

repository.ub.ac.id

**PENGENDALIAN SUHU DAN SALINITAS AIR PADA AKUARIUM
IKAN BADUT (*Amphiprion percula*) BERBASIS MIKROKONTROLER
ARDUINO DUE**

**SKRIPSI
TEKNIK ELEKTRO KONSENTRASI TEKNIK KONTROL**

Ditujukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik



**IKHFAL RUHYADI
NIM 125060307111019**

**UNIVERSITAS BRAWIJAYA
FAKULTAS TEKNIK
MALANG**

2016

LEMBAR PENGESAHAN

PENGENDALIAN SUHU DAN SALINITAS AIR PADA AKUARIUM IKAN BADUT (*Amphiprion percula*) BERBASIS MIKROKONTROLER ARDUINO DUE

SKRIPSI

TEKNIK ELEKTRO KONSENTRASI TEKNIK KONTROL

Ditujukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik



IKHFAL RUHYADI
NIM. 125060307111019

Skripsi ini telah direvisi dan disetujui oleh dosen pembimbing
Pada tanggal 12 Agustus 2016

Dosen Pembimbing I

Dosen Pembimbing II

Ir. Purwanto, M.T.
NIP. 19540424 198601 1 001

Goegoes Dwi Nusantoro, S.T., M.T.
NIP. 19711013 200604 1 001

Mengetahui,
Ketua Jurusan Teknik Elektro

M. Aziz Muslim, S.T., M.T., Ph.D.
NIP. 19741203 200012 1 001

JUDUL SKRIPSI :

*PENGENDALIAN SUHU DAN SALINITAS AIR PADA AKUARIUM IKAN
BADUT (Amphiprion percula) BERBASIS MIKROKONTROLER ARDUINO
DUE*

Nama Mahasiswa : Ikhfal Ruhyadi

NIM : 125060307111019

Program Studi : Teknik Elektro

Konsentrasi : Teknik Kontrol

KOMISI PEMBIMBING :

Ketua : Ir. Purwanto, M.T.

Anggota : Goegoes Dwi Nusantoro, S.T., M.T.

TIM DOSEN PENGUJI :

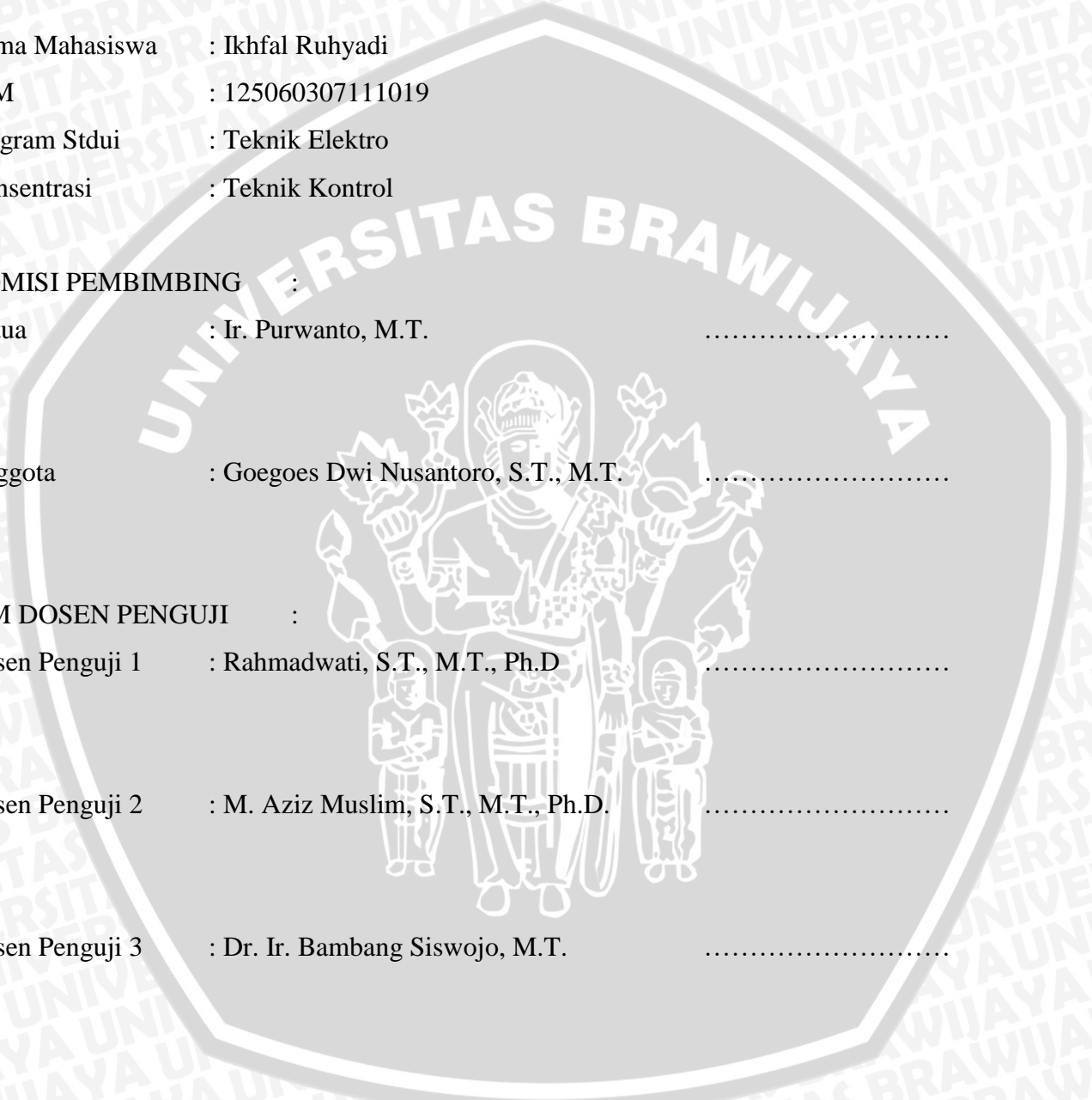
Dosen Penguji 1 : Rahmadwati, S.T., M.T., Ph.D.

Dosen Penguji 2 : M. Aziz Muslim, S.T., M.T., Ph.D.

Dosen Penguji 3 : Dr. Ir. Bambang Siswojo, M.T.

Tanggal Ujian : 9 Agustus 2016

SK Penguji : No.973/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, 12 Agustus 2016

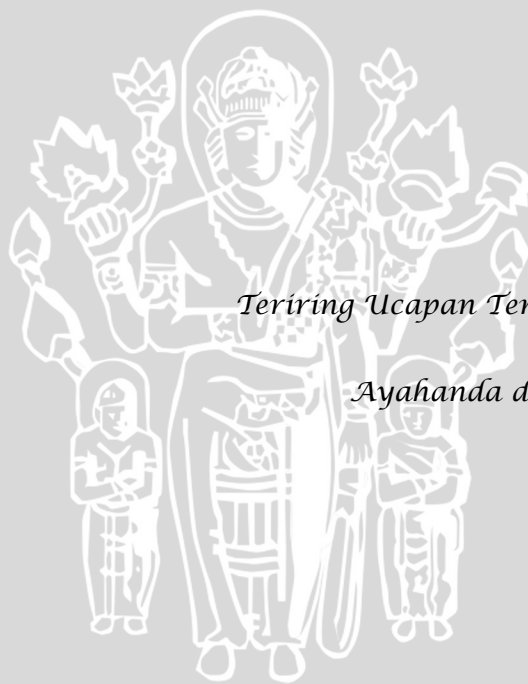
Mahasiswa,

IKHFAL RUHYADI

NIM. 125060307111019



UNIVERSITAS BRAWIJAYA



Teriring Ucapan Terima Kasih kepada:

Ayahanda dan Ibunda tercinta

RINGKASAN

Ikhfal Ruhyadi, Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya, Agustus 2016, *Pengendalian Suhu Dan Salinitas Air Pada Akuarium Ikan Badut (*Amphiprion percula*) Berbasis Mikrokontroler Arduino Due*, Dosen Pembimbing : Purwanto dan Goegoes Dwi Nusantoro.

Upaya pemanfaatan sumber daya laut yang optimal merupakan tuntutan bagi masyarakat untuk meningkatkan kemakmuran dan kesejahteraan masyarakat, memperluas lapangan kerja dan memberikan peluang usaha yang lebih besar, salah satunya adalah peluang usaha dalam pengembangan akuarium air asin atau air laut. Pemilihan ikan badut dalam penelitian ini dikarenakan ikan badut dapat dibudidayakan dan termasuk ikan hias air asin yang banyak ditemukan di Indonesia. Selain itu, ditinjau dari faktor ekonomi ikan badut juga ikan yang dapat dikatakan murah sehingga menjadi ikan yang banyak diminati para pecinta ikan hias air asin. Kendala yang sering terjadi dalam mengembangkan akuarium air asin / air laut adalah menjaga parameter kualitas air agar ikan pada akuarium air asin / laut dapat berkembang dengan baik. Parameter – parameter tersebut misalnya salinitas, kesadahan, ph air, zat, fosfat, unsur – unsur trase, nitrat, kapur, dan tempertur dari air tersebut.

Pada penelitian ini akan akan dirancang alat pengendali suhu dan salinitas air pada akuarium ikan badut (*Amphiprion percula*) agar keadaan parameter suhu dan salinitas air dalam akuarium dapat terjaga dengan suhu sekitar 25°C - 28°C dan dengan salinitas sekitar 26.6 ppt - 33.2 ppt. Sensor yang digunakan untuk melakukan perekaman keadaan suhu dan salinitas yaitu sensor suhu DS12B20 dan sensor salinitas berbasis hidrometer dan sensor *hall effect*.

Hasil yang didapatkan dalam penelitian ini yaitu kontrol suhu air pada akuarium ikan badut yang menggunakan kontroler on/off dengan *setpoint* 25°C - 28°C dapat berjalan dengan error rata-rata sebesar 3,12 % dengan pendingin ruangan mati dan error rata-rata sebesar 2,75 % dengan pendingin ruangan menyala terhadap nilai tengah (26,5 °C). Untuk Kontrol salinitas air pada akuarium ikan badut yang menggunakan kontroler logika *fuzzy* dengan *setpoint* 27 ppt - 33 ppt dapat berjalan dengan error rata-rata sebesar 1,7476 ppt atau sekitar 5,825 % terhadap nilai tengah (30 ppt).

Kata kunci : Pengendali suhu dan salinitas, ikan badut, kontroler on/off, logika *fuzzy*.

SUMMARY

Ikhfal Ruhyadi, Department of Electrical Engineering, Faculty of Engineering University of Brawijaya, August 2016, "*Control of Water Temperature and Salinity In Clown Fish (Amphiprion percula) Aquarium Based on Arduino Due Microcontroller*", Academic Supervisor: Purwanto and Goegoes Dwi Nusantoro.

Optimal efforts utilization of marine resources is a demand for people to increase the prosperity and welfare of the community, increase employment and provide greater business opportunities, one of which is the development of business opportunities in saltwater aquariums or seawater. Selection of a clown fish in this study because the clown fish can be cultivated and including saltwater ornamental fish which are found in Indonesia. Moreover, in terms of economic factors clown fish also inexpensive fish that can be said that it becomes a fish that attracted many lovers saltwater ornamental fish. The often problem of developing saltwater / seawater aquarium is maintaining water quality parameters in order to fish in saltwater / seawater aquarium thrive. The parameters for example salinity, hardness, water pH, solids, phosphates, trace elements, nitrate, limestone, and water temperature.

In this research will be designed a control device of water temperature and salinity in clown fish (*Amphiprion percula*) aquarium in order to state the parameters of temperature and salinity of the water in the aquarium can be maintained at a temperature of about $25^{\circ}\text{C} - 28^{\circ}\text{C}$ and the salinity of approximately 26.6 ppt - 33.2 ppt, Sensors are used to make the recording state is DS12B20 temperature sensor and salinity sensor based on hydrometer and hall effect sensor.

Results obtained in this research that water temperature control in the clown fish aquarium using the on / off controller with a setpoint of $25^{\circ}\text{C} - 28^{\circ}\text{C}$ can be run with an average error of 3.12% with no air conditioners and the average error of 2.75% with air conditioner turns on from the middle value of setpoint (26.5°C). And salinity control in the clown fish aquarium using fuzzy logic controller with setpoint 27 ppt - 33 ppt can be run with an average error of 1.7476 ppt or approximately 5.825% of the value of the middle setpoint (30 ppt).

Keywords: Temperature and salinity controller, clownish, on/off controller, *fuzzy logic*.



PENGANTAR

Bismillahirrohmanirrohim. Alhamdulillah, puji syukur penulis panjatkan kehadiran Allah SWT yang telah memberikan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul “Pengendalian Suhu dan Salinitas Air pada Akuarium Ikan Badut (*Amphiprion Percula*) Berbasis Mikrokontroler Arduino Due” dengan baik. Tak lepas shalawat serta salam tercurahkan kepada junjungan kita Nabi Muhammad SAW yang telah menjadi suri tauladan bagi yang mengharapkan rahmat dan hidayah-Nya.

Penulis menyadari bahwa penyusunan skripsi ini tidak lepas dari bantuan, bimbingan serta dorongan dari berbagai pihak. Pada kesempatan ini penulis menyampaikan rasa terima kasih yang sebesar – besarnya kepada:

1. Allah SWT yang telah memberikan kelancaran, kemudahan dan hidayah-Nya.
2. Kedua orang tua penulis, Bapak Junaedi dan Ibu Suprianah, kakak penulis Eva Yusniati dan Desiyatul Umroh, adik penulis Akis Bahrul Ulum dan Sucimanik Astagina, atas kasih sayang, dukungan, pengorbanan, serta memberikan semangat yang tanpa henti.
3. Kakak Ipar Dadang Hendrayana dan Luqman Hakim terima kasih atas waktu dan pengalaman serta bimbingan dalam pembuatan skripsi ini.
4. Bapak M. Aziz Muslim, S.T., M.T., Ph.D. selaku Ketua Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya Malang.
5. Bapak Hadi Suyono S.T., M.T., Ph.D selaku Sekretaris Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya.
6. Bapak M. Ali Mustofa, S.T., M.T. selaku Ketua Program Studi Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya telah banyak memberikan pengarahan dalam hal akademik dan penulisan skripsi.
7. Bapak Ir. Purwanto, M.T. dan Bapak Goegoes Dwi Nusantoro, S.T., M.T. selaku dosen pembimbing skripsi yang telah memberikan banyak waktu dan tenaganya untuk membimbing dari awal, memberikan saran, nasehat-nasehat, dan pelajaran.
8. Bapak, Ibu dosen serta segenap staf dan karyawan Jurusan Teknik Elektro baik secara langsung maupun tidak langsung yang telah banyak membantu dalam menyelesaikan skripsi ini.
9. Pranata Laboratorium Komputasi dan Jaringan Bapak Nugroho Madiantoko., A.Md. atas semua fasilitas dan bantuan yang telah disediakan selama pengerjaan skripsi.
10. Kakak-kakak, adik-adik, teman-teman asisten Laboratorium Komputasi dan Jaringan, Mas Fahad, Mas Krisna, Mas Firman, Mba Aya, Mas Abdur, Mas Fahmi, Mba Liza,

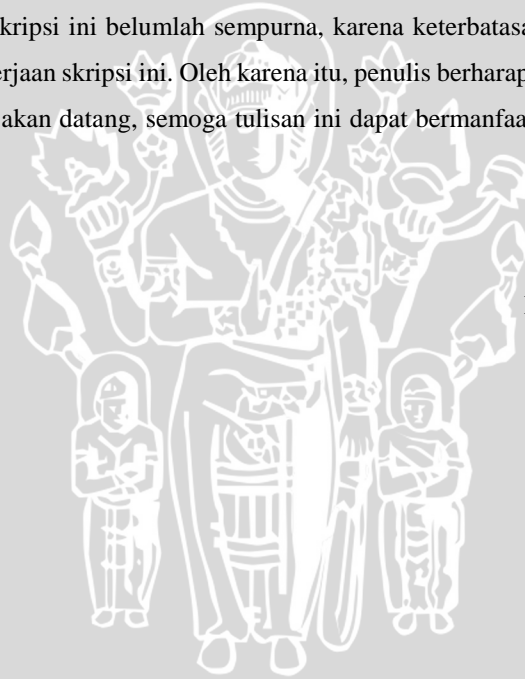
Akbar, Sirojul, Ryan, Muslichin, Gladys, Ainur, Ulya, terimakasih atas segala dukungan, motivasi dan diskusi dalam pengerjaan skripsi ini.

11. Teman-teman seperjuangan angkatan 2012 atas segala dukungan dan bantuan dalam pengerjaan skripsi ini.
12. Teman-teman Workshop Divisi Otomasi, terimakasih atas pengalaman, semangat dan kerjasama yang terjalin dalam organisasi.
13. Sahabat dan teman-teman, Else Surya Ningsih, Fadli, Toni, Rachmat, Adin, terimakasih atas segala diskusi, keceriaan, semangat dan dukungan dalam pengerjaan skripsi ini.
14. Semua pihak yang telah memberikan bantuan serta dukungan baik secara langsung maupun tidak langsung atas penyusunan skripsi ini.

Sekiranya Allah SWT membalas kebaikan semua pihak yang turut membantu skripsi ini terselesaikan. Akhirnya penulis menyadari bahwa skripsi ini belumlah sempurna, karena keterbatasan ilmu dan kendala – kendala lain yang terjadi selama pengerjaan skripsi ini. Oleh karena itu, penulis berharap kritik dan saran untuk penyempurnaan tulisan di masa yang akan datang, semoga tulisan ini dapat bermanfaat dan dapat digunakan untuk pengembangan lebih lanjut.

Malang, Agustus 2016

Penulis



DAFTAR ISI

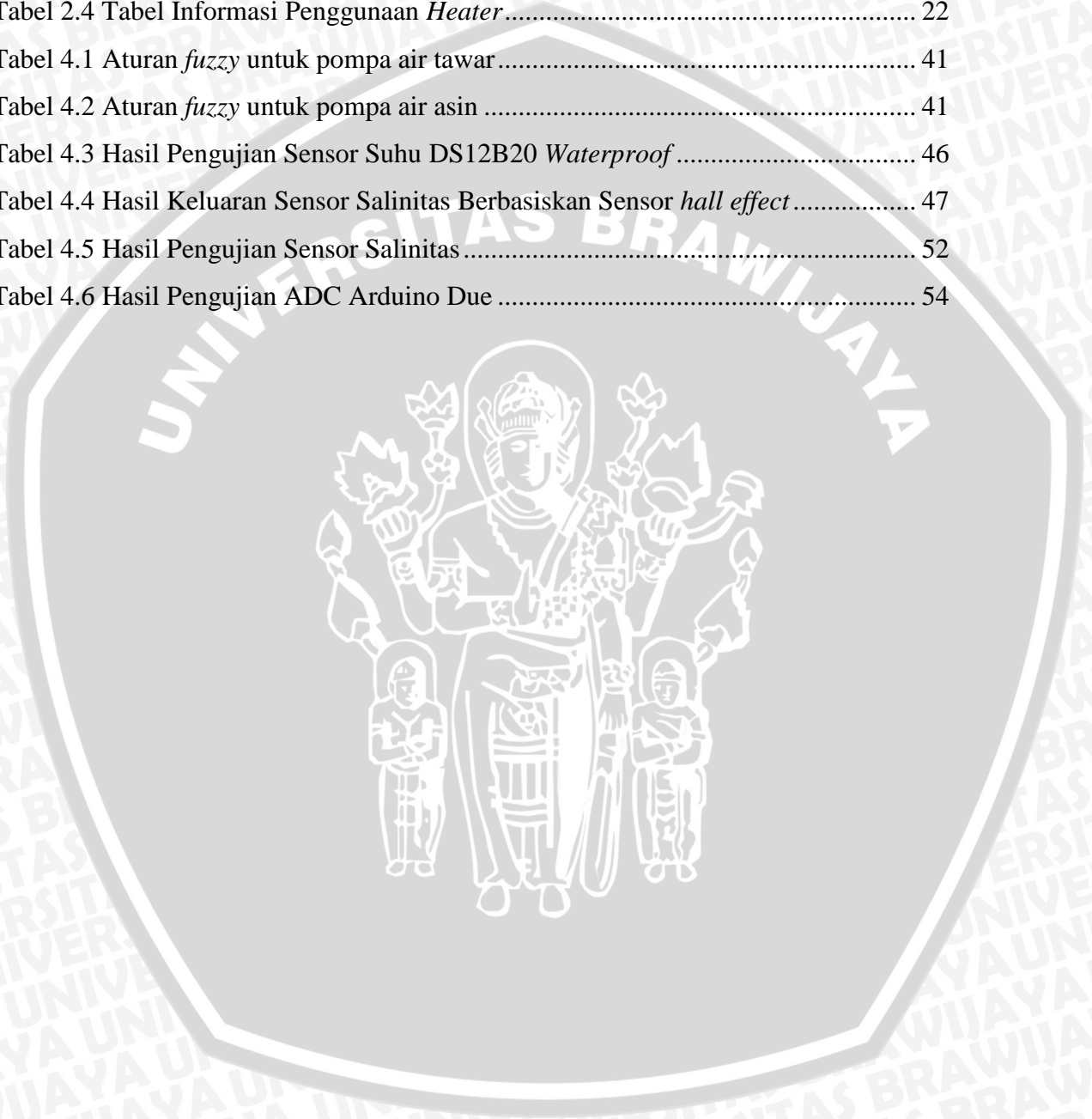
	Halaman
PENGANTAR	i
DAFTAR ISI	iii
DAFTAR TABEL	v
DAFTAR GAMBAR	vii
DAFTAR LAMPIRAN	ix
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Identifikasi Masalah	2
1.3 Rumusan Masalah	3
1.4 Batasan Masalah	3
1.5 Tujuan Penelitian	4
1.6 Sistematika Penulisan	4
BAB II TINJAUAN PUSTAKA	5
2.1 Akuarium Air Laut	5
2.2 Ikan Badut	6
2.3 Suhu Air	7
2.4 Amonia	8
2.5 Salinitas	9
2.6 Kontroler	9
2.7 Kontrol Logika <i>Fuzzy</i>	8
2.7.1 <i>Fuzzifikasi</i>	10
2.7.2 <i>Aturan Fuzzy (Fuzzy Rule)</i>	10
2.7.3 Metode Inferensi MIN-MAX	11
2.7.4 Metode Defuzzifikasi <i>Weighted Average</i>	12
2.8 Arduino Due	13
2.9 LCD 16x2	15
2.10 Sensor Suhu DS12B20 <i>Waterproof</i>	16
2.11 Hidrometer	18
2.12 Pompa Air 6V DC	19
2.13 Kipas 12V DC	21
2.14 <i>Heater</i>	22
BAB III METODE PENELITIAN	25



3.1 Penentuan Spesifikasi Alat	25
3.2 Perancangan dan Pembuatan Alat	26
3.2.1 Perancangan Mekanik	26
3.2.2 Perancangan Perangkat Keras	26
3.2.3 Perancangan Perangkat Lunak	26
3.3 Pengujian Alat	27
3.3.1 Pengujian Sub Sistem	27
3.3.2 Perancangan Keseluruhan Sistem	28
3.4 Pengambilan Kesimpulan dan Saran	28
BAB IV PERANCANGAN, PENGUJIAN DAN ANALISIS	29
4.1 Perancangan dan Pembuatan Alat	29
4.1.1 Perancangan Mekanik	29
4.1.2 Perancangan Perangkat Keras	30
4.1.2.1 Perancangan Rangkaian Sensor Suhu	32
4.1.2.2 Perancangan Rangkaian Sensor Salinitas	32
4.1.2.3 Perancangan Rangkaian Kipas dan <i>Heater</i>	35
4.1.2.4 Perancangan Rangkaian <i>Driver</i> Pompa Motor DC	36
4.1.3 Perancangan Perangkat Lunak	37
4.1.3.1 Perancangan Perangkat Lunak Kontrol Suhu	38
4.1.3.2 Perancangan Perangkat Lunak Kontrol Salinitas	39
4.1.3.3 Perancangan Perangkat Lunak Keseluruhan Sistem	43
4.2 Pengujian Sistem	45
4.2.1 Pengujian Sensor Suhu DS12B20 <i>Waterproof</i>	45
4.2.2 Pengujian Sensor Salinitas	47
4.2.3 Pengujian ADC (<i>Analog to Digital Converter</i>) Arduino Due	53
4.2.4 Pengujian Keseluruhan Sistem	55
4.2.4.1 Pengujian Kontrol Suhu	55
4.2.4.2 Pengujian Kontrol Salinitas	57
BAB V KESIMPULAN DAN SARAN	59
5.1 Kesimpulan	59
5.2 Saran	59
DAFTAR PUSTAKA	61
LAMPIRAN	63

DAFTAR TABEL

No	Judul	Halaman
Tabel 2.1	Kerabat Ikan Badut.....	6
Tabel 2.2	Spesifikasi teknis dari Arduino Due.....	13
Tabel 2.3	Daftar fungsi pin <i>Liquid Crystal Display</i> 16x2.....	16
Tabel 2.4	Tabel Informasi Penggunaan <i>Heater</i>	22
Tabel 4.1	Aturan <i>fuzzy</i> untuk pompa air tawar.....	41
Tabel 4.2	Aturan <i>fuzzy</i> untuk pompa air asin.....	41
Tabel 4.3	Hasil Pengujian Sensor Suhu DS12B20 <i>Waterproof</i>	46
Tabel 4.4	Hasil Keluaran Sensor Salinitas Berbasiskan Sensor <i>hall effect</i>	47
Tabel 4.5	Hasil Pengujian Sensor Salinitas.....	52
Tabel 4.6	Hasil Pengujian ADC Arduino Due.....	54





