```

```
FuzzyRule* rule1_2 = new FuzzyRule(2, antecedent1_2, thenMP);
```

```
fuzzy->addFuzzyRule(rule1_2);
```

```
FuzzyRule* rule1_3 = new FuzzyRule(3, antecedent1_3, thenMP);
```

```
fuzzy->addFuzzyRule(rule1_3);
```

```
FuzzyRule* rule1_4 = new FuzzyRule(4, antecedent1_4, thenP);
```

```
fuzzy->addFuzzyRule(rule1_4);
```

```
FuzzyRule* rule1_5 = new FuzzyRule(5, antecedent1_5, thenZ);
```

```
fuzzy->addFuzzyRule(rule1_5);
```

```
FuzzyRule* rule2_1 = new FuzzyRule(6, antecedent2_1, thenMP);
```

```
fuzzy->addFuzzyRule(rule2_1);
```

```
FuzzyRule* rule2_2 = new FuzzyRule(7, antecedent2_2, thenMP);
```

```
fuzzy->addFuzzyRule(rule2_2);
```

```
FuzzyRule* rule2_3 = new FuzzyRule(8, antecedent2_3, thenP);
```

```
fuzzy->addFuzzyRule(rule2_3);
```

```
FuzzyRule* rule2_4 = new FuzzyRule(9, antecedent2_4, thenZ);
```

```
fuzzy->addFuzzyRule(rule2_4);
```

```
FuzzyRule* rule2_5 = new FuzzyRule(10, antecedent2_5, thenN);
```

```
fuzzy->addFuzzyRule(rule2_5);
```

```
FuzzyRule* rule3_1 = new FuzzyRule(11, antecedent3_1, thenMP);
```

```
fuzzy->addFuzzyRule(rule3_1);  
FuzzyRule* rule3_2 = new FuzzyRule(12, antecedent3_2, thenP);  
fuzzy->addFuzzyRule(rule3_2);  
FuzzyRule* rule3_3 = new FuzzyRule(13, antecedent3_3, thenZ);  
fuzzy->addFuzzyRule(rule3_3);  
FuzzyRule* rule3_4 = new FuzzyRule(14, antecedent3_4, thenN);  
fuzzy->addFuzzyRule(rule3_4);  
FuzzyRule* rule3_5 = new FuzzyRule(15, antecedent3_5, thenMN);  
fuzzy->addFuzzyRule(rule3_5);  
  
FuzzyRule* rule4_1 = new FuzzyRule(16, antecedent4_1, thenP);  
fuzzy->addFuzzyRule(rule4_1);  
FuzzyRule* rule4_2 = new FuzzyRule(17, antecedent4_2, thenZ);  
fuzzy->addFuzzyRule(rule4_2);  
FuzzyRule* rule4_3 = new FuzzyRule(18, antecedent4_3, thenN);  
fuzzy->addFuzzyRule(rule4_3);  
FuzzyRule* rule4_4 = new FuzzyRule(19, antecedent4_4, thenMN);  
fuzzy->addFuzzyRule(rule4_4);  
FuzzyRule* rule4_5 = new FuzzyRule(20, antecedent4_5, thenMN);  
fuzzy->addFuzzyRule(rule4_5);  
  
FuzzyRule* rule5_1 = new FuzzyRule(21, antecedent5_1, thenZ);  
fuzzy->addFuzzyRule(rule5_1);  
FuzzyRule* rule5_2 = new FuzzyRule(22, antecedent5_2, thenN);  
fuzzy->addFuzzyRule(rule5_2);  
FuzzyRule* rule5_3 = new FuzzyRule(23, antecedent5_3, thenMN);  
fuzzy->addFuzzyRule(rule5_3);
```



```
FuzzyRule* rule5_4 = new FuzzyRule(24, antecedent5_4, thenMN);  
fuzzy->addFuzzyRule(rule5_4);  
FuzzyRule* rule5_5 = new FuzzyRule(25, antecedent5_5, thenMN);  
fuzzy->addFuzzyRule(rule5_5);  
}
```

```
void loop(){  
    e = ReadSpoint()-ReadFlow();  
    ce = e - e_old;  
    fuzzy->setInput(1, e);  
    fuzzy->setInput(2, ce);  
    fuzzy->fuzzify();  
    float output = fuzzy->defuzzify(1);  
    analogWrite(outPWM, output);  
    e_old = e;  
    delay(100);  
}
```