DAFTAR GAMBAR

No	Judul	Halaman
	Gambar 1.1 Negara penyuplai dan importir ikan hias.....	1
	Gambar 2.1. Akuarium Air Laut	6
	Gambar 2.2 Ikan badut asli (<i>Amphiprion percula</i>).....	7
	Gambar 2.3 (a) Termometer celup Digital, (b) Termometer Celup Anaog	8
	Gambar 2.4. Inferensi <i>fuzzy</i> dengan metode MAX-MIN	12
	Gambar 2.5 Bentuk fisik Arduino Due.....	13
	Gambar 2.6. Tampilan Software Arduino	15
	Gambar 2.7. LCD (<i>Liquid Crystal Display</i>)	15
	Gambar 2.8. Diagram blok Sensor Suhu DS18B20 <i>Waterproof</i>	17
	Gambar 2.9 Sensor Suhu DS18B20 <i>Waterproof</i>	18
	Gambar 2.10 Hidrometer celup	19
	Gambar 2.11 Pompa air DC	19
	Gambar 2.12 Pompa Sistem Rotasi.....	20
	Gambar 2.13 Pompa Sistem Sentrifugal	20
	Gambar 2.14 Kipas 12V DC	21
	Gambar 2.15 Stator dan Rotor <i>BLDC</i> Motor.....	22
	Gambar 2.16 Heater Akuarium	23
	Gambar 3.1 Diagram alir metode penelitian	25
	Gambar 4.1 Desain Alat Pengendalian Suhu dan Salinitas	29
	Gambar 4.2 Diagram Blok Sistem Pengendalian Suhu dan Salinitas	30
	Gambar 4.3 Diagram Blok Alat Pengendalian Suhu dan Salinitas	31
	Gambar 4.4 Rangkaian penggunaan sensor suhu DS18B20 <i>Waterproof</i>	32
	Gambar 4.5 Hasil Perancangan Sensor Salinitas Berbasis Sensor <i>hall effect</i>	33
	Gambar 4.6 Konfigurasi Pin Sensor Salinitas Berbasis Sensor <i>hall effect</i>	34
	Gambar 4.7 Konfigurasi Pin Arduino dan actuator (kipas dan <i>heater</i>).....	36
	Gambar 4.8 Konfigurasi Pin Arduino dan <i>driver L928N</i>	37
	Gambar 4.9 Diagram alir perangkat lunak kontrol suhu	38
	Gambar 4.10 (a) Fungsi keanggotaan <i>error</i> , (b) Fungsi Keanggotaan <i>delta error</i>	40
	Gambar 4.11 (a) Fungsi keanggotaan Pompa Air Tawar, (b) Fungsi Keanggotaan Pompa Air Asin	40
	Gambar 4.12 (a) Diagram Alir Kontrol Salinitas, (b) Diagram Alir Fungsi FIS	

(Fuzzy Inference System)43

Gambar 4.13 Diagram Alir Kontrol Suhu dan Salinitas44

Gambar 4.14 Grafik Data Termometer dan Sensor Suhu DS12B20.....47

Gambar 4.15 Grafik Data Keluaran Sensor Salinitas Berbasis Sensor *hall effect* ..48

Gambar 4.16 Kurva Keluaran Sensor Salinitas dan Hasil Regresi Data 1-15
(20 – 34 ppt).....50

Gambar 4.17 Kurva Keluaran Sensor Salinitas dan Hasil Regresi Data 15 – 21
(34 – 40 ppt).....51

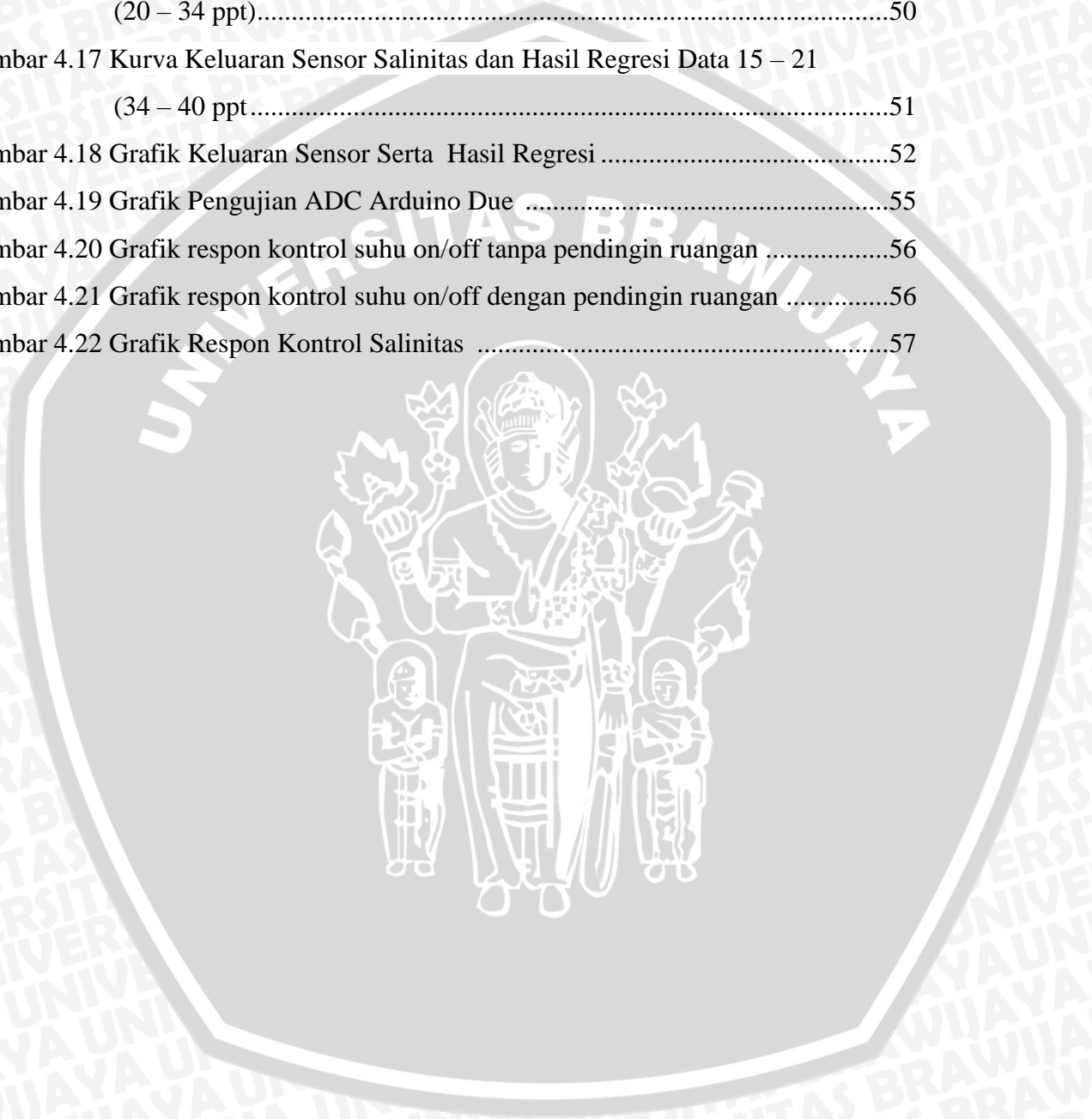
Gambar 4.18 Grafik Keluaran Sensor Serta Hasil Regresi52

Gambar 4.19 Grafik Pengujian ADC Arduino Due55

Gambar 4.20 Grafik respon kontrol suhu on/off tanpa pendingin ruangan56

Gambar 4.21 Grafik respon kontrol suhu on/off dengan pendingin ruangan56

Gambar 4.22 Grafik Respon Kontrol Salinitas57



DAFTAR LAMPIRAN

No	Judul	Halaman
Lampiran 1.	Dokumentasi Alat	63
Lampiran 2.	Listing Program	65
Lampiran 3.	<i>Datasheet</i> Komponen.....	77



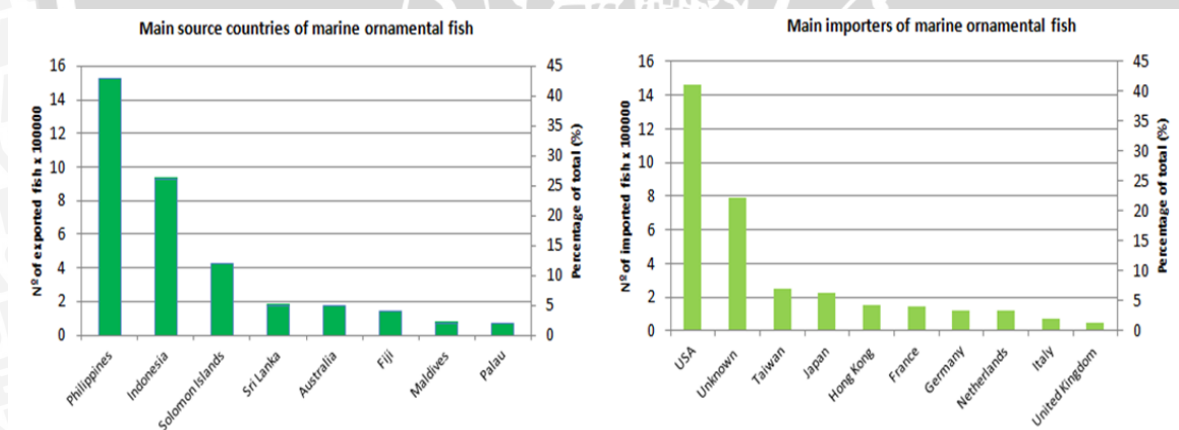


BAB I

PENDAHULUAN

1.1 Latar Belakang

Indonesia merupakan salah satu negara kepulauan yang memiliki sumber daya laut yang melimpah. Hasil dari penelitian Badan Informasi Geospasial (BIG) (2013), panjang total garis pantai Indonesia yang mencapai 99.093 km membuat Indonesia memiliki potensi yang sangat besar untuk pengembangan budidaya biota laut. Upaya pemanfaatan sumber daya laut yang optimal merupakan tuntutan bagi masyarakat untuk meningkatkan kemakmuran dan kesejahteraan masyarakat, memperluas lapangan kerja dan memberikan peluang usaha yang lebih besar, salah satunya adalah peluang usaha dalam pengembangan akuarium air asin atau air laut. Kendala yang sering terjadi dalam menembangkan akuarium air asin / air laut adalah menjaga parameter kualitas air agar terumbu karang dan ikan akuarium air asin / laut dapat berkembang dengan baik. Parameter – parameter tersebut misalnya salinitas, alkalinitas, kandungan ammonia, kalsium, nitrat, nitrit, fosfat, ph air, dan tempertur dari air tersebut (Albert, 2015). Menurut data dari eksportir, Filipina, Indonesia, Pulau Solomon, Sri Lanka, Australia, Fiji, Maldives dan Palau adalah negara-negara utama penyuplai ikan hias. 98% ikan hias disuplai oleh negara-negara tersebut dari seluruh jumlah ikan hias yang di import dari tahun 1997 hingga tahun 2002 (Wabnitz, 2003). Gambar negara-negara eksport dan importir ikan hias dapat ditunjukkan pada Gambar 1.1.



Gambar 1.1. Negara penyuplai dan importir ikan hias

(Sumber: María menggunakan data Wabnitz, 2003)



Ikan badut atau *clownish* (*Amphiprion percula*) adalah salah ikan laut dari keluarga *Pomacentridae* yang populer untuk dipelihara dalam akuarium air asin atau air laut. Selain keindahan ikan badut yang menarik hati para *hobbyist* akuarium air asin, ikan ini juga banyak terdapat di perairan Indonesia dan juga ikan badut sudah dapat dikembang biakan pada akuarium. Ikan badut juga merupakan salah satu ikan yang banyak ditemukan di perairan Indonesia.

Salinitas adalah kadar garam seluruh zat yang larut dalam 1.000 gram air laut, dengan asumsi bahwa seluruh karbonat telah diubah menjadi oksida, semua brom dan lod diganti dengan khlor yang setara dan semua zat organik mengalami oksidasi sempurna (Forch, 1902 dalam Sverdrup, 1942). Salinitas mempunyai peran penting dan memiliki ikatan erat dengan kehidupan organisme perairan termasuk ikan, dimana secara fisiologis salinitas berkaitan erat dengan penyesuaian tekanan osmotik ikan tersebut. Air laut secara alami merupakan air *saline* dengan kandungan garam sekitar 3,5% atau 35 ‰ / 35 ppt (*part per thousand*). Ikan badut sendiri membutuhkan keadaan air dengan salinitas 27 - 33 ppt.

Suhu adalah besaran numerik untuk mengetahui derajat panas atau dingin pada suatu benda. Suhu juga dapat didefinisikan sebagai suatu besaran termodinamika yang menunjukkan besarnya energi kinetik translasi rata-rata molekul dalam sistem gas. Menurut Laevastu dan Hela (1970), pengaruh suhu terhadap ikan adalah dalam proses metabolisme, seperti pertumbuhan dan pengambilan makanan, aktivitas tubuh, seperti kecepatan renang, serta dalam rangsangan syaraf. Organisme perairan seperti ikan maupun udang mampu hidup baik pada kisaran suhu 20°C - 30°C. Perubahan suhu di bawah 20°C atau di atas 30°C menyebabkan ikan mengalami stres yang biasanya diikuti oleh menurunnya daya cerna (Trubus Edisi 425, 2005). Untuk ikan badut, suhu optimum agar ikan dapat berkembang dengan baik adalah 25 °C – 28 °C. Oleh karena itu, dibutuhkan suatu kontroler yang dapat mempertahankan suhu air yaitu sekitar 25 °C – 28 °C dan juga salinitas air memiliki kandungan garam sekitar 26.6 - 33.2 ppt.

1.2 Identifikasi Masalah

Dalam penelitian ini, dirancang alat yang dapat digunakan untuk mengendalikan suhu dan salinitas air pada akuarium air laut dengan mengaplikasikan logika *fuzzy* sebagai kontroler. Pengontrolan suhu dan salinitas menggunakan sensor suhu dan sensor konduktivitas untuk mengidentifikasi kondisi air pada akuarium dan sebagai masukan yang akan diproses pada logika *fuzzy* untuk mengambil keputusan kontrol.

1.3 Rumusan Masalah

Berdasarkan uraian latar belakang di atas, maka dapat disusun rumusan masalah sebagai berikut:

- 1 Bagaimana merancang dan membuat sistem pengendalian suhu menggunakan kontrol ON/OFF dan salinitas air pada akuarium air laut menggunakan kontrol logika *fuzzy*?
- 2 Bagaimana merancang *software* sistem pengendalian suhu menggunakan kontrol ON/OFF dan salinitas air pada akuarium air laut menggunakan kontrol logika *fuzzy*?

- 3 Apakah sistem pengendalian suhu menggunakan kontrol ON/OFF dan salinitas air menggunakan kontrol logika *fuzzy* dapat diterapkan pada system akuarium ikan badut?

1.4 Batasan Masalah

Dengan mengacu pada permasalahan yang telah dirumuskan, maka hal-hal yang berkaitan dengan alat diberi batasan sebagai berikut :

1. Alat pengendalian salinitas air pada akuarium air laut yang digunakan adalah model miniatur yang dibuat dengan desain sendiri.
2. Didalam akuarium hanya terdapat ikan badut (*Amphiprion percula*) tanpa terumbu karang.
3. Menggunakan sensor suhu DS18B20 *Waterproof* dan sensor salinitas hasil perancangan.
4. Menggunakan 4 buah kipas motor dc 12V diameter 5 cm, dan 1 buah kipas motor dc 12V diameter 12 cm sebagai pendingin .
5. Menggunakan logika fuzzy dalam menentukan aksi kontrol dalam pengendalian salinitas air dan kontrol on/off untuk pengendalian suhu air pada akuarium ikan badut.
6. Menggunakan 2 buah pompa air 6V DC.
7. Menggunakan *Heater Kandila KD-75*.
8. Menggunakan *Driver Dual Motor Controller L928N H-Bridge*.
9. Menggunakan 2 buah modul relay 5 V.
10. Ukuran akuarium adalah 40 cm x 45 cm x 35 cm.
11. Pengujian dilakukan pada suhu ruang $\pm 27^{\circ}$ C.
12. Pada perangkat keras kontroler menggunakan Arduino Due.
13. Nilai suhu dan salinitas air ditampilkan pada LCD.

1.5 Tujuan Penelitian

Tujuan penelitian ini adalah merancang dan membuat alat pengendali suhu dan salinitas air pada akuarium ikan badut asli (*Amphiprion percula*) berbasis mikrokontroler Arduino Due.

1.6 Sistematika Penulisan

Sistematika penulisan dalam penelitian ini adalah sebagai berikut :

BAB I Pendahuluan

Berisi pendahuluan yang meliputi latar belakang, rumusan masalah, batasan, tujuan, manfaat dan sistematika pembahasan.

BAB II Landasan Teori

Membahas teori-teori yang mendukung dalam perancangan dan pembuatan alat.

BAB III Metodologi Penelitian

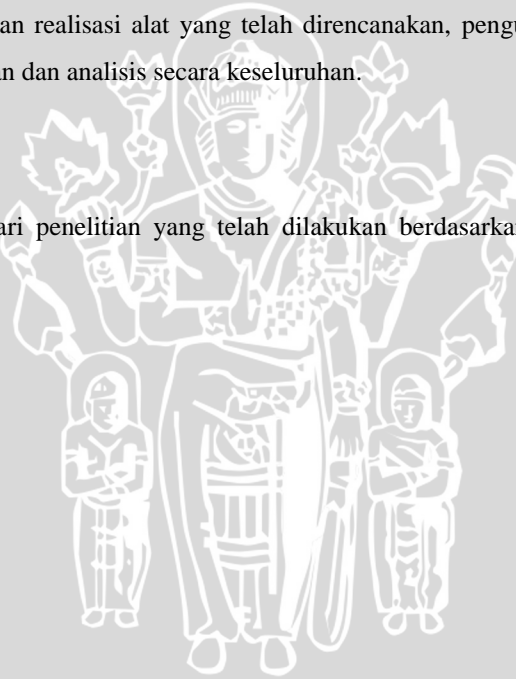
Berisi metode-metode dalam penelitian yang meliputi metode perancangan dan pembuatan alat. Perancangan dan pembuatan alat berisi tentang perancangan dan analisis terhadap spesifikasi alat, meknik, *hardware*, dan *software* dari alat.

BAB IV Perancangan, Pengujian dan Analisis

Berisi tentang perancangan dan realisasi alat yang telah direncanakan, pengujian dan analisis dari masing-masing blok alat serta pengujian dan analisis secara keseluruhan.

BAB V Kesimpulan dan Saran

Berisi tentang kesimpulan dari penelitian yang telah dilakukan berdasarkan masalah yang telah dirumuskan sebelumnya.



BAB II

TINJAUAN PUSTAKA

Pengetahuan dan pemahaman yang dibutuhkan peneliti yang mendukung perancangan dan realisasi alat meliputi pengetahuan tentang akuarium air laut, suhu dan salinitas air laut, kontroler, kontrol logika *fuzzy*, sensor suhu, sensor konduktivitas, mikrokontroler, dan motor dc.

2.1 Akuarium Air Laut

Akuarium (*aquarium* – *aqua* dalam bahasa latin berarti air dan akhiran *-rium* artinya tempat atau bangunan) adalah suatu tempat, yang umumnya terbuat dari bahan gelas atau plastik tembus pandang, berisi air dengan ikan, binatang, dan tumbuhan hidup didalamnya (Mismail, 2010 : 1). Pengertian Laut adalah ”kumpulan air asin yang banyak atau luas yang memisahkan benua dan benua, pulau dengan pulau dan sebagainya.” (Departemen Pendidikan dan Kebudayaan Republik Indonesia, 2001). Wujud akuarium laut merupakan suatu wadah untuk menampilkan kehidupan ekosistem laut dengan kondisi lingkungan yang dibuat menyerupai aslinya. Dalam kamus besar bahasa indonesia terbitan Balai Pustaka, akuarium memiliki pengertian yaitu suatu tempat atau sarana dimana koleksi-koleksi yang berhubungan dengan kehidupan dalam air disimpan dan diperagakan. Wujud akuarium berupa bak kaca (biasanya diberi tanaman air, dll) tempat memelihara ikan hias (Balai Pustaka, 1991).

Akuarium air laut dapat dikelompokkan menjadi 2 macam, yaitu akuarium air laut yang berisi ikan saja dan akuarium terumbu karang yang berisi koral hidup, ikan tak bertulang belakang (*invertebrate*) lain yang biasa hidup di terumbu karang di lautan bebas. Bahan utama untuk akuarium ada 2 macam, gelas dan akrilik. Umumnya akuarium akrilik tidak memiliki sambungan. Berbeda dengan akuarium gelas, biasanya terbuat dari kaca yang disambung dengan lem silikon (*silicone*). Disamping akuarium sebagai tempat tinggal dan peragaan utama biota laut, masih diperlukan peralatan lain yang menjamin kelangsungan hidup penghuni akuarium. Peralatan yang diperlukan guna menjamin kelangsungan hidup penghuninya adalah sarana yang menjaga mutu air dan ekologi akuarium itu sendiri. Ekologi akuarium yang ideal adalah yang dapat memberi kesetimbangan seperti yang terjadi dalam alam tetapi di lingkungan yang sangat terbatas. Dalam praktik hal itu hamper tidak mungkin dicapai. Dalam batas – batas tertentu pemilik harus ikut campur-tangan guna

menjaga keseimbangan tersebut. Gambar akuarium air laut dapat ditunjukkan pada Gambar 2.1.



Gambar 2.1. Akuarium Air Laut
(Sumber: Dokumen Pribadi, 2016)

2.2 Ikan Badut (*Amphiprion percula*)

Ikan badut atau dikenal juga sebagai ikan giru atau ikan nemo adalah ikan kecil dari famili *Pomacentridae*. Famili ini termasuk ikan betok atau *damsel*fishes (*Chrysiptera spp.*) dan clownfishes (*Amphiprion* dan *Premnas spp.*). *Amphiprioninae* adalah subfamili dari *Pomacentridae* yang merupakan ikan tropis (6°S - 26°S, 141°E - 155°E). Terdapat 29 spesies yang umum disebut sebagai ikan badut. 29 spesies dikelompokkan dalam genus *Amphiprion* dan sisanya 1 spesies sebagai anggota *Premnas*. 30 spesies kerabat ikan badut ditunjukkan dalam Tabel 2.1.

Tabel 2.1. Kerabat Ikan Badut

<i>Genus</i>	<i>Spesies</i>
<i>Amphiprion</i>	<i>percula, perideraion, polymnus, rubrocintus, sandraracinus, sebae, thiellei, tricinctus, mccullochi, fuscocaudatus, omanensis, akallopisos, latezonatus, latifasciatus, leukokranos, akallopisos, mccullochi, melanopus, nigrisep, ocellaris, akindynos, allardi, bicinctus, chagosensi, chrysogaster, chrysoptarus, clarkii, ephippium, frenatus</i>
<i>Premnas</i>	<i>biaculeatus</i>

(Sumber: FishBase, 2011)

Ikan badut asli (*Amphiprion percula*) hanya terdapat pada kawasan Indo-Pasifik (Rosenberg dan Cruz, 1988). Seperti semua ikan anemon, *A. percula* membentuk hubungan simbiosis dengan anemon laut dengan menggunakan inangnya baik sebagai tempat tinggal atau perlindungan dari predator. Simbiosis melibatkan *A. percula* dan spesies anemon laut seperti *Heteractis magnifica*, *Stichodactyla gigantean*, dan *Stichodactyla mertensii* biasanya ditemukan di alam (Elliott dan Mariscal, 1996). Kedua simbion ini berada di perairan pantai yang dangkal tropis di mana kedalaman jarang melebihi 12 meter dan rentang suhu air 25°C - 28°C (Fautin dan Allen, 1992 dalam Randall, 1997). Gambar ikan badut asli (*Amphiprion percula*) ditunjukkan pada Gambar 2.2.



Gambar 2.2. Ikan badut asli (*Amphiprion percula*)
(Sumber: Walsh, 2015)

A. percula dapat tumbuh hingga panjang 110 mm dan biasanya terdapat tiga garis vertikal putih pada tubuh oranye terang. Garis pertama tepat di belakang mata, garis putih kedua membagi ikan, dan garis putih terakhir di dekat sirip ekor. Selain warna putih, terdapat juga garis tipis warna hitam yang terdapat disetiap sirip dengan berbagai ketebalan (Fautin dan Allen, 1992; Grant, 1999). Tidak ada perbedaan pola warna antara jenis kelamin jantan dan betina. Meskipun demikian, biasanya betina lebih besar dari jantan.

2.3 Suhu Air

Laut tropis merupakan lingkungan yang paling stabil di dunia. Umumnya isi akuarium berasal dari daerah tropis, oleh karena itu keadaan akuarium laut harus mendekati keadaan tersebut. Organisme perairan seperti ikan maupun udang mampu hidup baik pada kisaran suhu 20-30°C. Perubahan suhu di bawah 20°C atau di atas 30°C menyebabkan ikan mengalami stres yang biasanya diikuti oleh menurunnya daya cerna (Trubus Edisi 425, 2005). Suhu dalam akuarium harus berada di sekitar 26°C, dan tidak boleh lebih dari 30°C (Mismail, 2010: 8). Sumber lain mengatakan bahwa suhu optimum untuk akuarium laut

adalah sekitar 75°F-82°F atau sekitar 24°C-28°C (Moe, 1992 :108). pengaruh suhu terhadap ikan adalah dalam proses metabolisme, seperti pertumbuhan dan pengambilan makanan, aktivitas tubuh, seperti kecepatan renang, serta dalam rangsangan syaraf. Akibatnya penghuni akuarium dapat menggunakan lebih banyak oksigen, karbon dioksida, nutrisi, kalsium dan alkalinitas pada suhu yang lebih tinggi. Dampak lain dari pengaruh suhu yaitu pada aspek kimia akuarium. Kelarutan gas-gas terlarut seperti oksigen dan karbon dioksida, berubah terhadap suhu. Oksigen misalnya, kurang larut pada suhu yang lebih tinggi.

Di Negara-negara dengan empat musim, umumnya diperlukan *heater* untuk memanaskan suhu air. Di Indonesia kecenderungannya adalah menurunkan suhu agar tidak melebihi 30°C. Menurunkan suhu ini dapat dilakukan dengan peralatan pendingin khusus (*chiller*) atau dengan kipas angin yang diletakan dibawah sangkup akuarium.

Termometer celup adalah alat yang umum digunakan untuk mengukur suhu dari air pada akuarium. Ada 2 macam thermometer celup untuk akuarium, yaitu termometer digital dan analog. Umumnya harga termometer celup digital lebih mahal dari termometer analog. Gambar termometer celup dapat ditunjukkan pada Gambar 2.3.



Gambar 2.3 (a) Termometer celup Digital

(b) Termometer Celup Anaog

2.4 Amonia

Pengaruh amonia bagi kondisi lingkungan sudah banyak dipelajari secara mendalam. Amonia bersifat racun baik pada ikan air tawar atau bagi ikan air laut. Dalam larutan, terdapat 2 bentuk amonia, terionisasi (NH_4^+) dan tidak terionisasi (NH_3), direpresentasikan dengan persamaan :



Wuhrmann dan Woker 1948, bersama Downing dan Merkens 1955 dalam Carol E. Bower dan Joseph P. Bidwel, NH_3 dianggap sebagai bentuk amonia yang merupakan racun bagi ikan air tawar dan air laut. Kadar amonia dapat dipengaruhi beberapa faktor seperti suhu, sainitas dan PH air. Konsentrasi NH_3 semakin besar ketika suhu bertambah, dan semakin kecil ketika salinitas semakin besar.

2.5 Salinitas

Air laut secara alami merupakan air *saline* dengan kandungan garam sekitar 3,5% atau 35 ‰ / 35 ppt (*part per thousand*). Beberapa danau garam di daratan dan beberapa lautan memiliki kadar garam lebih tinggi dari air laut umumnya. Berdasarkan persentase garam terlarut, salinitas air dibagi menjadi 4 kondisi. Jika persentase garam terlarut dalam air < 0,05 % air dikategorikan sebagai air tawar, 0,05 – 3 % dikategorikan sebagai air payau, 3 – 5 % dikategorikan sebagai air *saline*, dan > 5 % dikategorikan sebagai air *brine*. Faktor utama yang mempengaruhi salinitas air laut pada laut lepas adalah penguapan, curah hujan dan jumlah sungai yang bermuara pada laut tersebut.

2.6 Kontroler

Kontroler merupakan salah satu komponen sistem yang berfungsi mengolah sinyal umpan balik dan sinyal referensi menjadi sinyal kontrol sedemikian rupa sehingga performansi dari sistem yang dikendalikannya sesuai dengan spesifikasi performansi yang diinginkan.

Keberadaan kontroler dalam sebuah sistem kendali mempunyai kontribusi yang besar terhadap perilaku sistem. Hal ini disebabkan oleh tidak dapat diubahnya komponen penyusun sistem tersebut. Artinya karakteristik plant harus diterima sebagaimana adanya, sehingga perubahan perilaku sistem hanya dapat dilakukan melalui penambahan suatu subsistem yaitu seperti kontroler. 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, 1996: 197-204).

2.7 Kontrol Logika Fuzzy

Kontroler 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*.

Kontroler 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.

2.7.1 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, 1993). Proses fuzzifikasi diekspresikan sebagai berikut:

$$x = \text{fuzzifier}(x_0)$$

Dengan:

x_0 = nilai *crisp* variabel masukan

x = himpunan *fuzzy* variabel yang terdefinisi

fuzzifier = operator fuzzifikasi yang memetakan himpunan *crisp* ke himpunan *fuzzy*

Pedoman memilih fungsi keanggotaan untuk proses fuzzifikasi, menurut Yan, J., Ryan, M., dan Power, J. menggunakan :

1. Himpunan *fuzzy* dengan distribusi simetris.
2. Himpunan *fuzzy* yang digunakan berjumlah ganjil, berkaitan erat dengan jumlah kaidah (*rules*).
3. Himpunan *fuzzy* diatur agar saling menumpuk.
4. Fungsi keanggotaan yang digunakan bentuk segitiga atau trapesium.

2.7.2. 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 ahli. 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

Rule 1 if X is A₁ and Y is B₁ then Z is C₁

Rule 2 if X is A₂ and Y is B₂ then Z is C₂

· · ·
· · ·

Rule n if X is A_n and Y is B_n then Z is C_n

Dengan X, Y, Z merupakan variabel linguistik, dimana X dan Y merupakan variabel masukan, dan Z merupakan variabel keluaran sistem. A_n , B_n , dan C_n merupakan nilai linguistik dari X, Y, dan Z (Lee, 1990).

2.7.3. 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 :

$$\mu_{C'} = \bigcup_1^n \alpha_i \wedge \mu_{C_i} \quad (2-1)$$

Dengan

$$\alpha_i = \mu_{A_i}(x_0) \wedge \mu_{B_i}(y_0) \quad (2-2)$$

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

Rule 1 :

jika x adalah A₁ dan y adalah B₁ maka z adalah C₁

Rule 2 :

jika x adalah A₂ dan y adalah B₂ maka z adalah C₂

Pada metode penalaran MAX-MIN fungsi keanggotaan konsekuen dinyatakan dengan

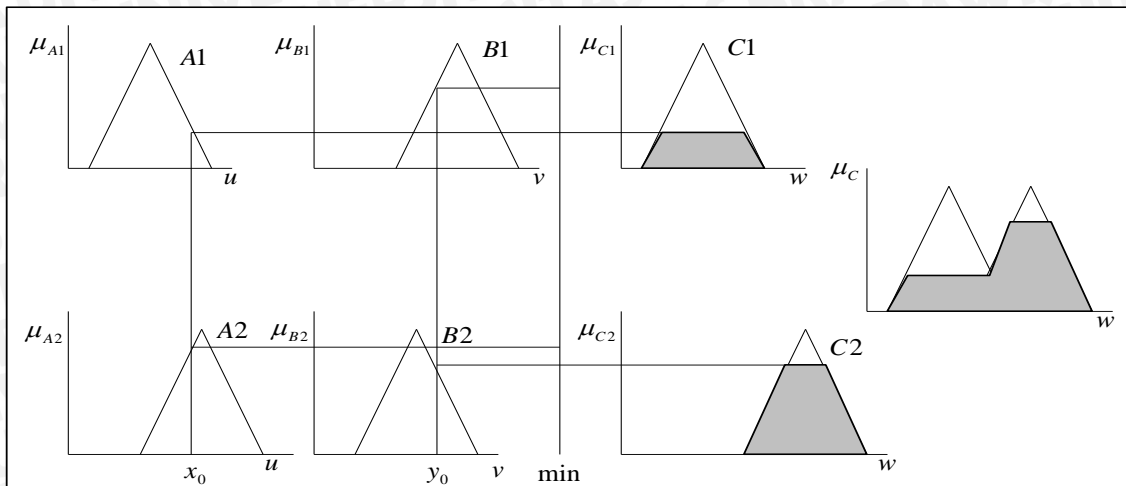
$$\mu_{C'}(w) = \mu_{C'_1} \vee \mu_{C'_2} = [\alpha_1 \wedge \mu_{C_1}(w)] \vee [\alpha_2 \wedge \mu_{C_2}(w)] \quad (2-3)$$

Dimana

$$\alpha_1 = \mu_{A_1}(x_0) \wedge \mu_{B_1}(y_0) \quad (2-4)$$

$$\alpha_2 = \mu_{A_2}(x_0) \wedge \mu_{B_2}(y_0) \quad (2-5)$$

Metode ini dideskripsikan dalam Gambar 2.4.



Gambar 2.4 Inferensi *fuzzy* dengan metode MAX-MIN

(Sumber : Yan, Ryan dan Power ,1993)

2.7.4. Metode Defuzzifikasi *Weighted Average*

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

$$y_0 = \text{defuzzifier}(y) \quad (2-6)$$

Dengan

y = aksi kontrol *fuzzy*

y_0 = aksi kontrol *crisp*

defuzzifier = operator defuzzifikasi

Metode *Weighted Average* menurut Ross didefinisikan sebagai berikut :

$$U = \frac{\sum_{i=1}^n w_i u_i}{\sum_{i=1}^n w_i} \quad (2-7)$$

Dengan

U = Keluaran

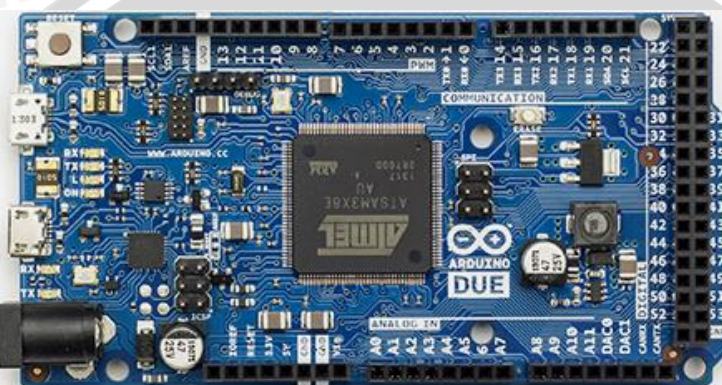
w_i = Bobot nilai benar w_i

u_i = Nilai linguistik pada fungsi keanggotaan keluaran

n = Banyak derajat keanggotaan

2.8 Arduino Due

Arduino Due merupakan board mikrokontroler berbasis CPU Atmel SAM3X8E ARM Cortex-M3. Arduino Due merupakan board Arduino pertama yang menggunakan CPU ARM-32 bit, dengan 32 bit memungkinkan operasi data yang luas hingga 4 byte . Modul ini memiliki 54 digital input/output di mana 12 dari pin yang tersedia digunakan untuk PWM output, 12 analog input, 4 UART, memiliki clock 84 MHz, koneksi USB OTG, 2 DAC (*digital to analog*), 2 TWI, power jack, Port SPI, port JTAG, tombol *reset* dan tombol *erase*. Gambar *hardware* modul Arduino Due dapat ditunjukkan pada Gambar 2.5.



Gambar 2.5 Bentuk fisik Arduino Due
(Sumber : www.arduino.cc)

Banyak pemula yang belajar mengenal robotika dan elektronika lewat Arduino karena mudah dipelajari. Tapi tidak hanya pemula, para hobbyist atau profesional pun ikut senang mengembangkan aplikasi elektronik menggunakan Arduino. Bahasa yang dipakai dalam Arduino bukan Assembler yang relatif sulit, tetapi bahasa C yang disederhanakan dengan bantuan pustaka-pustaka (*libraries*) Arduino. Arduino juga menyederhanakan proses bekerja dengan mikrokontroler. Spesifikasi teknis dari Arduino Due ditunjukkan dalam tabel 2.2.

Tabel 2.2. Spesifikasi teknis dari Arduino Due

<i>Microcontroller</i>	<i>AT91SAM3X8E</i>
<i>Operating Voltage</i>	<i>3.3V</i>
<i>Input Voltage (recommended)</i>	<i>7-12V</i>
<i>Input Voltage (limits)</i>	<i>6-16V</i>
<i>Digital I/O Pins</i>	<i>54 (of which 12 provide PWM output)</i>
<i>Analog Input Pins</i>	<i>12</i>

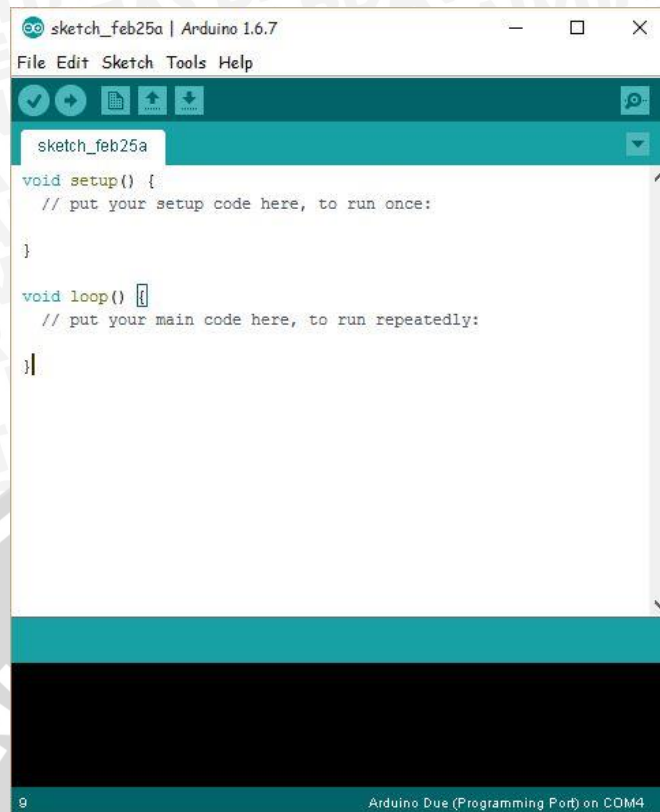
Tabel 2.2. Spesifikasi teknis dari Arduino Due

<i>Analog Output Pins</i>	2 (DAC)
<i>Total DC Output Current on all I/O lines</i>	130 mA
<i>DC Current for 3.3V Pin</i>	800 mA
<i>DC Current for 5V Pin</i>	800 mA
<i>Flash Memory</i>	512 KB all available for the user applications
<i>SRAM</i>	96 KB (two banks: 64KB and 32KB)
<i>Clock Speed</i>	84 MHz
<i>Length</i>	101.52 mm
<i>Width</i>	53.3 mm
<i>Weight</i>	36 g
<i>Microcontroller</i>	AT91SAM3X8E
<i>Operating Voltage</i>	3.3V

Dan sehubungan dengan pembahasan untuk saat ini software Arduino yang akan digunakan adalah Driver dan IDE, walaupun masih ada beberapa software lain yang sangat berguna selama pengembangan Arduino. IDE Arduino adalah software yang sangat *user friendly* dan ditulis dengan menggunakan Java. IDE Arduino terdiri dari:

1. Editor program, sebuah window yang memungkinkan pengguna menulis dan mengedit program dalam bahasa Processing.
2. Compiler, sebuah modul yang mengubah kode program (bahasa Processing) menjadi kode biner.
3. Uploader, sebuah modul yang memuat kode biner dari komputer ke dalam memory di dalam papan Arduino.

Gambar tampilan software Arduino dapat ditunjukkan pada Gambar 2.6.

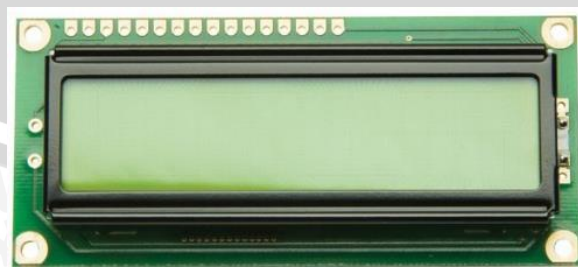


Gambar 2.6 Tampilan Software Arduino.

(Sumber : Dokumen Pribadi, 2016)

2.9 LCD 16x2

LCD (*Liquid Crystal Display*) adalah suatu jenis media tampil yang menggunakan kristal cair sebagai penampil utama yang pengoperasiannya menggunakan sistem *dot* matriks. LCD sudah digunakan diberbagai bidang misalnya alal-alat elektronik seperti televisi, kalkulator, ataupun layar komputer. *LCD* 16x2 artinya *LCD* tersebut dapat menampilkan 32 karakter (2 baris dan 16 kolom). Gambar *LCD* 16x2 dapat ditunjukkan pada Gambar 2.7.



Gambar 2.7 LCD (*Liquid Crystal Display*)

(Sumber : www.buydisplay.com)

Masukan modul *Liquid crystal display* (LCD) ini berupa bus data dan 3 sinyal kontrol yaitu RS, R/W dan E. Pengendali *dot* matriks *liquid crystal display* (LCD) dilakukan secara internal oleh kontroler yang sudah terpasang di dalam modul *liquid crystal display* (LCD). Pin dan fungsi ditunjukkan dalam table 2.3.

Tabel 2.3. Daftar fungsi pin *Liquid Crystal Display* 16x2

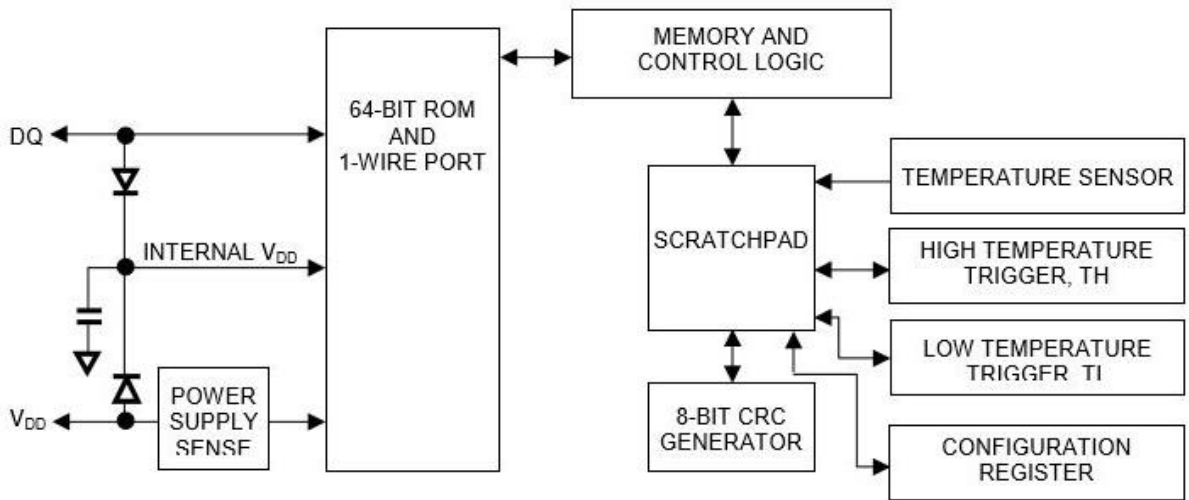
Pin No	Function	Name
1	Ground (0V)	Ground
2	Supply voltage; 5V (4.7V – 5.3V)	Vcc
3	Contrast adjustment; through a variable resistor	VEE
4	Selects command register when low; and data register when high	Register Select
5	Low to write to the register; High to read from the register	Read/write
6	Sends data to data pins when a high to low pulse is given	Enable
7	8-bit data pins	DB0
8		DB1
9		DB2
10		DB3
11		DB4
12		DB5
13		DB6
14		DB7
15	Backlight Vcc (5V)	Led+
16	Backlight Ground (0V)	Led-

(Sumber: www.engineersgarage.com)

2.10 Sensor Suhu DS18B20 *Waterproof*

Sensor suhu DS18B20 *Waterproof* merupakan suatu komponen elektronika yang dapat menangkap perubahan temperatur lingkungan lalu kemudian mengkonversinya menjadi besaran listrik. Sensor suhu DS18B20 *Waterproof* ini merupakan sensor suhu digital

tahan air dengan resolusi 9 – 12 bit yang dapat menunjukkan suhu dari *device* tersebut. Sensor suhu DS18B20 *Waterproof* memiliki range pembacaan suhu dari - 55°C hingga +125°C dengan ketelitian $\pm 0,5^\circ\text{C}$ dari pembacaan -10°C hingga +85°C. Gambar diagram blok sensor suhu DS18B20 dapat ditunjukkan pada Gambar 2.8.



Gambar 2.8. Diagram blok Sensor Suhu DS18B20 *Waterproof*
(Sumber : *Datasheet DS18B20*)

Sensor ini merupakan sensor digital yang menggunakan *1-wire* untuk berkomunikasi dengan mikrokontroler. Keunikan dari sensor ini adalah tiap sensor memiliki kode serial yang memungkinkan untuk penggunaan DS18B20 lebih dari satu dalam satu komunikasi *1-wire*.

Spesifikasi sensor suhu DS18B20:

1. Unik *1-Wire*® *interface* hanya memerlukan satu pin port untuk komunikasi secara *1-Wire*.
2. Setiap perangkat memiliki kode serial 64-bit yang disimpan dalam sebuah ROM onboard.
3. Tidak memerlukan ada komponen tambahan.
4. Bekerja pada kisaran tegangan 3 sampai 5,5V.
5. Dapat mengukur suhu pada kisaran -55 sampai 125 °C.
6. Akurasi $\pm 0,5^\circ\text{C}$ akurasi dari suhu -10 sampai 85 °C.
7. Resolusi dapat diprogram antara 9 sampai 12 bit.
8. Kecepatan mengkonversi suhu maksimal 750 ms.

Gambar sensor suhu DS18B20 *Waterproof* dapat ditunjukkan pada Gambar 2.9.



Gambar 2.9 Sensor Suhu DS18B20 *Waterproof*
(Sumber : www.pihomeserver.fr)

2.11 Hidrometer

Semakin besar zat padat yang terlarut dalam air, maka densitas larutan tersebut juga semakin besar. Hidrometer adalah alat yang digunakan untuk mengukur masa jenis relative (*specific gravity*) suatu cairan, yaitu rasio kepadatan atau densitas suatu fluida dengan densitas fluida standar (*reference*). Densitas fluida standar untuk zat cair adalah air dengan densitas 1 g/cm^3 atau 1000 kg/m^3 . *Specific Gravity (SG)* dinyatakan dalam angka 4 digit dielakang koma dan tidak memiliki satuan.

$$SG = \text{Densitas fluida} / \text{Densitas Fluida Standar} \quad (2-8)$$

Hidrometer merupakan salah satu alat ukur yang mengaplikasikan hukum Archimedes. Menurut hukum Archimedes, jika suatu benda dicelupkan kedalam zat cair, maka benda tersebut akan mendapatkan gaya keatas sebanding dengan volume cairan yang dipindahkan oleh benda tersebut. Secara matematis, dapat dituliskan :

$$F_A = \rho_c V_c g \quad (2-9)$$

Dengan :

$$F_A = \text{Gaya Keatas (N)}$$

$$\rho_c = \text{Masa jenis zat cair (kgm}^{-3}\text{)}$$

$$V_c = \text{Volume zat cair yang dipindahkan oleh benda (m}^3\text{)}$$

$$g = \text{Percepatan gravitasi (ms}^{-2}\text{)}$$

Gambar Hidrometer celup dapat ditunjukkan pada Gambar 2.10.



Gambar 2.10 Hidrometer celup
(Sumber: www.aliexpress.com/)

2.12 Pompa Air 6V DC

Pompa air adalah alat atau mesin yang digunakan untuk memompa atau mengalirkan fluida. Pada mesin pompa air ada saluran hisap dan ada saluran buang. ketika pompa dihidupkan atau dihubungkan dengan tegangan jala-jala, maka pompa akan berputar sehingga dibagian dalam pompa terjadi *vaccum* karena adanya perbedaan tekanan, sehingga air yang ada didalam tanah akan terhisap naik. Pompa air DC ditunjukkan dalam gambar 2.11.



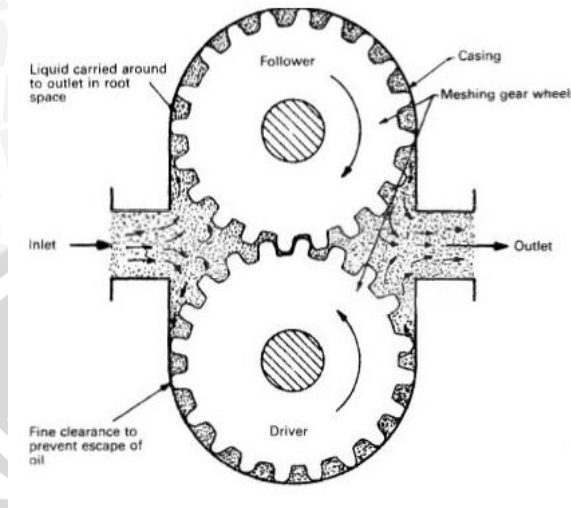
Gambar 2.11. Pompa air DC
(Sumber : www.ifuturetech.org)

Mesin pompa air pada dasarnya terdiri dari 2 jenis menurut cara kerja dan rancangannya. Jenis pompa tersebut adalah sebagai berikut :

1. Pompa Sistem Rotasi.

Sebuah pompa sistem rotasi daam perakitannya terdiri dari dua roda gigi, gigi yang terhubung dengan motor dan roda gigi yang dikopel dengan roda gigi yang lain. Ketika motor dijalankan, roda gigi akan berputar dan membuat fluida masuk melalui

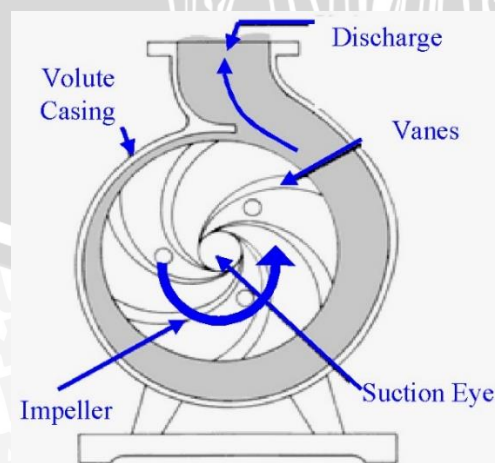
inlet pompa dan keluar melalui *outlet* pompa tersebut. Rotasi pompa terus menerus, gerakan dalam rongga terus berlangsung, sehingga memindahkan cairan di selama motor terus berputar. Prinsip kerja dari pompa sistem rotasi ini ditunjukkan dalam gambar 2.12.



Gambar 2.12. Pompa Sistem Rotasi
(Sumber : maritime523.rssing.com)

2. Pompa Sistem Sentrifugal.

Secara prinsip terdiri dari *casing* pompa dan impeler yang terpasang pada poros putar. *Casing* pompa berfungsi sebagai pelindung, batas tekan dan juga terdiri dari saluran- saluran yang untuk masukan (*suction*) dan keluaran (*discharge*). *Casing* ini memiliki *vent* dan *drain* yang berguna untuk melepas udara atau gas yang terjebak dalam casing selain untuk juga berguna perawatannya. Gambar sistem pompa sentrifugal dapat ditunjukkan pada Gambar 2.13.



Gambar 2.13. Pompa Sistem Sentrifugal
(Sumber : www.quora.com)

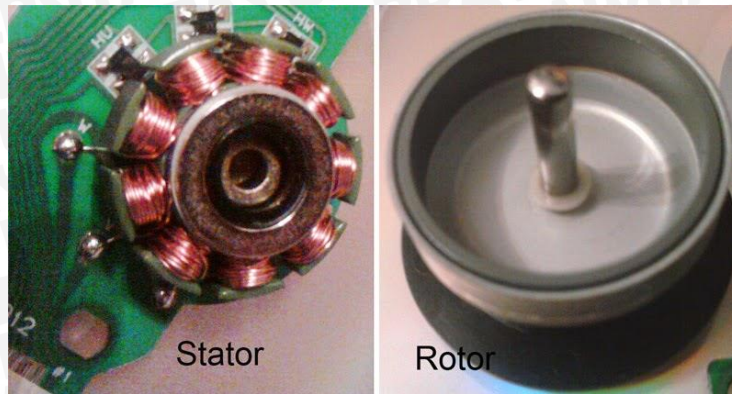
2.13 Kipas 12V DC

Kipas 12V DC atau *fan* adalah suatu alat elektronik yang berfungsi sebagai pendingin agar alat elektronik yang lain tidak *overheating*. Prinsip kerja Kipas 12V DC sama seperti motor DC tanpa kipas (*brushless DC Motor*). Brushless DC motor telah digunakan secara komersial sejak Tahun 1886. Motor BLDC mulai populer pada tahun 1992. Keterbatasan motor DC (menggunakan sikat sebagai media hantar arus dari sumber ke komutator), telah diatasi oleh motor BLDC ini, penggunaan sumber DC untuk pengontrolan, dan tanpa adanya brush adalah ciri utama dari motor BLDC. BLDC motor menghasilkan torsi maksimal saat putaran awal, selanjutnya torsi akan menurun seiring dengan bertambahnya kecepatan motor. Prinsip kerja dari kipas ini pada dasarnya kipas 12V berputar disebabkan oleh adanya arus listrik yang mengalir pada kumparannya. Arus listrik ini kemudian menciptakan medan elektromagnet sehingga terjadi gaya tarik dan gaya tolak di antara magnet-magnet stator kipas 12V. Kipas 12V DC ini ditunjukkan pada gambar Gambar 2.14.



Gambar 2.14. Kipas 12V DC
(Sumber : www.dhresource.com)

BLDC motor adalah suatu jenis motor sinkron. Artinya medan magnet yang dihasilkan oleh stator dan medan magnet rotor berputar di frekuensi dan kecepatan yang sama. BLDC motor tidak mengalami slip, tidak seperti yang terjadi pada motor induksi biasa. BLDC Motor merupakan sebuah perangkat elektromagnetis yang mengubah energi listrik menjadi energi mekanik. Energi mekanik ini digunakan untuk memutar impeller pompa, fan atau blower, menggerakkan kompresor, mengangkat bahan, dll. BLDC motor digunakan di industri maupun di rumah, seperti: mixer, bor listrik, kipas angin. Bagian *stator* dan *rotor* dari BLDC motor dapat ditunjukkan pada Gambar 2.15.



Gambar 2.15. Stator dan Rotor *BLDC* Motor
(Sumber : electricalstudyportal.blogspot.co.id)

2.14 Heater

Heater akuarium adalah alat yang berfungsi untuk meningkatkan suhu air pada akuarium. *Heater* yang digunakan untuk akuarium harus disesuaikan dengan volume akuarium. Dalam penelitian ini, *heater* yang digunakan adalah *heater* Kandila KD-75 (75 Watt). Tabel informasi penggunaan *heater* untuk masing-masing ukuran akuarium ditunjukkan pada Tabel 2.4.

Tabel 2.4 Tabel Informasi Penggunaan *Heater*

Ukuran <i>Heater</i>	Volume Akuarium (max)
25 Watt	< 25 Liter
50 Watt	< 50 Liter
75 Watt	< 75 Liter
100 Watt	< 100 Liter
150 Watt	< 150 Liter
200 Watt	< 200 Liter
300 Watt	< 300 Liter

(Sumber: <http://www.aquaone.co.uk/>)

Pada umumnya, *heater* akuarium berbentuk *heater* celup, artinya harus allat dimasukan dalam cairan yang dipanaskan. *Heater* terdiri dari tabung gelas yang didalamnya terdapat elemen pemanas. *Heater* Akuarium dapat ditunjukkan pada Gambar 2.17.



Gambar 2.16. Heater Akuarium
(Sumber: www.amazon.in)

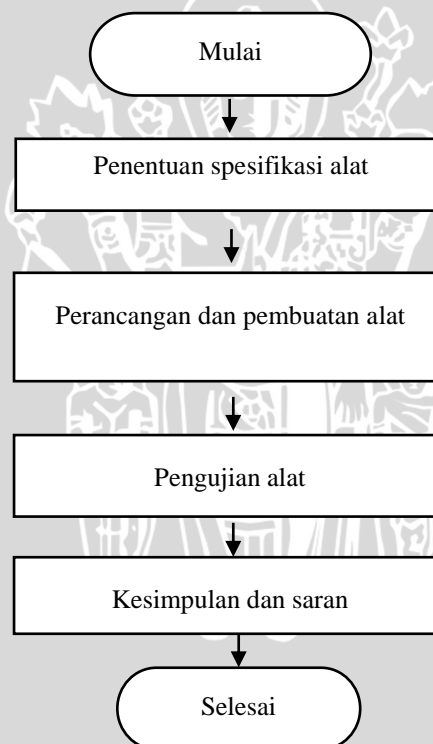




BAB III

METODE PENELITIAN

Pada bab ini akan diuraikan tentang metode penelitian yang akan digunakan dalam proses perancangan dan realisasi alat untuk pengendalian suhu dan salinitas air pada akuarium ikan badut (*Amphiprion percula*) berbasis mikrokontroler arduino due. Hasil luaran yang diinginkan dalam penelitian ini yaitu dapat merealisasikan alat untuk pengendalian suhu dan salinitas air pada akuarium ikan badut (*Amphiprion percula*) yang dapat digunakan untuk pengendalian suhu dan salinitas air pada akuarium sesuai dengan nilai *setpoint* yang diinginkan. Metode yang digunakan secara umum dapat ditunjukkan pada Gambar 3.1.



Gambar 3.1 Diagram alir metode penelitian

3.1 Penentuan Spesifikasi Alat

Spesifikasi komponen-komponen sistem pengendalian salinitas air pada akuarium air laut menggunakan kontrol logika *fuzzy* adalah sebagai berikut :

- 1) Menggunakan akuarium yang berbahan dasar kaca dengan ukuran 40 cm x 35 cm x 35 cm.
- 2) Menggunakan *Volume* air sebanyak 40 L.
- 3) Menggunakan Arduino Due sebagai kontroller.
- 4) Menggunakan sensor suhu DS18B20 *Waterproof* dan sensor salinitas perancangan sendiri.
- 5) Aktuator yang digunakan untuk menurunkan suhu air akuarium adalah 4 buah kipas 5x5 cm 12v dan 1 buah kipas 12x12 cm 12v.
- 6) Aktuator yang digunakan untuk mengalirkan air tawar dan air asin adalah 2 buah pompa air 6V.
- 7) Menggunakan *LCD* 16x2.
- 8) Menggunakan *Aquarium Pump Kandila KD-104* untuk sirkulasi air.
- 9) Sumber tegangan yang digunakan yaitu *Power Supply* 12V, 2A.
- 10) Menggunakan regulator DC-DC *step-down* LM2596.
- 11) Menggunakan *Driver* motor DC L298N.

3.2 Perancangan dan Pembuatan Alat

Perancangan dan realisasi alat dalam penelitian ini dibagi menjadi dua bagian yaitu *hardware*, dan *software*.

3.2.1 Perancangan Mekanik

Perancangan mekanik pada penelitian ini diperlukan untuk membuat desain alat untuk pengendalian suhu dan salinitas air pada akuarium ikan badut (*Amphiprion percula*). Bentuk mekanik secara umum dirancang menggunakan perangkat lunak *Corel Draw*.

3.2.2 Perancangan Perangkat Keras

Perancangan perangkat keras pada penelitian ini diperlukan untuk membuat alat untuk pengendalian suhu dan salinitas air pada akuarium ikan badut (*Amphiprion percula*). Perancangan *hardware* terdiri dari perancangan tangki akuarium, perancangan sensor dan aktuator serta mekanisme dari pengendalian suhu dan salinitas. Bentuk sistem secara umum dirancang menggunakan perangkat lunak *Corel Draw*.

3.2.3 Perancangan Perangkat Lunak

Setelah melakukan perancangan dan pembuatan perangkat keras baru dilakukan perancangan dan pembuatan perangkat lunak. Perangkat lunak akan dibuat menggunakan

software IDE Arduino yang merupakan salah satu IDE yang *open source* untuk membuat *project*, melakukan *editing*, *troubleshooting*, dan *compiling* untuk modul Arduino. Perangkat lunak yang akan dibuat berfungsi untuk memberikan perintah ke aktuator pompa air 6V dan kipas 12V dengan data masukan dari sensor suhu dan sensor salinitas yang digunakan.

Perangkat lunak dirancang dari pembuatan diagram alir (*flowchart*) sistem secara keseluruhan dengan kontroller on/off dan logika *fuzzy*. Kemudian dilakukan penulisan program menggunakan bahasa pemrograman C pada *software* IDE Arduino.

3.3 Pengujian Alat

Untuk menguji alat apakah sudah sesuai dengan yang telah direncanakan atau tidak maka diperlukan pengujian sistem. Pengujian keseluruhan sistem dilakukan dengan menghubungkan blok-blok perangkat keras (*hardware*) dan mengoprasikan sistem kemudian dapat dilakukan analisis apakah perangkat keras sudah bekerja sesuai yang diharapkan. Setelah perangkat keras sudah beroperasi sesuai yang diharapkan, maka perangkat lunak dapat diuji. Apabila perangkat keras dan perangkat lunak bekerja dengan baik dan saling bersinergi maka keseluruhan sistem sudah bekerja sesuai dengan spesifikasi rancangan.

3.3.1 Pengujian Sub Sistem

Pengujian ini dilakukan untuk masing-masing blok pada alat. Pengujian-pengujian yang dilakukan adalah sebagai berikut :

1) Pengujian Sensor Suhu.

Pengujian sensor suhu dilakukan untuk mengetahui apakah sensor berjalan dengan baik dan mengetahui error dari sensor jika dibandingkan dengan termometer analog.

2) Pengujian Sensor Salinitas.

Pengujian sensor salinitas dilakukan untuk mengukur tegangan keluaran terhadap beberapa kadar salinitas berbeda yang terbaca oleh sensor.

3) Pengujian Kipas 12V.

Pengujian kipas dilakukan untuk mengetahui apakah kipas berjalan dengan baik dan mengetahui suhu air maksimum yang dapat diturunkan oleh kipas.

4) Pengujian Pompa Air 6V.

Pengujian pompa air dilakukan untuk mengetahui apakah pompa air berjalan dengan baik.

5) Pengujian Mikrokontroller.

Pengujian mikrokontroller bertujuan untuk mengetahui apakah mikrokontroller sudah dapat mengolah data dan menjalankan instruksi berupa kode program yang telah didefinisikan atau tidak.

3.3.2 Pengujian Keseluruhan Sistem

Pengujian keseluruhan sistem dilakukan dengan menghubungkan blok-blok sistem yang sesuai dengan diagram blok sistem dan memasukkan program kedalam mikrokontroller untuk mengontrol *hardware* dari keseluruhan sistem yang telah dibuat. Sistem akan berjalan dengan baik apabila sudah berjalan sesuai dengan *flowchart* yang telah dibuat.

3.4 Pengambilan Kesimpulan dan Saran

Kesimpulan didapat berdasarkan dari perealisasiian dan pengujian alat pengendalian salinitas air pada akuarium air laut menggunakan kontrol logika *fuzzy* sesuai dengan tujuan dan rumusan masalah. Saran diberikan setelah melihat adanya kekurangan dalam sistem yang telah dibuat, dengan harapan nantinya alat ini dapat dikembangkan dengan baik.



BAB IV

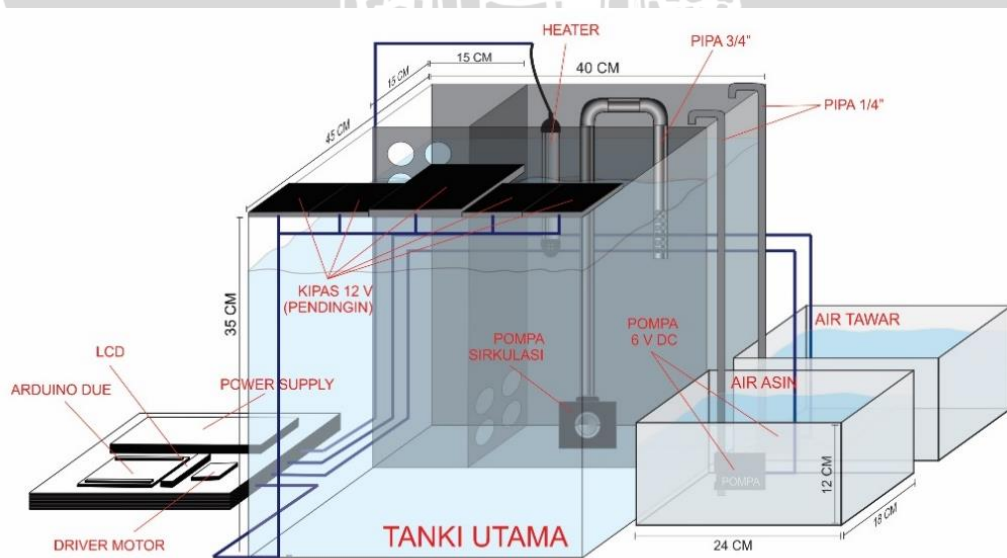
PERANCANGAN, PENGUJIAN DAN ANALISIS

4.1 Perancangan dan Pembuatan Alat

Penyusunan penelitian ini berkaitan dengan perancangan dan pembuatan alat untuk pengendalian suhu dan salinitas air pada akuarium ikan badut (*Amphiprion percula*) yang didasarkan pada rumusan masalah yang telah di rumuskan. Perancangan dilakukan untuk merealisasikan alat yang telah direncanakan dengan langkah-langkah sebagai berikut yaitu perancangan mekanik, perancangan perangkat keras (*hardware*), dan perancangan perangkat lunak (*software*).

4.1.1 Perancangan Mekanik

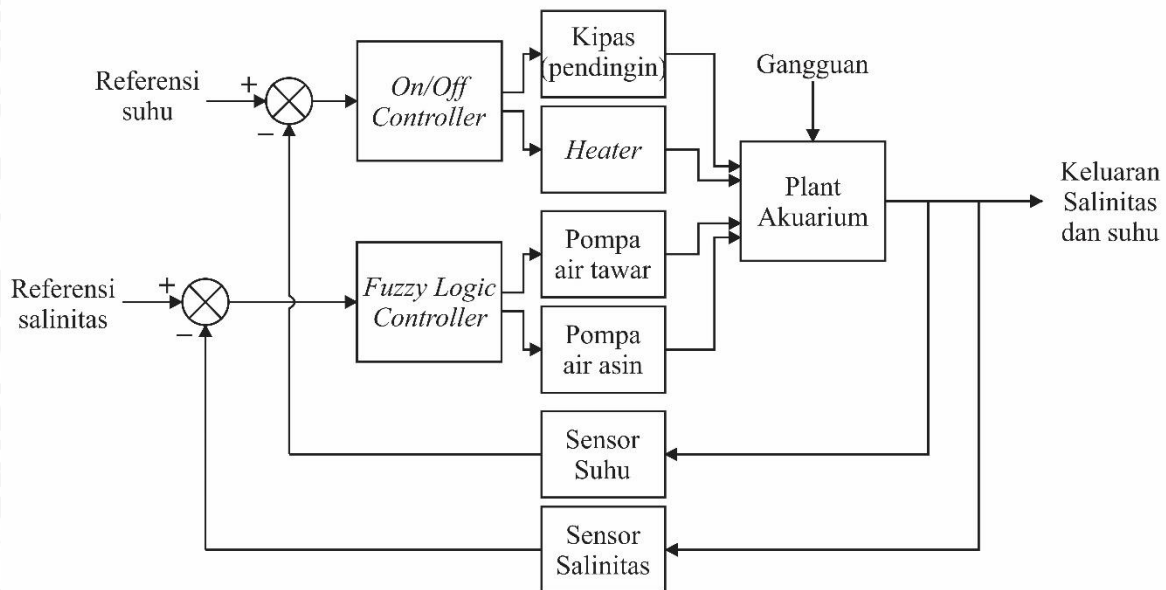
Perancangan mekanik alat pengendalian suhu dan salinitas air pada akuarium ikan badut (*Amphiprion percula*) meliputi desain alat secara keseluruhan dengan menggunakan *software Corel Draw*. Perancangan menggunakan bahan – bahan kaca dengan tebal 5mm untuk tangki utama akuarium, 2 tangki plastik ukuran 24cm x 18cm x 12cm, kontroler utama serta sensor dan aktuator yang diperlukan. Perancangan sistem pengendalian suhu dan salinitas air pada akuarium ikan badut (*Amphiprion percula*) dapat ditunjukkan pada Gambar 4.1.



Gambar 4.1 Desain Alat Pengendalian Suhu dan Salinitas.

4.1.2 Perancangan Perangkat Keras

Perancangan perangkat keras lebih ditekankan kepada perancangan antarmuka dari masing-masing blok seperti antarmuka sensor dengan mikrokontroler dan antarmuka aktuator dengan mikrokontroler. Selain itu, dilakukan juga perancangan *main board* kontroler yang digunakan sebagai inti yang menjalankan operasi dan aksi pengontrolan. Diagram blok sistem secara keseluruhan dapat ditunjukkan pada Gambar 4.2.

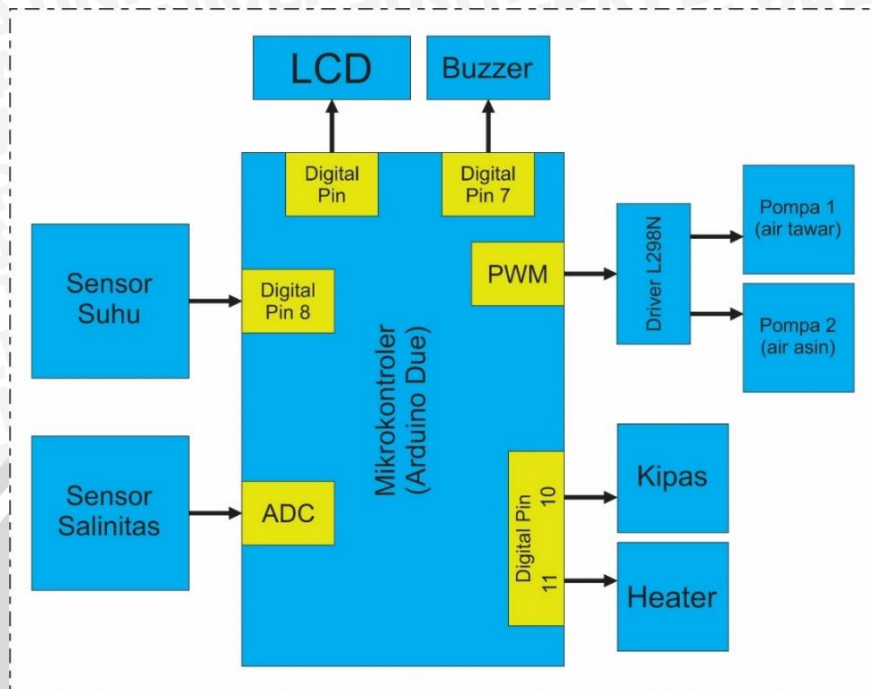


Gambar 4.2. Diagram Blok Sistem Pengendalian Suhu dan Salinitas.

Diagram blok sistem secara garis besar terbagi dalam dua bagian utama, yaitu *loop* dalam pengontrolan parameter salinitas dan *loop* untuk pengontrolan parameter suhu. Kontroler yang digunakan untuk pengontrolan suhu menggunakan metode *on/off controller*, sedangkan untuk pengontrolan salinitas air menggunakan metode *fuzzy logic controller*. Pemilihan metode tersebut didasarkan atas karakteristik masing-masing parameter.

Pemilihan *on/off controller* untuk pengendalian suhu disebabkan karakteristik suhu yang perubahannya yang lambat, sehingga *on/off controller* dirasa paling optimum dalam aplikasi pengendalian suhu. Untuk metode pengendalian salinitas air dalam akuarium menggunakan metode *fuzzy logic controller*. Pemilihan *fuzzy logic controller* sebagai kontroler utama dalam pengendalian salinitas air didasarkan pada karakteristik dari perubahan salinitas terhadap waktu yang sangat lambat karena faktor penguapan yang mempengaruhi salinitas air berubah dengan lambat. Selain itu, pengontrolan salinitas pada akuarium ikan badut ini lebih baik jika perubahan yang terjadi setelah aksi kontrol dilakukan

tidak derastis atau dalam kata lain *raise time* (T_r) tidak terlalu cepat dan curam. Diagram blok alat secara keseluruhan dapat ditunjukkan pada Gambar 4.3.



Gambar 4.3 Diagram Blok Alat Pengendalian Suhu dan Salinitas.

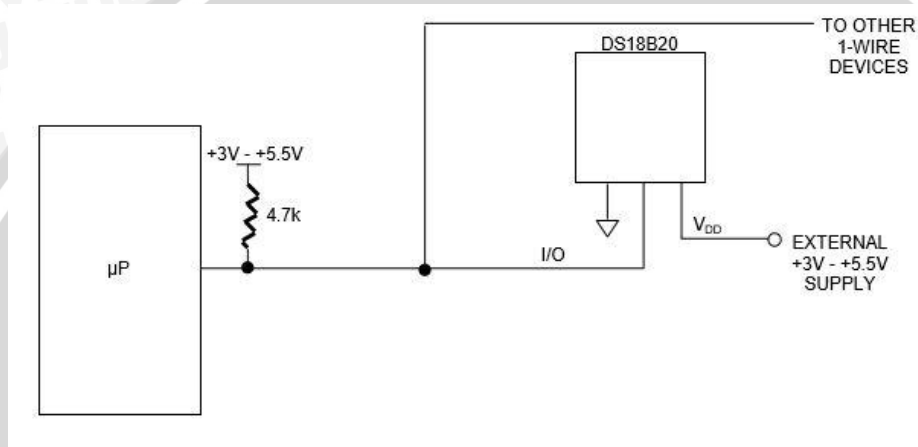
Diagram blok alat terdiri dari 2 input atau *setpoint*. Input yang pertama yaitu input referensi suhu yang masuk dalam pin *digital* (pin 8) mikrokontroler (Arduino Due) sebagai kontroler utama pada sistem dengan menggunakan sensor suhu digital DS18B20 *waterproof*. Setelah itu, mikrokontroler akan memproses sinyal masukan dan akan mendapatkan sinyal *error*. Dengan kontroler on/off, mikrokontroler akan memberikan aksi kontrol berupa menghidupkan atau mematikan aktuator *heater* atau kipas melalui pin *digital* (pin 10 dan 11) tergantung pada keadaan suhu yang terbaca sensor.

Input yang kedua yaitu input referensi suhu yang masuk dalam pin analog (A7). Untuk mengetahui keadaan salinitas air dilakukan kalibrasi terhadap sensor. Tegangan operasi mikrokontroler Arduino due adalah sebesar 3,3 V sehingga semua sensor dalam sistem ini diberikan tegangan masukan sebesar 3,3 V. Setelah input tegangan masuk, selanjutnya nilai masukan akan diubah kedalam bentuk digital menggunakan ADC (*Analog Digital Converter*). Nilai yang masuk selanjutnya diproses oleh mikrokontroler dengan mencari sinyal *error* yang kemudian diproses kembali dalam kontroler logika *fuzzy* yang telah dibuat. mikrokontroler kemudian akan memberikan aksi kontrol berupa memberikan keluaran analog pada aktuator (Pompa 1 dan Pompa 2) tergantung hasil dari pembacaan

sensor dan proses kontroler pada mikrokontroler. Keluaran analog direpresentasikan dengan keluaran PWM (*Pulse Width Modulation*) dengan resolusi 10 bit pada pin Arduino due.

4.1.2.1 Perancangan Rangkaian Sensor Suhu

Sensor suhu DS18B20 *Waterproof* merupakan alat yang dapat membaca keadaan perubahan suhu disekitar sensor tersebut. Sensor suhu DS18B20 *Waterproof* ini termasuk sensor suhu *digital* yang dapat membaca perubahan suhu dari $-55\text{ }^{\circ}\text{C}$ sampai dengan $125\text{ }^{\circ}\text{C}$ yang dapat diprogram resolusi pembacaannya dari 9 hingga 12 bit. Rangkaian penggunaan sensor suhu DS18B20 *Waterproof* ditunjukkan pada Gambar 4.4.



Gambar 4.4. Rangkaian penggunaan sensor suhu DS18B20 *Waterproof*.

Penggunaan resistor 4,7 k Ω digunakan sebagai resistor *pullup* untuk suplai sensor DS18B20 pada saat konversi suhu. Dengan resolusi 9, 10, 11 atau 12 bit pembacaan suhu secara berurutan dengan resolusi suhu 0.5°C , 0.25°C , 0.125°C , dan 0.0625°C . Secara *default*, pembacaan sensor suhu DS18B20 di-*setting* dengan resolusi 12 bit.

4.1.2.2 Perancangan Rangkaian Sensor Salinitas

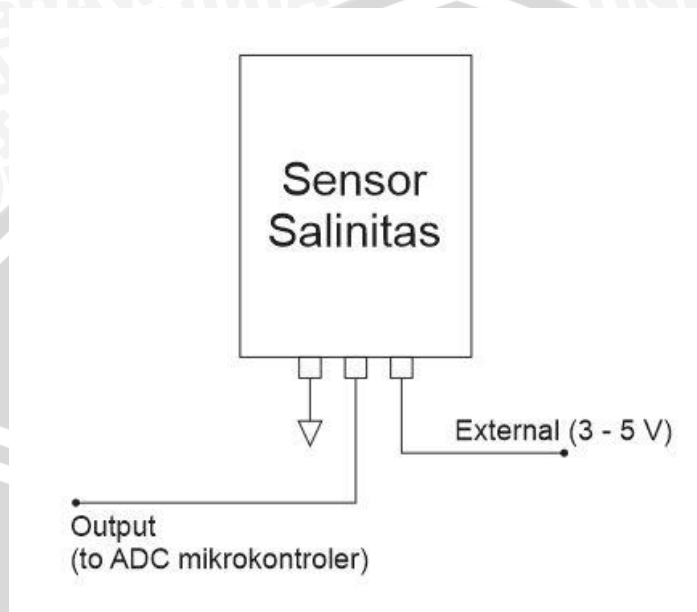
Sensor salinitas adalah sensor yang mendeteksi perubahan salinitas (kadar garam) dalam larutan. Pada umumnya sensor yang digunakan adalah sensor *TDS* (*Total Dissolve Solid*) yang artinya sensor tersebut mengukur total zat padat yang terlarut dalam suatu larutan. Dengan mengukur konduktifitasnya, kadar garam yang terkandung dalam suatu larutan dapat diketahui. Selain itu, refraktometer dan hidrometer juga dapat mengukur salinitas secara tidak langsung dengan mengukur SG (*Specific Gravity*) dari suatu cairan atau larutan.

Pada penelitian ini, sensor yang digunakan adalah hidrometer yang dimodifikasi dengan menambahkan mangnet kecil dibagian atas hidrometer, lalu dibagian atas diletakan sensor *hall effect* yang akan membaca perubahan posisi hidrometer. Semakin besar *TDS* (*Total Dissolve Solid*) atau dalam penelitian ini adalah kadar garam yang terlarut maka densitas larutan akan semakin besar sehingga akan memberikan gaya angkat pada hidrometer yang semakin besar pula. Sensor *hall effect* digunakan karena agar meminimalisir modifikasi pada hidrometer dengan hanya menempelkan magnet bulat kecil berdiameter 1 cm dan dengan ketebalan 1 mm sehingga sehingga tidak terlalu membebani hidrometer. Selanjutnya pada bagian kanan dan kiri bawah hidrometer ditempelkan 2 buah gabus agar membantu memberikan gaya angkat pada hidrometer sehingga bagian atas hidrometer tetap dapat mengapung dan tidak tenggelam. Selanjutnya sensor *hall effect* dietakan pada bagian atas hidrometer yang telah ditempelkan magnet. Naik turunnya hidrometer akan menaikan dan menurunkan juga magnet yang telah ditempelkan, kemudian sensor *hall effect* akan membaca perubahan medan magnet yang ada pada sekitar *gap* antara sensor *hall effect* dan magnet yang tertempel pada hidrometer. Selanjutnya, hidrometer dan sensor *hall effect* dimasukan kedalam pipa berdiameter $\frac{3}{4}$ inchi dengan tinggi 27 cm yang telah dilubangi sebagian dari tinggginya. Gabus diberikan pada pipa agar sensor *hall effect* tidak terkena cairan yang akan diukur dan agar pembacaan hidrometer tepat mengikuti permukaan air sebagai referensi pengukurannya. Pemberat pada bagian bawah pipa agar posisi sensor tetap pada posisi vertikal. Gambar sensor hasil perancangan dapat ditunjukkan pada Gambar 4.5.



Gambar 4.5 Hasil Perancangan Sensor Salinitas Berbasis Sensor *hall effect*.

Sensor *hall effect* memiliki 3 pin utama, yaitu 2 pin untuk *power supply*, dan 1 pin untuk keluaran sensor. Sensor dicatu dengan tegangan 3,3V (Tegangan operasi mikrokontroler Arduino Due), lalu keluaran dihubungkan pada pin analog Arduino Due. Dengan memanfaatkan ADC, masukan analog kemudian diubah dalam bentuk digital dan selanjutnya data diproses oleh mikrokontroler. Gambar konfigurasi pin sensor salinitas dapat ditunjukkan pada Gambar 4.6.



Gambar 4.6 Konfigurasi Pin Sensor Salinitas Berbasis Sensor *hall effect*.

ADC yang digunakan dalam perancangan ini yaitu ADC dengan resolusi 10-bit sehingga nilai sampling maksimal yaitu 1024. Proses sampling dari ADC dapat diketahui melalui persamaan (4-1) berikut :

$$ADC = \frac{V_{out\ sensor}}{Tegangan\ refrensi} \times 2^{n-1} \quad (4-1)$$

Tegangan refrensi yang digunakan yaitu 3,3 volt karena tegangan maksimal yang dapat dikonversi oleh mikrokontroler Arduino Due menjadi nilai ADC adalah 3,3 volt. Pada Persamaan (4-1), n merupakan jumlah bit yang digunakan, karena bit yang digunakan yaitu 10 bit maka n sama dengan 10 sehingga 2^{n-1} sama dengan 1023. Maka persamaannya menjadi :

$$ADC = \frac{V_{out\ sensor}}{3,3} \times 1023 \quad (4-2)$$

Vout sensor merupakan keluaran dari rangkaian sensor *hall effect* berupa tegangan analog. Nilai *n* dapat berubah sesuai dengan jumlah bit yang digunakan. Untuk mengetahui resolusi dari ADC sendiri dapat diketahui melalui persamaan (4-3) berikut :

$$\text{Resolusi} = \frac{\text{tegangan referensi}}{2^n - 1} \quad (4-3)$$

Untuk tegangan referensi keluaran arduino yang telah diukur menggunakan votmeter yang digunakan dalam penelitian ini yaitu sebesar 3,3 V, Jadi resolusi ADC sebesar :

$$\text{Resolusi} = \frac{3,3}{2^{10} - 1}$$

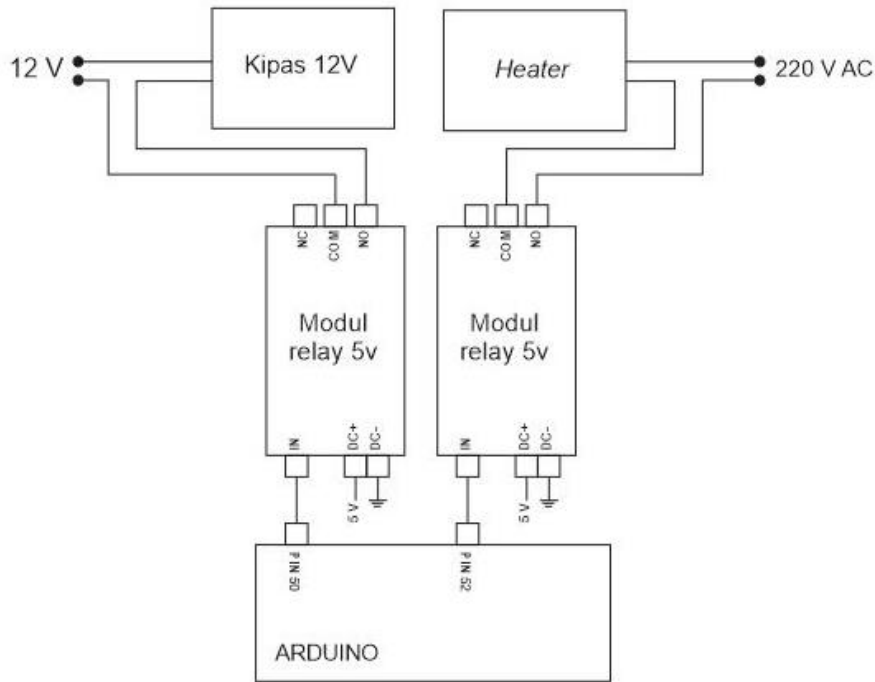
$$\text{Resolusi} = \frac{3,3}{1023}$$

$$\text{Resolusi} = 3,2226 \text{ mV}$$

Dengan resolusi 3,2226 mV artinya setiap kenaikan 3,2226 mV pada keluaran sensor, nilai ADC yang terbaca oleh mikrokontroler akan mengalami kenaikan sebesar 1 nilai.

4.1.2.3 Perancangan Rangkaian Kipas dan Heater

Karena tegangan operasi dari Arduino Due sebesar 3,3 V, tidak memungkinkan untuk mengontrol kipas 12 V dan *heater* yang membutuhkan tegangan AC 220V. Oleh karena itu, dalam penelitian ini digunakan modul relay untuk mengisolasi pin – pin Arduino yang tegangan operasinya lebih kecil dari aktuator yang digunakan (kipas dan *heater*). Relay 1 dikontrol dengan mikrokontroler Arduino melalui pin 50 pada pin Arduino yang akan mengaktifkan kipas 12 V jika logika pin 50 pada Arduino berlogika *HIGH* ($\pm 3,3V$) dan akan mematikan kipas 12 V jika logika pin 50 pada Arduino berlogika *LOW* ($\pm 0V$). Sedangkan relay 2 dikontrol dengan mikrokontroler Arduino melalui pin 52 pada pin Arduino yang akan mengaktifkan *heater* jika logika pin 52 pada Arduino berlogika *HIGH* ($\pm 3,3V$) dan akan mematikan *heater* jika logika pin 52 pada Arduino berlogika *LOW* ($\pm 0V$). Gambar konfigurasi pin pada Arduino, kipas dan *heater* dapat ditunjukkan pada Gambar 4.7.

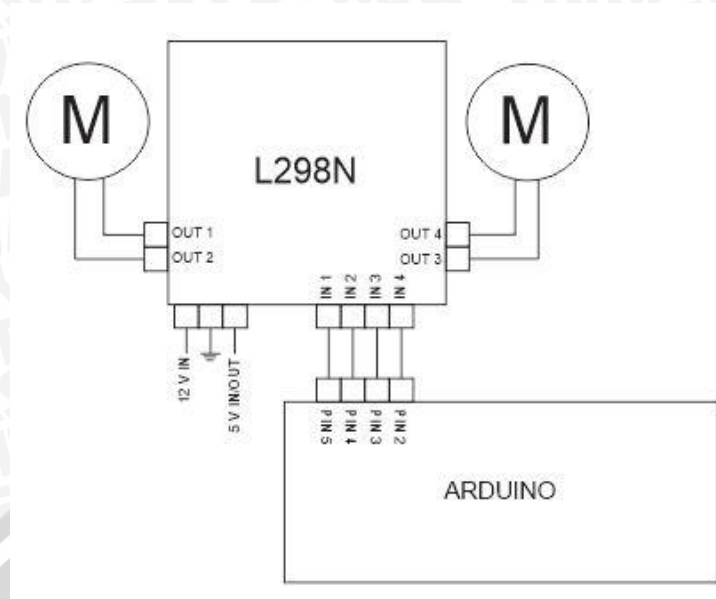


Gambar 4.7 Konfigurasi Pin Arduino dan aktuator (kipas dan *heater*)

4.1.2.4 Perancangan Rangkaian *Driver* Pompa Motor DC

Driver motor adalah alat elektronik yang berfungsi sebagai pengatur kecepatan motor DC. Pada penelitian ini *driver* yang digunakan adalah modul *driver dual motor controller L298N H-Bridge*. Dengan modul *driver* ini, 2 motor DC sekaligus dapat dikontrol dengan menggunakan PWM (*Pulse Width Modulation*) pada mikrokontroler.

Pada penelitian ini, pin yang digunakan pada mikrokontroler Arduino Due adalah pin 2, 3, 4, dan 5 yang dapat menghasilkan sinyal PWM yang diinginkan. *Out1* dan *out2* pada modul *driver L298N* dihubungkan dengan pompa 1 (untuk air tawar), sedangkan *Out3* dan *out4* pada modul *driver L298N* dihubungkan dengan pompa 2 (untuk air asin), dan catu daya sebesar 12 V untuk *power supply* modul *driver L298N*. Gambar konfigurasi pin pada Arduino dan modul *driver L298N* dapat ditunjukkan pada gambar 4.8.



Gambar 4.8 Konfigurasi Pin Arduino dan *driver* L928N

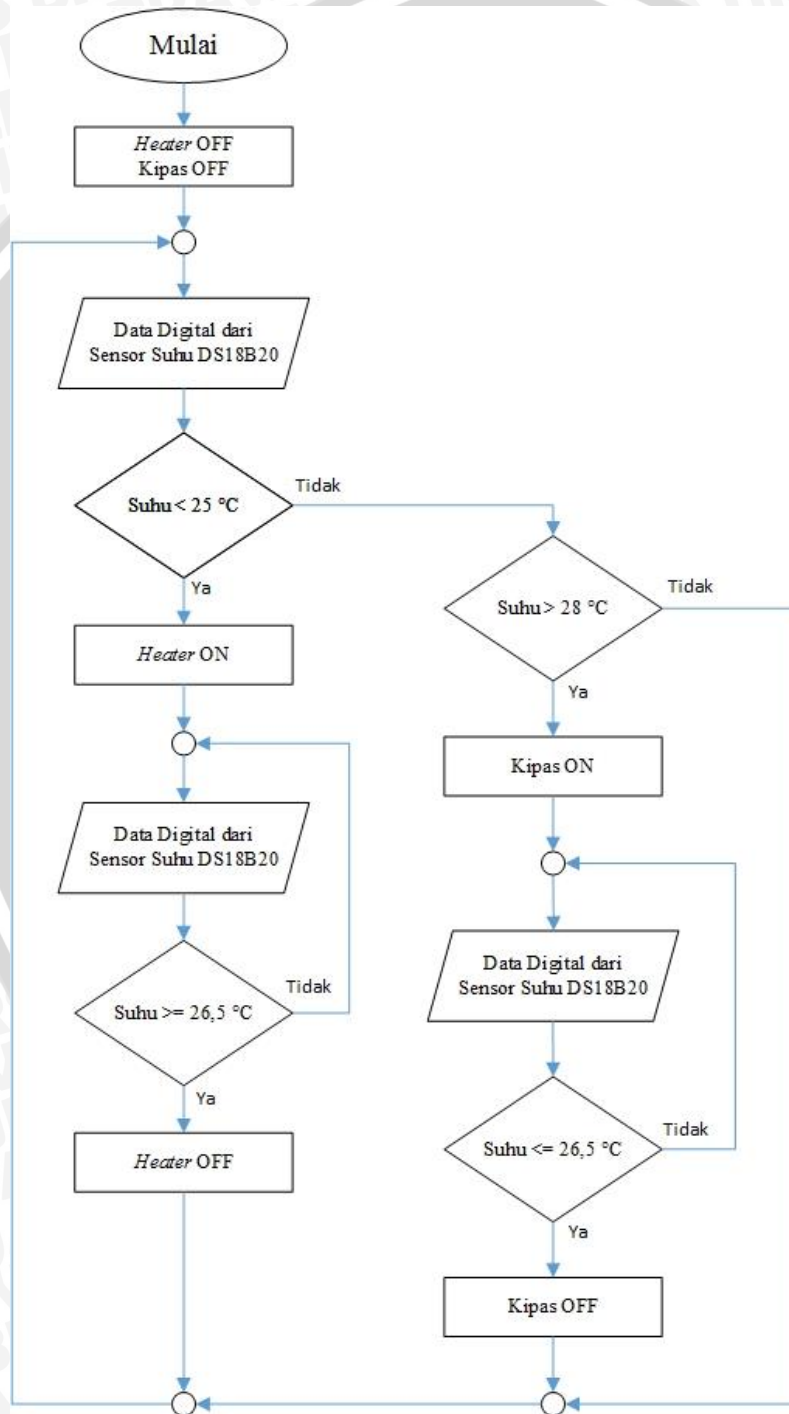
4.1.3 Perancangan Perangkat Lunak

Perancangan ini dilakukan untuk mengimplemetasikan secara terprogram setiap sub-sub diagram blok yang telah dirancang pada Gambar 4.3. Perancangan perangkat lunak secara keseluruhan dibagi menjadi dua bagian yaitu perangkat lunak untuk kontrol suhu dengan kontroler on/off dan untuk kontrol salinitas dengan kontroler *fuzzy logic*. Setelah penentuan spesifikasi sistem, maka tahap selanjutnya yaitu pengembangan perangkat lunak yang merupakan proses pengubahan spesifikasi sistem menjadi sistem yang dapat dijalankan. Tahap ini yaitu mencangkup perancangan dan pemrograman perangkat lunak. Perancangan perangkat lunak merupakan deskripsi struktur yang akan diimplemetasikan berupa data-data yang diolah dalam sistem, *interface* antar komponen elektronik pendukung, dan penentuan algoritma yang tepat dalam sistem. Proses perancangan perangkat lunak secara umum akan membahas tiga hal yaitu :

1. Perancangan *interface*, yaitu perancangan komunikasi antara mikrokontroler dengan sub-sub sistem seperti sensor, aktuator, dan alat transmisi data.
2. Perancangan struktur data, yaitu proses pengolahan data secara terprogram dengan melakukan perhitungan data yang didapatkan dari sensor agar dapat dirubah menjadi nilai suhu dan salinitas.
3. Perancangan algoritma, yaitu membuat program yang sesuai dengan kontroler yang digunakan.

4.1.3.1 Perancangan Perangkat Lunak Kontrol Suhu

Perancangan perangkat lunak kontrol suhu secara garis besar meliputi pengenalan port yang akan digunakan dengan melakukan spesifikasi port tersebut sesuai dengan kebutuhan yang diinginkan misalnya port *I/O* dan port *serial* untuk *log* data suhu yang terbaca oleh sensor ke komputer. Diagram alir perangkat lunak kontrol suhu secara lengkap dapat ditunjukkan pada Gambar 4.9.



Gambar 4.9 Diagram alir perangkat lunak kontrol suhu

Pada Gambar 4.7 secara umum keluaran data suhu dari sensor DS12B20 berbentuk digital sehingga masukan berbentuk sinyal digital dapat digunakan *peripheral* pin digital pada Arduino due. Pertama kali, aktuator kipas dan *heater* dimatikan terlebih dahulu dengan memberikan logika *low* pada pin yang terhubung dengan relay untuk kipas dan *heater*. Selanjutnya, data suhu yang didapatkan dari sensor diolah oleh mikrokontroler. *Setpoint* suhu yang diinginkan adalah sebesar $25\text{ }^{\circ}\text{C} - 28\text{ }^{\circ}\text{C}$. Ketika suhu air kurang dari sama dengan $25\text{ }^{\circ}\text{C}$, *heater* akan aktif hingga suhu naik menjadi $26,5\text{ }^{\circ}\text{C}$. sebaliknya, ketika suhu air mencapai lebih dari sama dengan $28\text{ }^{\circ}\text{C}$, kipas akan aktif hingga suhu air turun menjadi $26,5\text{ }^{\circ}\text{C}$. Suhu $26,5\text{ }^{\circ}\text{C}$ didapatkan dari nilai tengah *setpoint* suhu yang diinginkan.

4.1.3.2 Perancangan Perangkat Lunak Kontrol Salinitas

Perancangan perangkat lunak kontrol salinitas air pada penelitian ini menggunakan logika *fuzzy* dalam menjalankan proses kontrol. Dalam perancangan perangkat lunak disini, digunakan program Matlab sebagai pembantu dalam merancang kontrol logika *fuzzy* dengan *tools* yang telah disediakan. Dengan program Matlab, perancang juga dapat melihat hasil *surface* dan mensimulasikan aturan *fuzzy* atau *rule* yang telah dibuat. Pada bab sebelumnya telah dibahas tentang proses perancangan kontroler dengan menerapkan logika *fuzzy*, yaitu terdiri dari :

1) Fuzzifikasi

Proses fuzzifikasi merupakan proses untuk mengubah variabel non fuzzy (variabel numerik) menjadi variabel fuzzy (variabel linguistik) dalam bentuk derajat keanggotaan atau fungsi keanggotaan masukan dan keluaran.

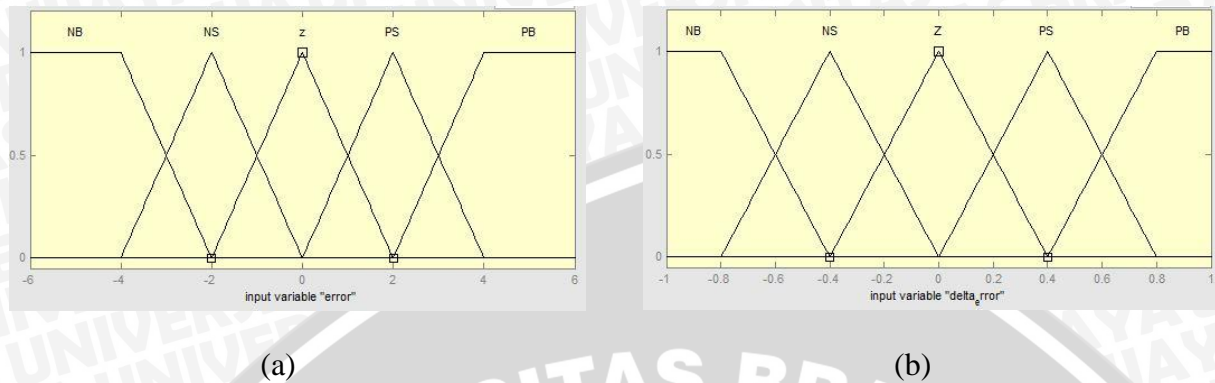
a. Fungsi Keanggotaan Masukan

Dalam derajat keanggotaan masukan, Terdapat dua variabel yang akan diproses dalam fuzzifikasi ini, yaitu *error* dan *delta error* dari keluaran sensor salinitas. *Error* adalah selisih antara nilai keluaran terhadap *setpoint* yang diinginkan, *delta error* adalah selisih antara *error* sekarang dengan *error* sebelumnya. Pada masing – masing variabel, terdiri dari 5 label, yaitu *Negative Big* (NB), *Negative Small* (NS), *Zero* (Z), *Positive Small* (PS) dan *Positive Big* (PB).

Banyaknya fungsi keanggotaan tergantung dari sistem dan respon yang kita inginkan. Semakin banyak label fungsi keanggotaan, maka semakin banyak pula variasi aksi kontrol

yang akan terjadi, dan tentunya semakin presisi atau *error steady state* dapat lebih kecil.

Fungsi keanggotaan masukan ditunjukkan dalam Gambar 4.10.

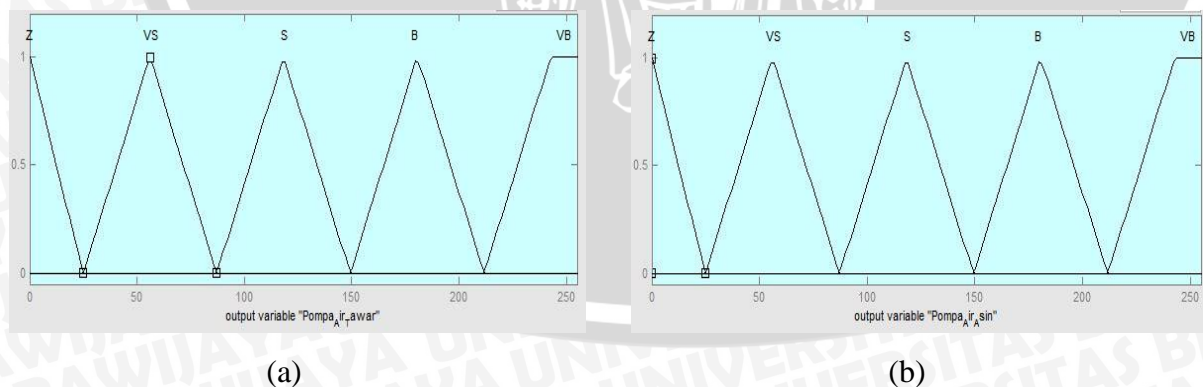


Gambar 4.10 (a) Fungsi keanggotaan *error*

(b) Fungsi Keanggotaan *delta error*

b. Fungsi Keanggotaan Keluaran

Dalam derajat keanggotaan keluaran, terdapat dua variabel yang akan diproses dalam fuzzifikasi, yaitu derajat keanggotaan pompa air tawar dan pompa air asin. Pada masing – masing variabel, terdiri dari 5 label, yaitu *Zero (Z)*, *Very Small (VS)*, *Small (S)*, *Big (B)* dan *Very Big (VB)*. Range nilai dari keluaran adalah range nilai dari pwm mikrokontroler dengan resolusi 8 bit, artinya dengan nilai yang memiliki range 0 – 255 sebanding dengan pwm dengan *duty cycle* dari 0 – 100 %. Gambar dari fungsi keanggotaan keluaran ditunjukkan dalam Gambar 4.11.



Gambar 4.11 (a) Fungsi keanggotaan Pompa Air Tawar

(b) Fungsi Keanggotaan Pompa Air Asin

2) Menentukan aturan *fuzzy*

Aturan *fuzzy* atau *rule* adalah aturan yang menentukan nilai keluaran setelah fuzzifikasi yang akan diproses pada defuzzifikasi yang dalam penentuan aturan ini cenderung tergantung pada *plant* dan pengalaman dari perancang. Jumlah aturan *fuzzy* yang digunakan sebanyak 25 aturan atau *rule*. Aturan *fuzzy* diperoleh dari 5 fungsi keanggotaan masukan *error* dan *delta error*. Aturan *fuzzy* yang digunakan ditunjukkan dalam Tabel 4.1 dan Tabel 4.2.

Tabel 4.1 Aturan *fuzzy* untuk pompa air tawar

		Error				
		NB	NS	Z	PS	PB
Delta Error	NB	Z	Z	Z	S	VB
	NS	Z	Z	Z	S	B
	Z	Z	Z	Z	S	B
	PS	Z	Z	Z	Z	Z
	PB	Z	Z	Z	Z	Z

Tabel 4.2 Aturan *fuzzy* untuk pompa air asin

		Error				
		NB	NS	Z	PS	PB
Delta Error	NB	Z	Z	Z	Z	Z
	NS	Z	Z	Z	Z	Z
	Z	B	S	Z	Z	Z
	PS	B	S	Z	Z	Z
	PB	VB	S	Z	Z	Z

3) Menentukan Inferensi *fuzzy* Dengan Metode Inferensi Mamdani

Inferensi *fuzzy* Dengan Metode Inferensi Mamdani dikenal juga dengan metode min – max. Inferensi adalah proses mendapatkan keluaran dari *IF-THEN rule*. Selain itu ada juga operasi *and*, *or* dan *not* yang digunakan untuk koneksi antara variabel input atau variabel output. Pada proses inferensi ini, *rule* yang dibuat akan dievaluasi dan didapatkan suatu daerah output.

4) Defuzzifikasi

Defuzzifikasi merupakan proses untuk mengubah keluaran *fuzzy* menjadi keluaran *crisp*. Hasil defuzzifikasi digunakan untuk mengatur kecepatan motor pompa. Metode defuzzifikasi yang digunakan adalah *weighted average*. Metode *weighted average* sering digunakan karena efisien dan tidak terlalu rumit dalam perhitungannya. Persamaan dibawah adalah persamaan keluaran untuk mengubah keluaran menjadi keluaran *crisp* (berbentuk non-fuzzy).

$$Z^* = \frac{\sum \mu_c(z) \cdot z}{\sum \mu_c(z)} \quad (4-4)$$

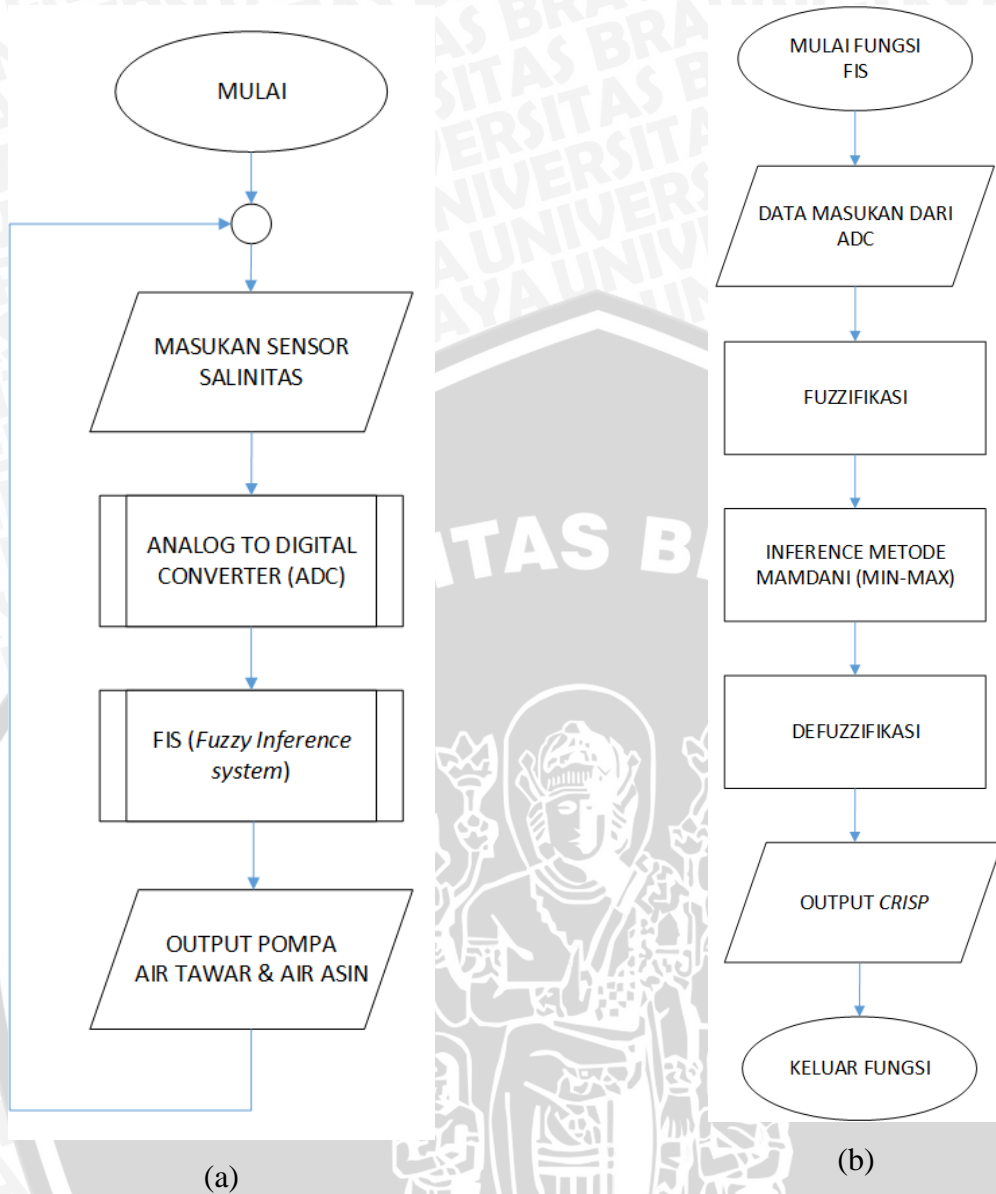
Dimana :

Z^* = Keluaran *crisp*.

μ_c = Nilai keanggotaan masing – masing daerah output.

z = Nilai tengah masing – masing daerah output.

Setelah perancangan kontrol logika *fuzzy* didapatkan, sistem sudah dapat diaplikasikan pada *plant* yaitu akuarium air laut. Gambar diagram alur kontrol salinitas dengan logika *fuzzy* secara keseluruhan dapat ditunjukkan dalam Gambar 4.12.

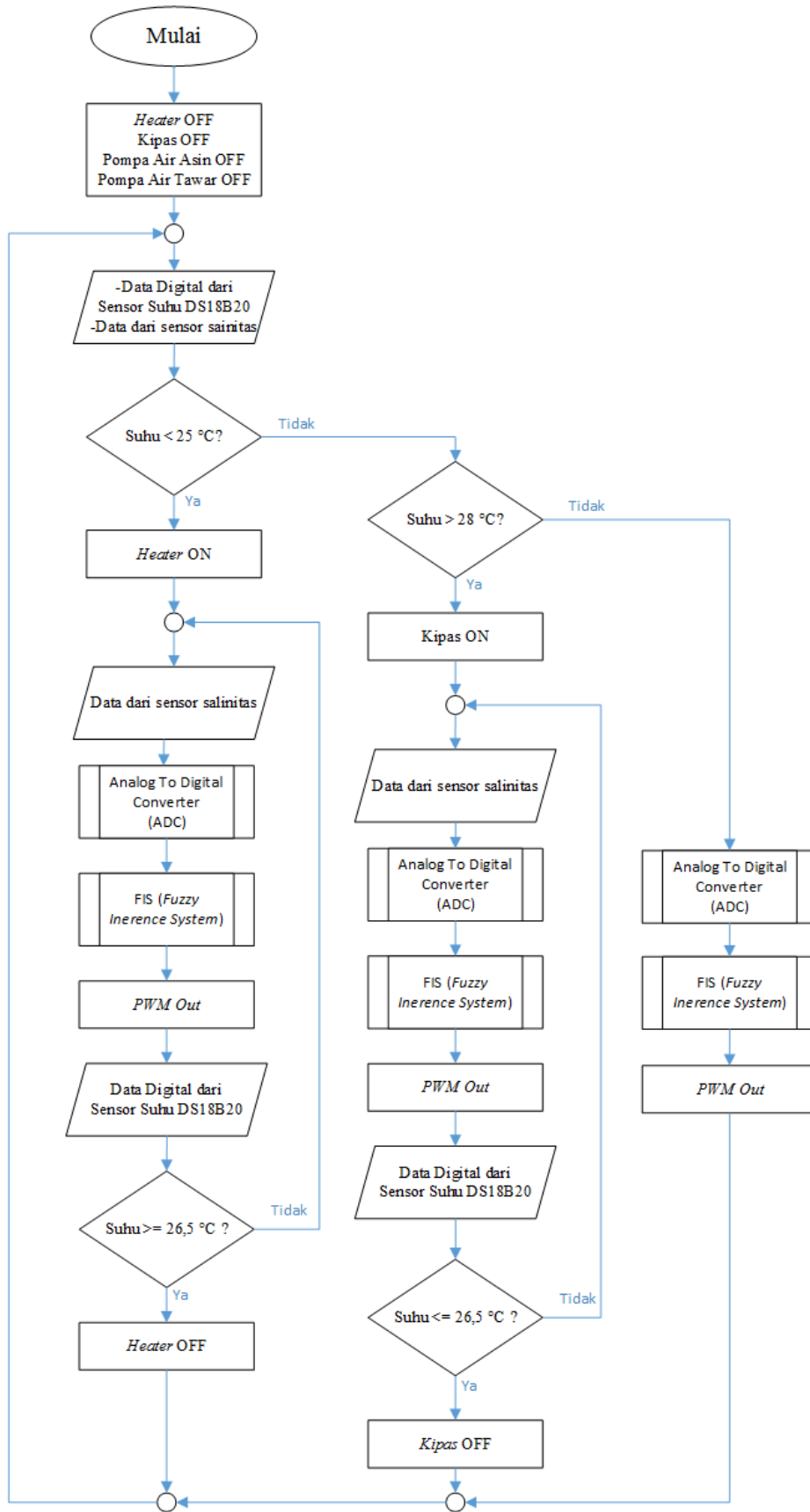


Gambar 4.12 (a) Diagram Alir Kontrol Salinitas

(b) Diagram Alir Fungsi FIS (Fuzzy Inference System)

4.1.3.3 Perancangan Perangkat Lunak Keseluruhan Sistem

Setelah perancangan perangkat lunak kontrol suhu dan salinitas, perancangan keseluruhan sistem ini dilakukan agar kedua subsistem dapat berjalan dengan baik dan sesuai dengan yang diinginkan. Diagram alir sistem keseluruhan dapat ditunjukkan pada Gambar 4.13.



Gambar 4.13 Diagram Alir Kontrol Suhu dan Salinitas



Pada gambar diagram alir diatas, dapat ditunjukkan cara kerja dari sistem kontrol suhu dan salinitas pada akuarium ikan badut tersebut. Pertama semua aktuator di-*setting* off atau mati sebagai kondisi awal. Selanjutnya mikrokontroler mendapatkan masukan data suhu dan salinitas dari sensor suhu dan sensor salinitas yang akan diproses dalam mikrokontroler. Sebelumnya telah dijelaskan, *setpoint* suhu yang diinginkan adalah sebesar $25\text{ }^{\circ}\text{C} - 28\text{ }^{\circ}\text{C}$. Ketika suhu air kurang dari sama dengan $25\text{ }^{\circ}\text{C}$, *heater* akan aktif hingga suhu naik menjadi $26,5\text{ }^{\circ}\text{C}$. sebaliknya, ketika suhu air mencapai lebih dari $28\text{ }^{\circ}\text{C}$. Pada masing – masing proses tersebut terdapat 2 proses perulangan. Agar fungsi pengontrolan salinitas tidak terganggu dengan perulangan diatas, proses fungsi kontrol salinitas air juga dimasukkan dalam masing – masing perulangan tersebut.

4.2 Pengujian Sistem

Pengujian terdiri dari pengujian komponen – komponen dari sistem yang terdiri dari pengujian sensor suhu, sensor salinitas, ADC mikrokontroler dan pengujian keseluruhan sistem.

4.2.1 Pengujian Sensor Suhu DS12B20 *Waterproof*

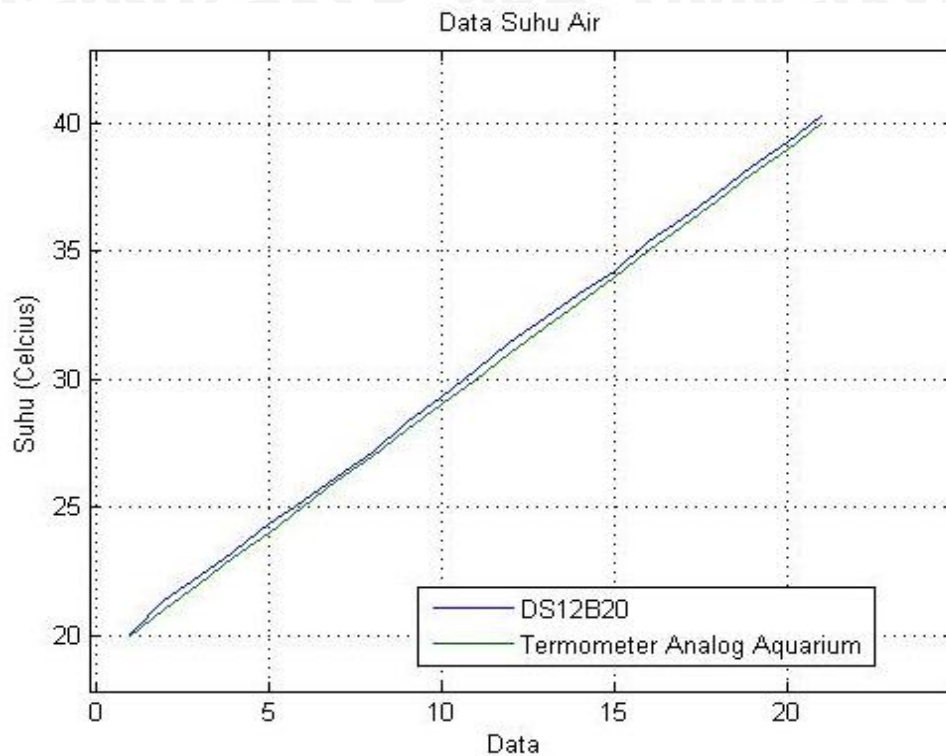
Pengujian sensor suhu DS12B20 *Waterproof* bertujuan untuk menguji keluaran sensor serta memberikan gambaran jika dibandingkan dengan termometer untuk akuarium pada umumnya, serta *error* yang timbul dibandingkan dengan termometer. Pengujian sensor suhu DS12B20 *Waterproof* membutuhkan termometer akuarium dan komunikasi serial pada mikrokontroler agar data dapat direkam langsung oleh komputer.

Pada pengujian sensor suhu DS12B20 *Waterproof* ini variabel yang diamati adalah keluaran suhu yang terbaca oleh sensor dengan range suhu $20^{\circ}\text{C} - 40^{\circ}\text{C}$ dengan kenaikan setiap 1°C . Pengujian dilakukan dengan cara memasukan sensor suhu dan termometer akuarium kedalam air percobaan dan pencuplikan nilai suhu yang terbaca oleh sensor dilakukan ketika termometer menunjukkan suhu air dengan range suhu $20^{\circ}\text{C} - 40^{\circ}\text{C}$ dengan kenaikan setiap 1°C . Percobaan dilakukan pada range $20^{\circ}\text{C} - 40^{\circ}\text{C}$ sudah memenuhi karena *setpoint* sistem ini tercakupi ($25^{\circ}\text{C} - 28^{\circ}\text{C}$). Hasil pengujian sensor suhu DS12B20 *Waterproof* dapat ditunjukkan pada Tabel 4.3.

Tabel 4.3 Hasil Pengujian Sensor Suhu DS12B20 *Waterproof*

Termometer	DS12B20 <i>Waterproof</i>	<i>error</i>
20 °C	20,01 °C	0,01 °C
21 °C	21,38 °C	0,38 °C
22 °C	22,31 °C	0,31 °C
23 °C	23,25 °C	0,25 °C
24 °C	24,35 °C	0,35 °C
25 °C	25,25 °C	0,25 °C
26 °C	26,18 °C	0,18 °C
27 °C	27,11 °C	0,11 °C
28 °C	28,31 °C	0,31 °C
29 °C	29,31 °C	0,31 °C
30 °C	30,31 °C	0,31 °C
31 °C	31,44 °C	0,44 °C
32 °C	32,38 °C	0,38 °C
33 °C	33,31 °C	0,31 °C
34 °C	34,25 °C	0,25 °C
35 °C	35,39 °C	0,39 °C
36 °C	36,3 °C	0,3 °C
37 °C	37,3 °C	0,3 °C
38 °C	38,3 °C	0,3 °C
39 °C	39,3 °C	0,3 °C
40 °C	40,3 °C	0,3 °C

Berdasarkan data diatas, *error* pembacaan sensor suhu ini cukup kecil. Jika dirata-rata *error* hanya sekitar 0,278 °C. Percobaan ini juga dapat membuktikan bahwa sensor berjalan dengan baik dan sesuai dengan *datasheet* yang tersedia. Gambar grafik perbandingan antara termometer dan keluaran Sensor Suhu DS12B20 *Waterproof* dapat ditunjukkan pada Gambar 4.14.



Gambar 4.14 Grafik Data Termometer dan Sensor Suhu DS12B20

4.2.2 Pengujian Sensor Salinitas

Pengujian sensor salinitas bertujuan untuk mengkalibrasi sensor yang telah dibuat serta menguji keluaran karakteristik sensor salinitas yang berbasis sensor *hall effect* tersebut. Kalibrasi dilakukan dengan cara mengujinya pada air yang memiliki salinitas yang berbeda dengan range salinitas 20 – 40 ppt dan dengan *step* 1 ppt. Setiap 5 detik data akan dimasukkan kedalam *data log* yang telah dibuat hingga terkumpul 400 data. Hasil rata – rata dari 400 data keluaran sensor salinitas berbasis sensor *hall effect* ini dapat ditunjukkan pada Tabel 4.4.

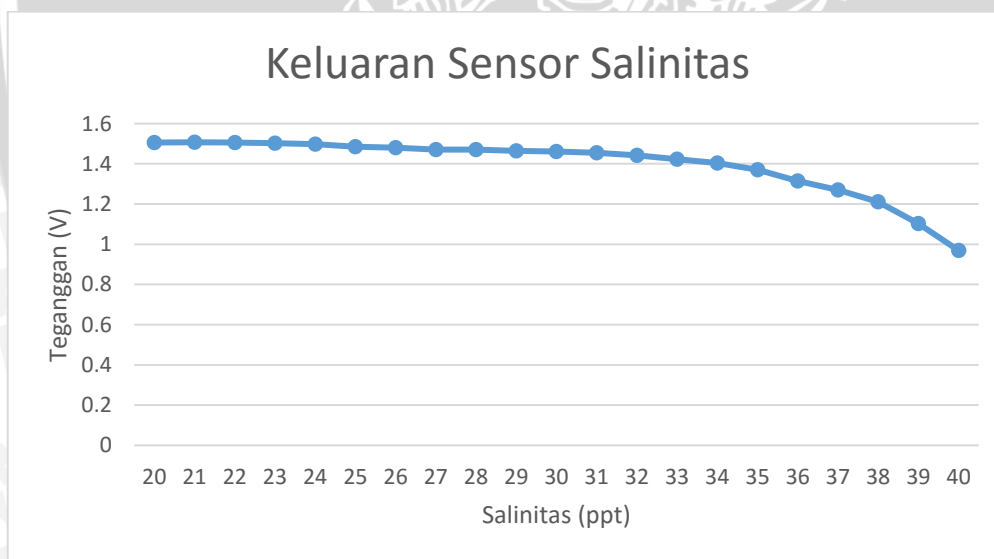
Tabel 4.4 Hasil Keluaran Sensor Salinitas Berbasis Sensor *hall effect*

Salinitas	Vout Sensor
20 ppt	1.506125 V
21 ppt	1.506995 V
22 ppt	1.505125 V
23 ppt	1.502275 V
24 ppt	1.498425 V
25 ppt	1.48425 V
26 ppt	1.47995 V
27 ppt	1.47125 V
28 ppt	1.46995 V

Tabel 4.4 Hasil Keluaran Sensor Salinitas Berbasiskan Sensor *hall effect*

Salinitas	Vout Sensor
29 ppt	1.46425V
30 ppt	1.46125 V
31 ppt	1.45425 V
32 ppt	1.44125 V
33 ppt	1.42275 V
34 ppt	1.40425 V
35 ppt	1.370275 V
36 ppt	1.315125 V
37 ppt	1.270125 V
38 ppt	1.210995 V
39 ppt	1.102275 V
40 ppt	0.969125 V

Range pengujian 20 – 40 ppt dapat dilaksanakan karena sudah mewakili range yang dibutuhkan sebagai setpoint, yaitu 27 - 33 ppt. Agar limit dari *setpoint* tidak dilewati, maka penulis membulatkan *setpoint* menjadi 28 - 32 ppt. Grafik dari data diatas dapat ditunjukkan pada Gambar 4.15.

Gambar 4.15 Grafik Data Keluaran Sensor Salinitas Berbasiskan Sensor *hall effect*

Pada gambar diatas, dapat diketahui karakteristik dari sensor salinitas berbasis sensor *hall effect* ini tidak linier, sehingga penulis melakukan 2 kali regresi linear yang dapat dianggap menggambarkan kondisi data yang berlaku untuk limit data dari 20 ppt hingga 40 ppt. Regresi yang pertama untuk data 20 – 34 ppt, karena pada data diatas keluaran sensor

turun dengan curam pada data ke 34. Regresi kedua dilakukan dari data 34 – 40 ppt. Untuk memperoleh persamaan terhadap data diatas, bentuk umum regresi linear dapat dituiskan:

$$y = a_0 + a_1x \quad (4-5)$$

Untuk mendapatkan persamaan tersebut, a_0 , a_1 , dan a_2 didapatkan dari sistem persamaan linier:

$$na_0 + a_1 \sum x_i = \sum y_i \quad (4-6)$$

$$a_0 \sum x_i + a_1 \sum x_i^2 = \sum y_i x_i \quad (4-7)$$

Untuk data 1 – 15 (20 – 34 ppt) dengan cara diatas, didapatkan sistem persamaan linear:

$$15 a_0 + 405 a_1 = 22,072$$

$$405 a_0 + 11215 a_1 = 594,06$$

Solusi dari SPL diatas adalah :

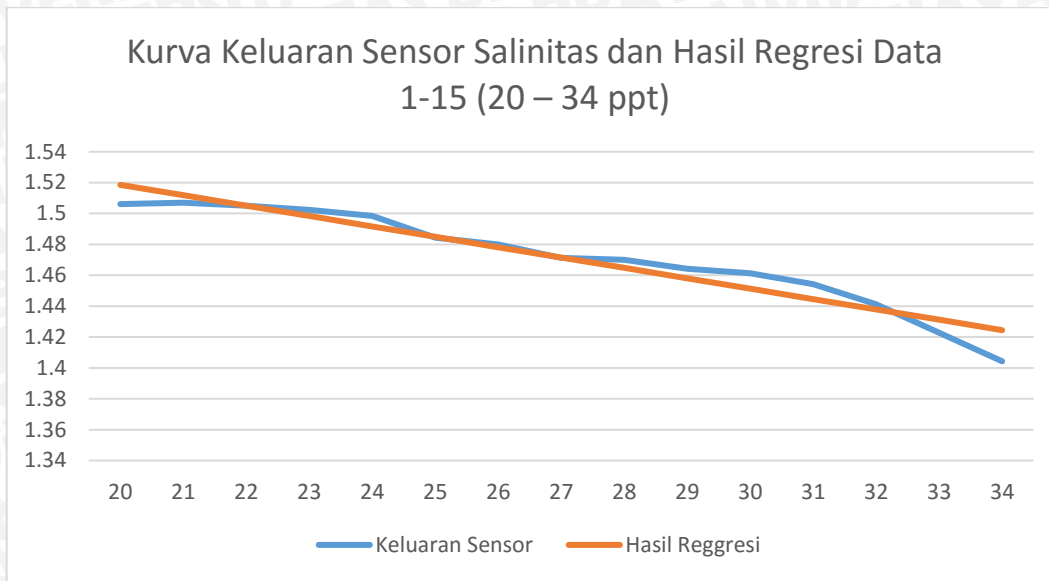
$$a_0 = 1,652972238$$

$$a_1 = -0,00672243$$

Jika kita substitusikan nilai a_0 dan a_1 pada persamaan (4-5) diatas maka didapatkan:

$$y_1 = 1,652972238 + (-0,00672243 x)$$

Persamaan diatas adalah persamaan yang menggambarkan keluaran sensor untuk data 1-15 (20 – 34 ppt). Grafik data dan hasil regresi data 1-15 (20 – 34 ppt) dapat ditunjukkan pada Gambar 4.16.



Gambar 4.16 Kurva Keluaran Sensor Salinitas dan Hasil Regresi Data 1-15 (20 – 34 ppt).

Dengan sama seperti cara diatas, dilakukan regresi terhadap data dari 15 – 21 (34 – 40 ppt), didapatkan sistem persamaan linear:

$$7 a_0 + 259 a_1 = 8,64217$$

$$259 a_0 + 9611 a_1 = 317,814785$$

Solusi dari SPL diatas adalah :

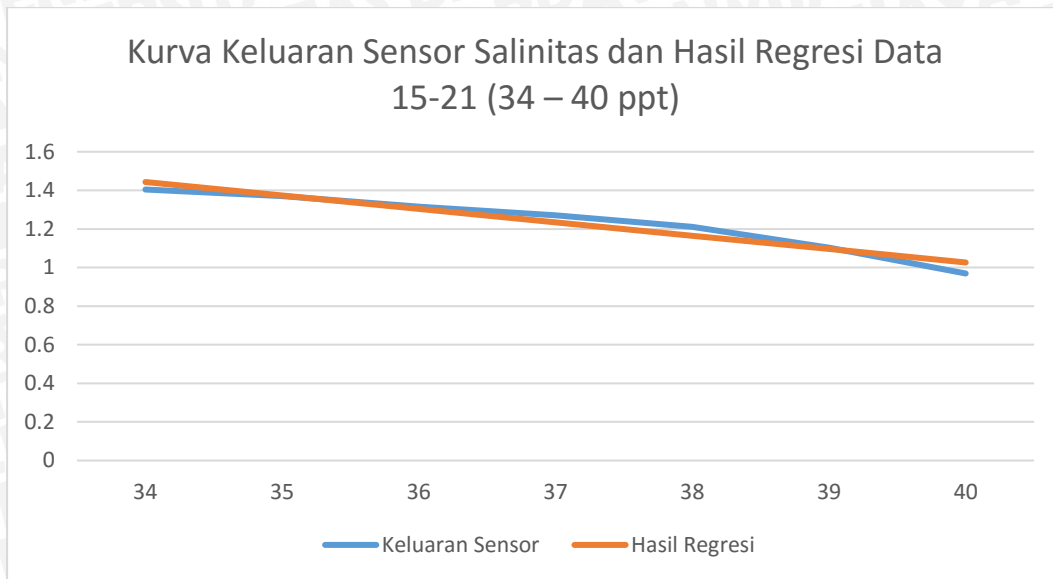
$$a_0 = 3.805441607$$

$$a_1 = -0.069482321$$

Jika kita substitusikan nilai a_0 dan a_1 pada persamaan (4-5) diatas maka didapatkan:

$$y_2 = 3.805441607 + (-0.069482321 x)$$

Persamaan diatas adalah persamaan yang menggambarkan keluaran sensor untuk data 15 – 21 (34 – 40 ppt). Grafik data dan hasil regresi data 15 – 21 (34 – 40 ppt) dapat ditunjukkan pada Gambar 4.17.



Gambar 4.17 Kurva Keluaran Sensor Salinitas dan Hasil Regresi Data 15 – 21 (34 – 40 ppt).

Untuk mencari persamaan yang menggambarkan keseuruhan sistem dari dua regresi tersebut, dicari nilai pada sumbu x (nilai dari salinitas) agar kedua persamaan memiliki hasil yang sama, atau dalam arti lain nilai pada sumbu y sama untuk mencari batas persamaan yang terpenuhi.

$$y_1 = y_2$$

$$1,652972238 + (-0,00672243 x) = 3,805441607 + (-0,069482321 x)$$

$$-0,00672243 x + 0,069482321 x = 3,805441607 - 1,652972238$$

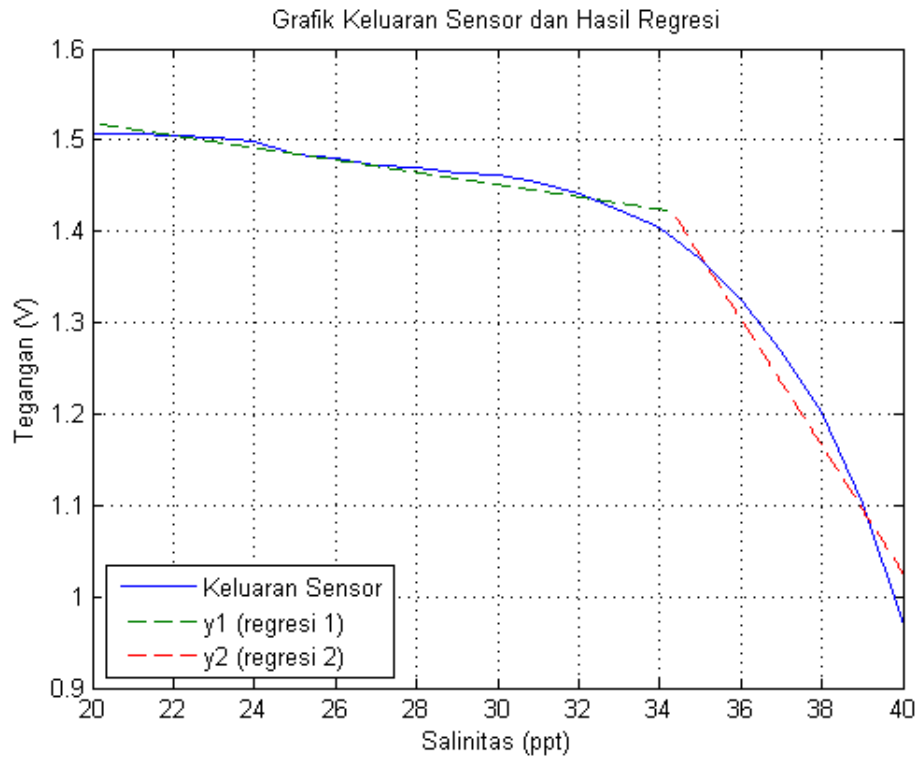
$$0,062759891 x = 2,152469369$$

$$x = 34,29689464 \approx 34,23$$

Setelah didapatkan batas nilai x antara kedua persamaan, maka dapat dituliskan hasil regresi yang menggambarkan

$$f(x) = \begin{cases} 1,652972238 + (-0,00672243 x), & 20 \leq x < 34,23 \\ 3,805441607 + (-0,069482321 x), & 34,23 \leq x < 40 \end{cases}$$

Grafik keluaran sensor serta persamaan kedua hasil regresi dengan batas yang telah ditentukan diatas ditunjukkan pada Gambar 4.18.



Gambar 4.18 Grafik Keluaran Sensor Serta Hasil Regresi

Setelah didapatkan regresi dari data keluaran sensor, dilakukan percobaan terhadap sensor dengan salinitas yang berbeda – beda. Tabel hasil pengujian sensor dapat ditunjukkan pada Tabel 4.5.

Tabel 4.5 Hasil Pengujian Sensor Salinitas

Salinitas Air	Pembacaan Sensor
20	21,6426
21	21,7080
22	22,5410
23	23,4420
24	24,3110
25	25,7593
26	26,3534
27	27,6286
28	28,4250
29	29,9208
30	29,9883
31	31,2676
32	32,5631
33	32,6849
34	33,5749

Tabel 4.5 Hasil Pengujian Sensor Salinitas

Salinitas Air	Pembacaan Sensor
28	28,4250
29	29,9208
30	29,9883
31	31,2676
32	32,5631
33	32,6849
34	33,5749
35	35,0876
36	36,2948
37	37,4172
38	38,6088
39	39,1156
40	39,3760

Hasil pengambilan data diatas dilakukan dengan mengambil 120 data pada masing – masing tingkatan salinitas, kemudian dirata-rata kan. Dari data diatas, dapat diketahui *error* rata – rata yang terjadi sebesar 1,83 %. Dengan *error* yang cukup kecil, sensor tersebut dapat digunakan sebagai pengukur salinitas air.

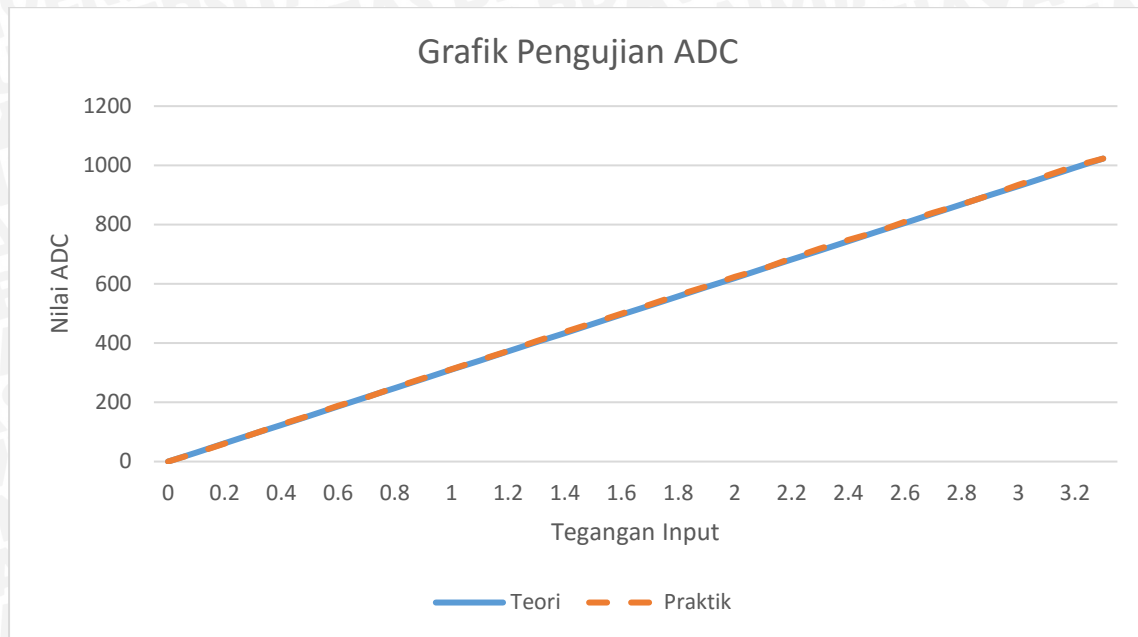
4.2.3 Pengujian ADC (*Analog to Digital Converter*) Arduino Due

ADC (*Analog to Digital Converter*) adalah proses pengubahan sinyal analog yang dihasilkan oleh sensor menjadi sinyal digital agar dapat diproses oleh mikrokontroler. Resolusi ADC pada mikrokontroler Arduino Due dapat diprogram dengan resolusi mulai dari 8, 10, 12, hingga 16 bit. Tegangan kerja Arduino Due adalah sebesar 3,3V sehingga untuk ADC dengan 10 bit (resolusi *default* ADC Arduino) tegangan masukan 0 – 3,3 V setara dengan nilai digital dari 0 – 1023. Menurut teori yang telah dibahas sebelumnya, resolusi ADC 3,2226 mV artinya setiap kenaikan 3,2226 mV pada keluaran sensor, nilai ADC yang terbaca oleh mikrokontroler akan mengalami kenaikan sebesar 1 nilai. Dengan menaikan *input* atau masukan analog pada pin analog Arduino kita dapat membandingkan antara teori dan praktik pada lapangan. Hasil dari pengujian ADC dapat ditunjukkan pada Tabel 4.6.

Tabel 4.6 Hasil Pengujian ADC Arduino Due

Input (V)	Nilai ADC	
	Perhitungan	Praktik
0	0	0
0,1	31	30
0,2	62	61
0,3	93	93
0,4	124	126
0,5	155	156
0,6	186	188
0,7	217	216
0,8	248	250
0,9	279	281
1	310	312
1,1	341	343
1,2	372	373
1,3	403	407
1,4	434	438
1,5	465	468
1,6	496	500
1,7	527	530
1,8	558	562
1,9	589	591
2	620	624
2,1	651	651
2,2	682	687
2,3	713	719
2,4	744	748
2,5	775	775
2,6	806	810
2,7	837	840
2,8	868	868
2,9	899	900
3	930	934
3,1	961	966
3,2	992	998
3,3	1023	1023

Dari data diatas, *error* yang terjadi setelah pengujian ADC sebesar 0.645 %, artinya ADC dari mikrokontroler Arduino Due memiliki kepresisian sangat baik. Gambar dari grafik pengujian ADC ditunjukkan pada Gambar 4.19.



Gambar 4.19 Grafik Pengujian ADC Arduino Due

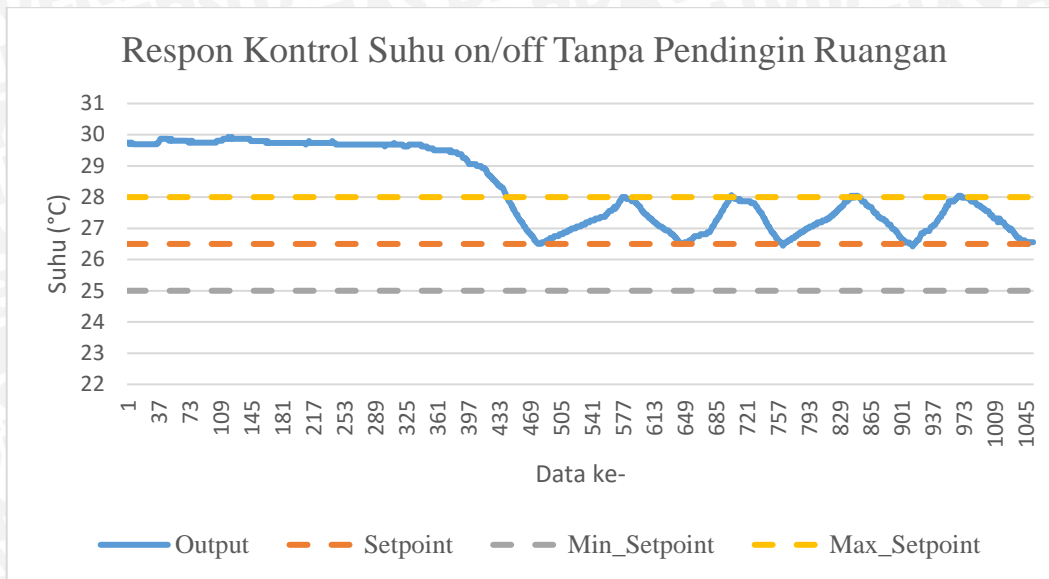
4.2.4 Pengujian Keseluruhan Sistem

Setelah pengujian pada masing – masing komponen penyusun, pengujian keseluruhan sistem dilakukan. Pengujian keseluruhan sistem dilakukan dengan tujuan mengetahui respon dan kestabilan sistem kontrol suhu dengan kontroler on/ff dan sistem kontrol salinitas air dengan kontroler logika *fuzzy*.

4.2.4.1 Pengujian Kontrol Suhu

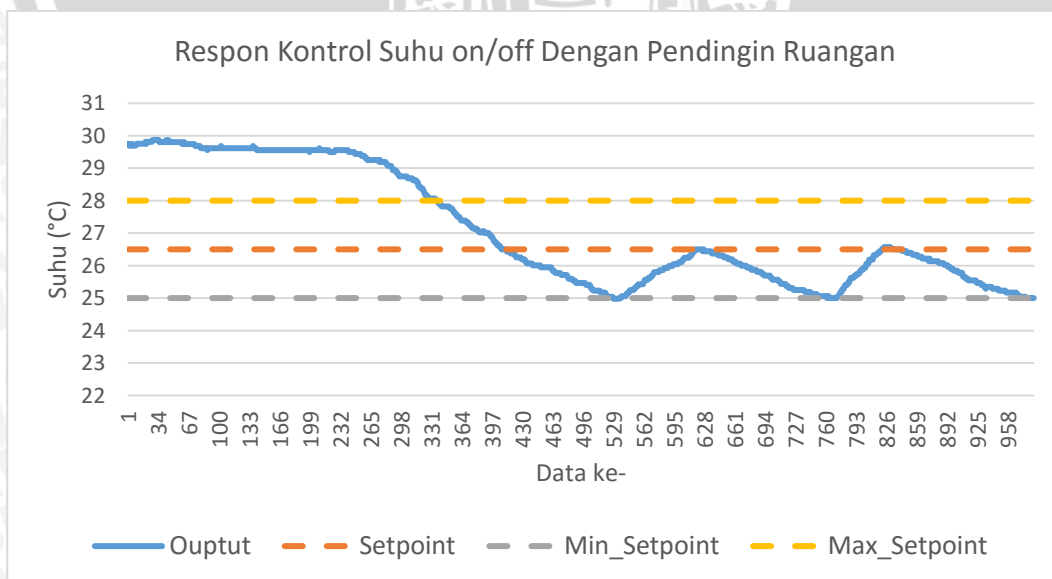
Pada kontrol suhu, metode yang digunakan adalah metode kontroler on/off. Pengujian dilakukan dua kali pengujian, yaitu dalam suhu ruangan dengan pendingin ruangan dan tanpa pendingin ruangan. Data diambil dengan *data logger* dengan menggunakan software Gobetwino. Data dikirimkan dari Arduino Due melewati komunikasi serial dan software Gobetwino akan mendapatkan data tersebut, kemudian menyimpan pada memori local PC.

Gambar respon dari kontrol suhu tanpa pendingin ruangan dapat ditunjukkan pada Gambar 4.20.



Gambar 4.20 Grafik respon kontrol suhu on/off tanpa pendingin ruangan

Pada gambar diatas, dapat diketahui bahwa kontroler ON/OFF ini dapat mempertahankan suhu dengan range antara 25 – 28 °C. Pada saat suhu air lebih dari 28 °C saat sistem pertama kali dijalankan, kipas aktif sehingga menurunkan suhu hingga suhu mencapai 26,5 °C. Setelah mencapai suhu 26,5 °C kipas mati, kemudian suhu air naik kembali. Ketika suhu air naik kembali hingga mencapai 28 °C, kipas aktif kembali, begitu seterusnya. Pengujian selanjutnya yaitu plant ditempatkan pada ruangan dengan pendingin ruangan. Gambar respon dari kontrol suhu dengan pendingin ruangan dapat ditunjukkan pada Gambar 4.21.

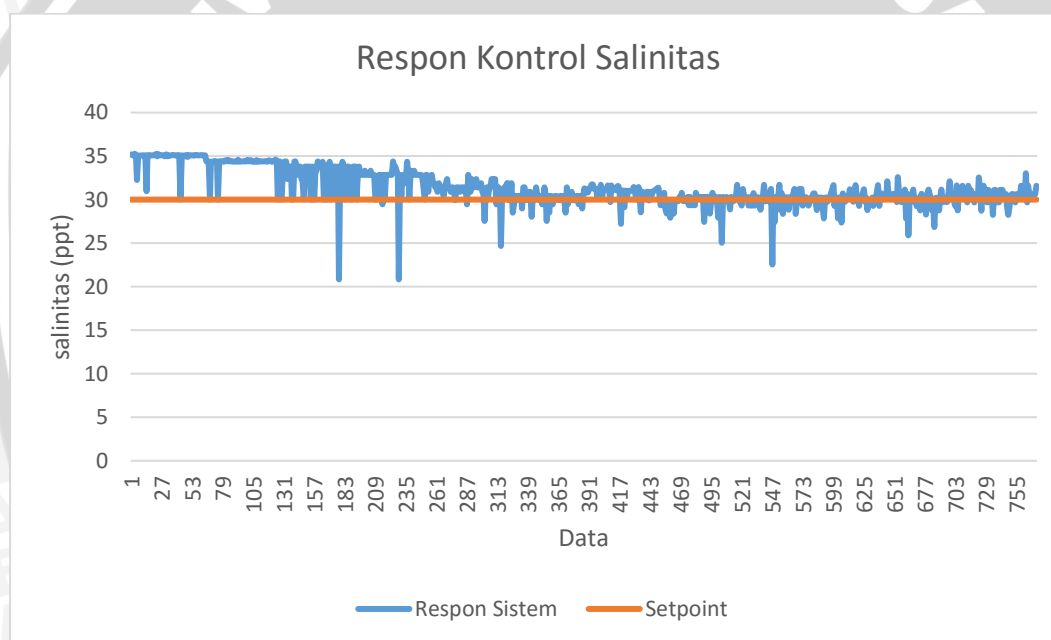


Gambar 4.21 Grafik respon kontrol suhu on/off dengan pendingin ruangan

Ketika sistem ditempatkan pada ruangan dengan pendingin ruangan, suhu air turun hingga mencapai 25 °C, kemudian pemanas akuarium (*heater*) aktif dan memanaskan air hingga mencapai 26,5 °C. Pada saat suhu mencapai suhu 26,5 °C, *heater* akan mati dan pendingin ruangan mendinginkan air akuarium kembali hingga 25 °C lalu *heater* aktif kembali, begitu seterusnya.

4.2.4.2 Pengujian Kontrol Salinitas

Pengambilan data kontrol salinitas air pada sistem dilakukan pada air laut yang memiliki salinitas sekitar 35 ppt. *Data logging* yang dilakukan pada kontrol salinitas ini sama seperti pada kontrol suhu. Dengan data yang didapatkan dari sensor salinitas, didapatkan respon sistem pada kontrol salinitas dengan logika *fuzzy*. Respon sistem kontrol salinitas air pada akuarium ikan badut dapat ditunjukkan pada Gambar 4.22.



Gambar 4.22 Grafik Respon Kontrol Salinitas

Dari data yang telah didapatkan, dapat diketahui bahwa diketahui *error steady state* yaitu sekitar 1,7476 ppt atau sekitar 5,825 %. Tetapi ada gangguan yang terjadi cukup besar yang diakibatkan oleh sensor yang kurang baik. Pembacaan sensor fluktuatif juga diakibatkan oleh sensor salinitas yang kurang baik dan gelombang air yang mempengaruhi hidrometer.





BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan percobaan yang telah dilakukan didapatkan kesimpulan sebagai berikut:

1. Perancangan perangkat keras alat direalisasikan dengan membuat diagram blok keseluruhan sistem dan selanjutnya merancang subsistem yang terdiri dari perancangan sensor dan aktuator.
2. Perancangan perangkat lunak atau *software* sistem dengan membuat *flowchart* kontroler on/off untuk mengontrol parameter suhu dan kontroler logika *fuzzy* untuk mengontrol parameter salinitas yang selanjutnya diaplikasikan pada program Arduino Due.
3. Kontrol suhu air pada akuarium ikan badut menggunakan kontroler on/off dengan *setpoint* 25 °C - 28 °C dapat berjalan dengan baik dengan *error steady state* sebesar 3,12 % dengan pendingin ruangan mati dan *error steady state* sebesar 2,75 % dengan pendingin ruangan menyala terhadap nilai tengah (26,5 °C).
4. Kontrol salinitas air pada akuarium ikan badut menggunakan kontroler logika *fuzzy* dengan *setpoint* 27 – 33 ppt dapat berjalan dengan baik dengan *error steady state* sebesar 1,7476 ppt atau sekitar 5,825 % terhadap nilai tengah *setpoint* (30 ppt).
5. Dalam percobaan yang dilakukan selama ±2 bulan hingga makalah ilmiah ini selesai, ikan badut dapat hidup dan tidak mengalami kematian, yang sebelumnya hanya dapat bertahan hidup ±20 hari tanpa menggunakan kontroler suhu dan salinitas dengan perlakuan selain parameter suhu dan salinitas yang sama.

5.2 Saran

Berdasarkan analisis yang telah dilakukan pada skripsi ini, saran yang dapat diberikan untuk pengembangan penelitian yaitu:

1. Pengembangan selanjutnya dapat menggunakan sensor salinitas yang lebih baik agar *error* berkurang dan stabilitas pembacaan sensor dapat ditingkatkan.
2. Pengembangan pengendalian parameter lain pada akuarium seperti PH, kadar amonia, kadar zat kapur, nitrat, dll.
3. Pengembangan alat sebagai tempat pengembang biakan ikan laut tertentu.



DAFTAR PUSTAKA

- Bower, Carol E, & Joseph P. Bidwell. 1978. *Ionization of Ammonia in Seawater: Effects of Temperature, PH, and Salinity*. West Hartord, USA.
- Walsh, Margaret Paton. 2015. *Status Review Report: Orange Clownfish (Amphiprion percula)*. National Marine Fisheries Service.
- Ogata, K. 1997. *Teknik Kontrol Automatik (Sistem Pengaturan)*. Terjemahan Edi Leksono, Jakarta : Erlangga.
- Yan, Jun., James Power, & Michael Ryan. 1994. *Using Fuzzy Logic*. New Jersey : Penerbit Prentice Hall.
- Kamajaya. 2007. *Cerdas Belajar Fisika*. Jakarta : Penerbit Grafindo Media Pratama.
- Nursailuf, Andi. 2004. *Akuarium Laut : cara mudah memindahkan panorama kehidupan laut kerumah anda*, Depok : Niaga Swadaya.
- Mismail, Budiono. 2010. *Akuarium Terumbu Karang*, Malang : Penerbit Universitas Brawijaya Press (UB Press).
- Peraturan Direktur Jendral Perlindungan Hutan dan Konservasi Alam, "Pedoman Penangkaran/Transplantasi Karang Hias yang Diperdagangkan" SK.09/IV/Set-3/2008, 29 Januari 2008.*
- http://animaldiversity.org/accounts/Amphiprion_percula/#20020904145365 (Diakses 21 februari 2016, 11.30 WIB)
- <http://reefkeeping.com/issues/2004-05/rhf/> (Diakses 21 februari 2016, 11.30 WIB)
- <http://nationalgeographic.co.id/berita/2013/10/terbaru-panjang-garis-pantai-indonesia-capai-99000-kilometer/> (Diakses 21 februari 2016, 11.44 WIB)





LAMPIRAN

Lampiran 1. Dokumentasi Alat



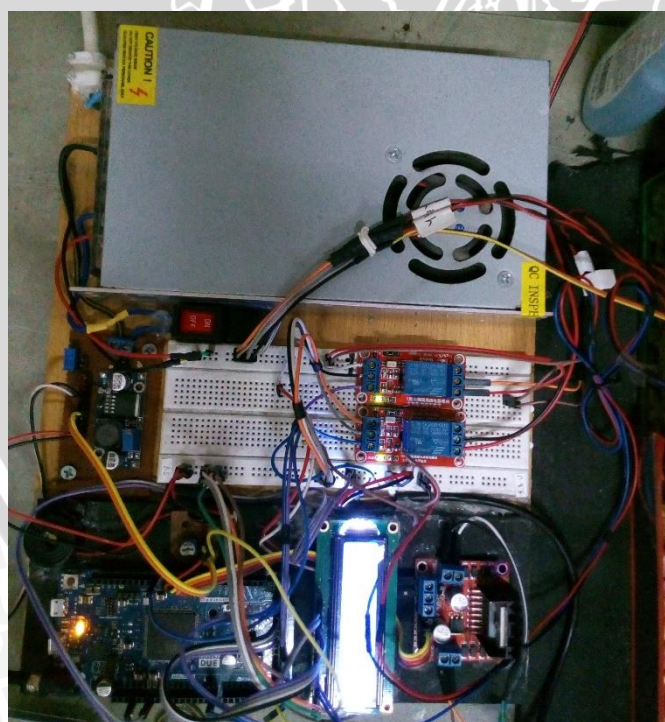
Gambar 1 Tanki Utama Akuarium dan Suplai Air Tawar dan Asin



Gambar 2 Aktuator (kipas)



Gambar 3 Aktuator (*Heater*)



Gambar 3 Power Supply dan Kontroler

Lampiran 2. Listing Program

```

//S_SUHU DS18B20
#include <OneWire.h>
OneWire ds(8);
unsigned int suhu_max=30;
unsigned int suhu_min=20;

// VAR FOR DS18BB20
//byte i;
byte present = 0;
byte type_s;
byte data[12];
byte addr[8];
float celsius, fahrenheit;

//sensor salinitas
const int analogInPin = A7;
int sensorValue = 0;
double salinitas=0;

//motor
//Right side motor control
int in_1=5,in_2=4;
//Left side motor control
int in_3=3,in_4=2;

//FAN
const int relay_fan = 50;

//heater
const int relay_heater = 52;

//BUZZER
const int buzz_pin = 7;

//LCD
#include <LiquidCrystal.h>
LiquidCrystal lcd(22, 24, 28, 30, 32, 34);

//for salinitas
#include <stdio.h>
#include <math.h>
#include <stdlib.h>

int i,j;
float output_tawar, output_asin;

float rule_tawar[5][5],rule_asin[5][5];
float error=0,d_error=0,hold=0;

float batas_error[5]={-4,-2,0,2,4}; //batas error
float batas_d_error[5]={-0.8,-0.4,0,0.4,0.8}; //batas d_error

```



```
float N_Tengah_out[5]= {12.5,56.03,118.3,180.5,239.6}; //batas out tawar & asin
```

```
float NB_1=0,NS_1=0,ZE_1=0,PS_1=0,PB_1=0; //Degree Of Membership error
float NB_2=0,NS_2=0,ZE_2=0,PS_2=0,PB_2=0; //Degree Of Membership d_error
```

```
float Z_1,VS_1,S_1,B_1,VB_1; //Degree Of Membership out_tawar
float Z_2,VS_2,S_2,B_2,VB_2; //Degree Of Membership out_asin
```

```
//fungsi2 for sainitas
```

```
void BACA_SALINITAS ();
void fuzzifikasi_error ();
void fuzzifikasi_d_error ();
void periksa_rule ();
void periksa_rule_tawar();
void periksa_rule_asin();
void defuzzifikasi ();
```

```
//fungsi2 for suhu
```

```
void BACA_SUHU ();
void PROSES_SUHU ();
```

```
//fungsi lcd
```

```
void LCD ();
```

```
void setup()
```

```
{
//RELAY_FAN_HEATER
pinMode(relay_fan, OUTPUT);
pinMode(relay_heater, OUTPUT);
```

```
//motor
```

```
pinMode(in_1,OUTPUT);
pinMode(in_2,OUTPUT);
pinMode(in_3,OUTPUT);
pinMode(in_4,OUTPUT);
```

```
//LCD
```

```
lcd.begin(16,2);
```

```
//BUZZER
```

```
pinMode(buzz_pin, OUTPUT);
```

```
//Komunikasi serial
```

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

```
}
```

```
void SALINITAS ()
```

```
{
BACA_SALINITAS ();
fuzzifikasi_error ();
fuzzifikasi_d_error ();
periksa_rule ();
defuzzifikasi ();
```

UNIVERSITAS BRAWIJAYA



```

}
void SUHU()
{
  BACA_SUHU ();
  PROSES_SUHU ();
}

void loop()
{
  SALINITAS ();
  SUHU ();
}

void BACA_SALINITAS ()
{
  // f(x)= 1,652972238 - 0,00672243 x    20 ≤x<34,23
  // x = ( 1,652972238 - y ) / 0,00672243    1.4224 < y < 1.5185
  // ADC 3,2258 mV                        441 < y < 470

  // f(x)= 3.805441607 - 0.069482321 x    34,23 ≤x<40
  // x = ( 3.805441607 - y ) / 0.069482321    1.0261 < y < 1.4224
  //
  //                                318 < y < 441

  sensorValue = analogRead(analogInPin);

  if (441 < sensorValue && sensorValue < 470)
  {
    salinitas= ( 1.652972238 - (sensorValue*0.0032258) ) / 0.00672243;
  }
  else if (318 < sensorValue && sensorValue < 441)
  {
    salinitas= ( 3.805441607 - (sensorValue*0.0032258) ) / 0.069482321 ;
  }
  else // diluar range
  {
    salinitas=30;
  }
}

void fuzzifikasi_error ()
{
  NB_1=0;NS_1=0;ZE_1=0;PS_1=0;PB_1=0;

  error= salinitas-30;

  if (error<=-4)
  {
    NB_1=1;
  }
  else if (error > -4 && error <=-2)
  {
    NB_1=(batas_error[1]-error)/(batas_error[1]-batas_error[0]);
  }
}

```

```

    NS_1=(batas_error[0]-error)/(batas_error[0]-batas_error[1]);
}
else if (error > -2 && error <=0)
{
    NS_1=(batas_error[2]-error)/(batas_error[2]-batas_error[1]);
    ZE_1=(batas_error[1]-error)/(batas_error[1]-batas_error[2]);
}
else if (error > 0 && error <=2)
{
    ZE_1=(batas_error[3]-error)/(batas_error[3]-batas_error[2]);
    PS_1=(batas_error[2]-error)/(batas_error[2]-batas_error[3]);
}
else if (error > 2 && error <=4)
{
    PS_1=(batas_error[4]-error)/(batas_error[4]-batas_error[3]);
    PB_1=(batas_error[3]-error)/(batas_error[3]-batas_error[4]);
}
else
{
    PB_1=1;
}
}

void fuzzifikasi_d_error ()
{
    NB_2=0;NS_2=0;ZE_2=0;PS_2=0;PB_2=0;

    d_error=error-hold;
    hold=error;

    //printf("\n\n %lf,%lf ",error,d_error);
    if (d_error<=-0.8)
    {
        NB_2=1;
    }
    else if (d_error > -0.8 && d_error <=-0.4)
    {
        NB_2=(batas_d_error[1]-d_error)/(batas_d_error[1]-batas_d_error[0]);
        NS_2=(batas_d_error[0]-d_error)/(batas_d_error[0]-batas_d_error[1]);
    }
    else if (d_error > -0.4 && d_error <=0)
    {
        NS_2=(batas_d_error[2]-d_error)/(batas_d_error[2]-batas_d_error[1]);
        ZE_2=(batas_d_error[1]-d_error)/(batas_d_error[1]-batas_d_error[2]);
    }
    else if (d_error > 0 && d_error <=0.4)
    {
        ZE_2=(batas_d_error[3]-d_error)/(batas_d_error[3]-batas_d_error[2]);
        PS_2=(batas_d_error[2]-d_error)/(batas_d_error[2]-batas_d_error[3]);
    }
    else if (d_error > 0.4 && d_error <=0.8)
    {
        PS_2=(batas_d_error[4]-d_error)/(batas_d_error[4]-batas_d_error[3]);
    }
}

```



```

    PB_2=(batas_d_error[3]-d_error)/(batas_d_error[3]-batas_d_error[4]);
  }
  else
  {
    PB_2=1;
  }
}

void periksa_rule ()
{
  periksa_rule_tawar ();
  periksa_rule_asin();
}

void periksa_rule_tawar()
{
  for (i=0;i<=4;i++)
  {
    for (j=0;j<=4;j++)
    {
      {rule_tawar[i][j]=0;}
    }
  }

  if (NB_1>NB_2)
    {rule_tawar[0][0]=NB_2;}
  else
    {rule_tawar[0][0]=NB_1;}
  if (NS_1>NB_2)
    {rule_tawar[0][1]=NB_2;}
  else
    {rule_tawar[0][1]=NS_1;}
  if (ZE_1>NB_2)
    {rule_tawar[0][2]=NB_2;}
  else
    {rule_tawar[0][2]=ZE_1;}
  if (PS_1>NB_2)
    {rule_tawar[0][3]=NB_2;}
  else
    {rule_tawar[0][3]=PS_1;}
  if (PB_1>NB_2)
    {rule_tawar[0][4]=NB_2;}
  else
    {rule_tawar[0][4]=PB_1;}

  if (NB_1>NS_2)
    {rule_tawar[1][0]=NS_2;}
  else
    {rule_tawar[1][0]=NB_1;}
  if (NS_1>NS_2)
    {rule_tawar[1][1]=NS_2;}
  else
    {rule_tawar[1][1]=NS_1;}

```



```

if (ZE_1>NS_2)
  {rule_tawar[1][2]=NS_2;}
else
  {rule_tawar[1][2]=ZE_1;}
if (PS_1>NS_2)
  {rule_tawar[1][3]=NS_2;}
else
  {rule_tawar[1][3]=PS_1;}
if (PB_1>NS_2)
  {rule_tawar[1][4]=NS_2;}
else
  {rule_tawar[1][4]=PB_1;}

if (NB_1>ZE_2)
  {rule_tawar[2][0]=ZE_2;}
else
  {rule_tawar[2][0]=NB_1;}
if (NS_1>ZE_2)
  {rule_tawar[2][1]=ZE_2;}
else
  {rule_tawar[2][1]=NS_1;}
if (ZE_1>ZE_2)
  {rule_tawar[2][2]=ZE_2;}
else
  {rule_tawar[2][2]=ZE_1;}
if (PS_1>ZE_2)
  {rule_tawar[2][3]=ZE_2;}
else
  {rule_tawar[2][3]=PS_1;}
if (PB_1>ZE_2)
  {rule_tawar[2][4]=ZE_2;}
else
  {rule_tawar[2][4]=PB_1;}

if (NB_1>PS_2)
  {rule_tawar[3][0]=PS_2;}
else
  {rule_tawar[3][0]=NB_1;}
if (NS_1>PS_2)
  {rule_tawar[3][1]=PS_2;}
else
  {rule_tawar[3][1]=NS_1;}
if (ZE_1>PS_2)
  {rule_tawar[3][2]=PS_2;}
else
  {rule_tawar[3][2]=ZE_1;}
if (PS_1>PS_2)
  {rule_tawar[3][3]=PS_2;}
else
  {rule_tawar[3][3]=PS_1;}
if (PB_1>PS_2)
  {rule_tawar[3][4]=PS_2;}

```



```

else
  {rule_tawar[3][4]=PB_1;}

if (NB_1>PB_2)
  {rule_tawar[4][0]=PB_2;}
else
  {rule_tawar[4][0]=NB_1;}
if (NS_1>PB_2)
  {rule_tawar[4][1]=PB_2;}
else
  {rule_tawar[4][1]=NS_1;}
if (ZE_1>PB_2)
  {rule_tawar[4][2]=PB_2;}
else
  {rule_tawar[4][2]=ZE_1;}
if (PS_1>PB_2)
  {rule_tawar[4][3]=PB_2;}
else
  {rule_tawar[4][3]=PS_1;}
if (PB_1>PB_2)
  {rule_tawar[4][4]=PB_2;}
else
  {rule_tawar[4][4]=PB_1;}

//Mencari DOM output (MAX)
VB_1= rule_tawar[0][4];
if (rule_tawar[1][4]>rule_tawar[2][4])
  {B_1=rule_tawar[1][4];}
else if (rule_tawar[1][4] < rule_tawar[2][4])
  {B_1=rule_tawar[2][4];}

S_1=rule_tawar[0][3];
if (S_1<rule_tawar[1][3])
  {S_1=rule_tawar[1][3];}
if (S_1<rule_tawar[2][3])
  {S_1=rule_tawar[2][3];}

Z_1=rule_tawar[0][0];
for (i=0;i<=2;i++)
  {
    for (j=0;j<=2;j++)
      {
        if (Z_1<rule_tawar[i][j])
          {Z_1=rule_tawar[i][j];}
      }
  }
for (i=3;i<=4;i++)
  {
    for (j=0;j<=4;j++)
      {
        if (Z_1<rule_tawar[i][j])
          {Z_1=rule_tawar[i][j];}
      }
  }

```



```

}

void periksa_rule_asin()
{
    for (i=0;i<=4;i++)
    {
        for (j=0;j<=4;j++)
        {
            {rule_asin[i][j]=0;}
        }
    }

    if (NB_1>NB_2)
        {rule_asin[0][0]=NB_2;}
    else
        {rule_asin[0][0]=NB_1;}
    if (NS_1>NB_2)
        {rule_asin[0][1]=NB_2;}
    else
        {rule_asin[0][1]=NS_1;}
    if (ZE_1>NB_2)
        {rule_asin[0][2]=NB_2;}
    else
        {rule_asin[0][2]=ZE_1;}
    if (PS_1>NB_2)
        {rule_asin[0][3]=NB_2;}
    else
        {rule_asin[0][3]=PS_1;}
    if (PB_1>NB_2)
        {rule_asin[0][4]=NB_2;}
    else
        {rule_asin[0][4]=PB_1;}

    if (NB_1>NS_2)
        {rule_asin[1][0]=NS_2;}
    else
        {rule_asin[1][0]=NB_1;}
    if (NS_1>NS_2)
        {rule_asin[1][1]=NS_2;}
    else
        {rule_asin[1][1]=NS_1;}
    if (ZE_1>NS_2)
        {rule_asin[1][2]=NS_2;}
    else
        {rule_asin[1][2]=ZE_1;}
    if (PS_1>NS_2)
        {rule_asin[1][3]=NS_2;}
    else
        {rule_asin[1][3]=PS_1;}
    if (PB_1>NS_2)
        {rule_asin[1][4]=NS_2;}
    else
        {rule_asin[1][4]=PB_1;}
}

```



```

if (NB_1>ZE_2)
  {rule_asin[2][0]=ZE_2;}
else
  {rule_asin[2][0]=NB_1;}
if (NS_1>ZE_2)
  {rule_asin[2][1]=ZE_2;}
else
  {rule_asin[2][1]=NS_1;}
if (ZE_1>ZE_2)
  {rule_asin[2][2]=ZE_2;}
else
  {rule_asin[2][2]=ZE_1;}
if (PS_1>ZE_2)
  {rule_asin[2][3]=ZE_2;}
else
  {rule_asin[2][3]=PS_1;}
if (PB_1>ZE_2)
  {rule_asin[2][4]=ZE_2;}
else
  {rule_asin[2][4]=PB_1;}

if (NB_1>PS_2)
  {rule_asin[3][0]=PS_2;}
else
  {rule_asin[3][0]=NB_1;}
if (NS_1>PS_2)
  {rule_asin[3][1]=PS_2;}
else
  {rule_asin[3][1]=NS_1;}
if (ZE_1>PS_2)
  {rule_asin[3][2]=PS_2;}
else
  {rule_asin[3][2]=ZE_1;}
if (PS_1>PS_2)
  {rule_asin[3][3]=PS_2;}
else
  {rule_asin[3][3]=PS_1;}
if (PB_1>PS_2)
  {rule_asin[3][4]=PS_2;}
else
  {rule_asin[3][4]=PB_1;}

if (NB_1>PB_2)
  {rule_asin[4][0]=PB_2;}
else
  {rule_asin[4][0]=NB_1;}
if (NS_1>PB_2)
  {rule_asin[4][1]=PB_2;}
else
  {rule_asin[4][1]=NS_1;}
if (ZE_1>PB_2)
  {rule_asin[4][2]=PB_2;}

```



```

else
    {rule_asin[4][2]=ZE_1;}
if (PS_1>PB_2)
    {rule_asin[4][3]=PB_2;}
else
    {rule_asin[4][3]=PS_1;}
if (PB_1>PB_2)
    {rule_asin[4][4]=PB_2;}
else
    {rule_asin[4][4]=PB_1;}

//Mencari DOM output (MAX)
VB_2= rule_asin[4][0];

if (rule_asin[2][0]>rule_asin[3][0])
    {B_2=rule_asin[2][0];}
else if (rule_asin[2][0] < rule_asin[3][0])
    {B_2=rule_asin[3][0];}

S_2=rule_asin[2][1];
if (S_2<rule_asin[3][1])
    {S_2=rule_asin[3][1];}
if (S_2<rule_asin[4][1])
    {S_2=rule_asin[4][1];}

Z_2=rule_asin[0][0];
for (i=0;i<=1;i++)
{
    for (j=0;j<=4;j++)
    {
        if (Z_2<rule_asin[i][j])
            {Z_2=rule_asin[i][j];}
    }
}
for (i=2;i<=4;i++)
{
    for (j=2;j<=4;j++)
    {
        if (Z_2<rule_asin[i][j])
            {Z_2=rule_asin[i][j];}
    }
}
}

void defuzzifikasi ()
{
    output_tawar=
(N_Tengah_out[0]*Z_1+N_Tengah_out[1]*VS_1+N_Tengah_out[2]*S_1+N_Tengah_out[3]*B_1
+N_Tengah_out[4]*VB_1)/(Z_1+VS_1+S_1+B_1+VB_1);
    output_asin=
(N_Tengah_out[0]*Z_2+N_Tengah_out[1]*VS_2+N_Tengah_out[2]*S_2+N_Tengah_out[3]*B_2
+N_Tengah_out[4]*VB_2)/(Z_2+VS_2+S_2+B_2+VB_2);
    analogWrite(in_1,output_tawar);
}

```



```
analogWrite(in_3,output_asin);  
Z_1=0;VS_1=0;S_1=0;B_1=0;VB_1=0;  
Z_2=0;VS_2=0;S_2=0;B_2=0;VB_2=0;  
}
```

```
void BACA_SUHU ()  
{  
  if ( !ds.search(addr))  
  {  
    ds.reset_search();  
    delay(250);  
    return;  
  }  
}
```

```
ds.reset();  
ds.select(addr);  
ds.write(0x44,1);  
delay(1000);
```

```
present = ds.reset();  
ds.select(addr);  
ds.write(0xBE);
```

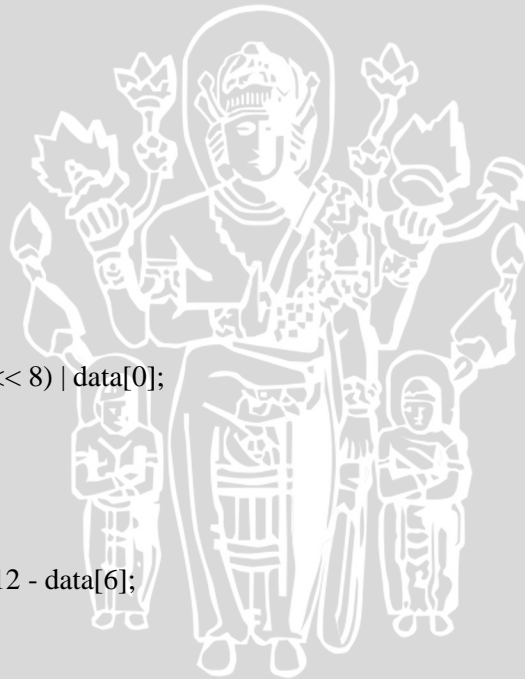
```
for ( i = 0; i < 9; i++)  
{  
  data[i] = ds.read();  
}
```

```
unsigned int raw = (data[1] << 8) | data[0];  
if (type_s)  
{  
  raw = raw << 3;  
  if (data[7] == 0x10)  
  {  
    raw = (raw & 0xFFF0) + 12 - data[6];  
  }  
}
```

```
else  
{  
  byte cfg = (data[4] & 0x60);  
  if (cfg == 0x00) raw = raw << 3;  
  else if (cfg == 0x20) raw = raw << 2;  
  else if (cfg == 0x40) raw = raw << 1;  
}
```

```
//nilai suhu  
celsius = (float)raw / 16.0;  
fahrenheit = celsius * 1.8 + 32.0;  
}
```

```
void PROSES_SUHU ()  
{
```



```

//Algoritma
if (celsius < 25)
{
  digitalWrite (relay_heater, HIGH);
  while (celsius <=26.5)
  {BACA_SUHU (); SALINITAS (); LCD ();}
  digitalWrite (relay_heater, LOW);
}

else if (celsius > 28)
{
  digitalWrite (relay_fan, HIGH);
  while (celsius >=26.5)
  {BACA_SUHU (); SALINITAS (); LCD ();}
  digitalWrite (relay_fan, LOW);
}
LCD ();
}

void LCD ()
{
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Temp: ");
  lcd.print(celsius);
  lcd.print(" oC");
  Serial.print("suhu = ");
  Serial.println(celsius);

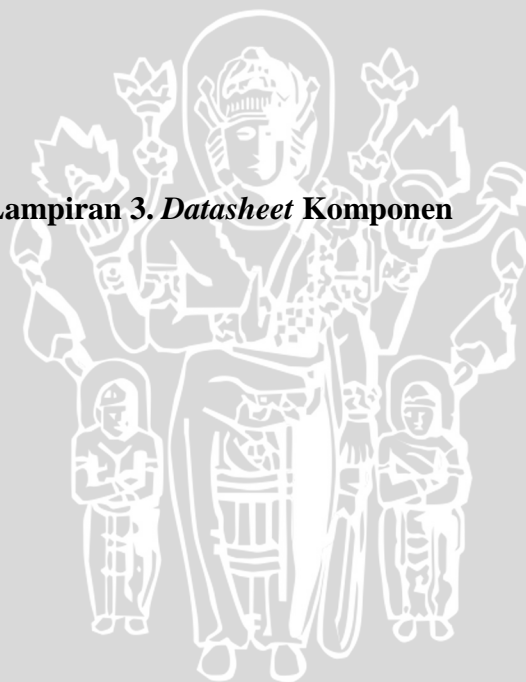
  //LCD for saline test
  lcd.setCursor(0,1);
  lcd.print("Salinitas: ");
  lcd.print(salinitas);
  Serial.print("suhu = ");
  Serial.println(salinitas);
  delay(1000);
}

```



UNIVERSITAS BRAWIJAYA

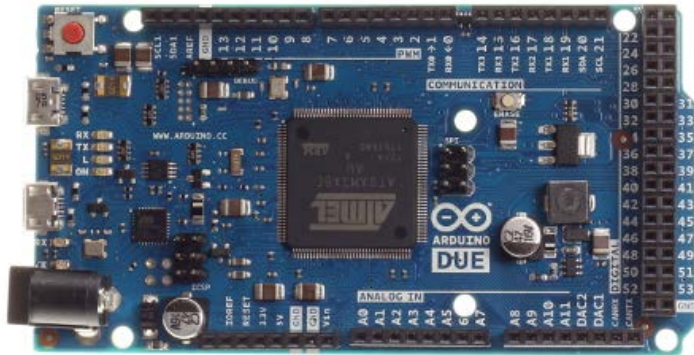
Lampiran 3. *Datasheet* Komponen



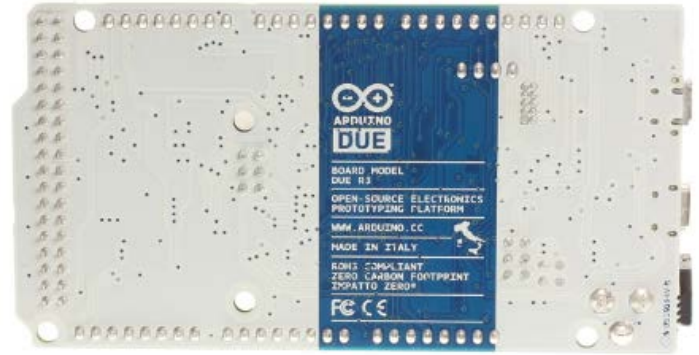




Arduino Due



Arduino Due Front



Arduino Due Back

Overview

The Arduino Due is a microcontroller board based on the Atmel SAM3X8E ARM Cortex-M3 CPU ([datasheet](#)). It is the first Arduino board based on a 32-bit ARM core microcontroller. It has 54 digital input/output pins (of which 12 can be used as PWM outputs), 12 analog inputs, 4 UARTs (hardware serial ports), a 84 MHz clock, an USB OTG capable connection, 2 DAC (digital to analog), 2 TWI, a power jack, an SPI header, a JTAG header, a reset button and an erase button.

Warning: Unlike other Arduino boards, the Arduino Due board runs at 3.3V. The maximum voltage that the I/O pins can tolerate is 3.3V. Providing higher voltages, like 5V to an I/O pin could damage the board.

The board contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Due is compatible with all Arduino shields that work at 3.3V and are compliant with the 1.0 Arduino pinout.

The Due follows the 1.0 pinout:

- ✦ TWI: SDA and SCL pins that are near to the AREF pin.
- ✦ The IOREF pin which allows an attached shield with the proper configuration to adapt to the voltage provided by the board. This enables shield compatibility with a 3.3V board like the Due and AVR-based boards which operate at 5V.

- ✦ An unconnected pin, reserved for future use.

The Due has a dedicated forum for discussing the board.

ARM Core benefits

The Due has a 32-bit ARM core that can outperform typical 8-bit microcontroller boards. The most significant differences are:

- ✦ A 32-bit core, that allows operations on 4 bytes wide data within a single CPU clock. (for more information look int type page).
- ✦ CPU Clock at 84Mhz.
- ✦ 96 KBytes of SRAM.
- ✦ 512 KBytes of Flash memory for code.
- ✦ a DMA controller, that can relieve the CPU from doing memory intensive tasks.

Schematic, Reference Design & Pin Mapping

EAGLE files: [arduino-Due-reference-design.zip](#)

Schematic: [arduino-Due-schematic.pdf](#)

Pin Mapping: [SAM3X Pin Mapping page](#)

Summary

Microcontroller	AT91SAM3X8E
Operating Voltage	3.3V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 12 provide PWM output)
Analog Input Pins	12
Analog Outputs Pins	2 (DAC)
Total DC Output Current on all I/O lines	130 mA
DC Current for 3.3V Pin	800 mA
DC Current for 5V Pin	800 mA
Flash Memory	512 KB all available for the user applications
SRAM	96 KB (two banks: 64KB and 32KB)
Clock Speed	84 MHz

Power

The Arduino Due can be powered via the USB connector or with an external power supply. The power source is selected automatically.

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

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

The power pins are as follows:

- ✦ **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the

USB connection or other regulated power source). You can supply voltage through this pin, or if supplying voltage via the power jack, access it through this pin.

+ **5V**. 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.

+ **3.3V**. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 800 mA. This regulator also provides the power supply to the SAM3X microcontroller.

+ **GND**. Ground pins.

+ **IOREF**. This pin on the Arduino 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 SAM3X has 512 KB (2 blocks of 256 KB) of flash memory for storing code. The bootloader is preburned in factory from Atmel and is stored in a dedicated ROM memory. The available SRAM is 96 KB in two contiguous bank of 64 KB and 32 KB. All the available memory (Flash, RAM and ROM) can be accessed directly as a flat addressing space.

It is possible to erase the Flash memory of the SAM3X with the onboard erase button. This will remove the currently loaded sketch from the MCU. To erase, press and hold the Erase button for a few seconds while the board is powered.

Input and Output

+ Digital I/O: pins from 0 to 53

Each of the 54 digital pins on the Due can be used as an input or output, using `pinMode()`, `digitalWrite()`, and `digitalRead()` functions. They operate at 3.3 volts. Each pin can provide (source) a current of 3 mA or 15 mA, depending on the pin, or receive (sink) a current of 6 mA or 9 mA, depending on the pin. They also have an internal pull-up resistor (disconnected by default) of 100 KOhm. 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 (with 3.3 V level). Pins 0 and 1 are connected to the corresponding pins of the ATmega16U2 USB-to-TTL Serial chip.

+ PWM: Pins 2 to 13

Provide 8-bit PWM output with the `analogWrite()` function. the resolution of the PWM can be changed with the `analogWriteResolution()` function.

+ SPI: SPI header (ICSP header on other Arduino boards)

These pins support SPI communication using the [SPI library](#). The SPI pins are broken out on the central 6-pin header, which is physically compatible with the Uno, Leonardo and Mega2560. The SPI header can be used only to communicate with other SPI devices, not for programming the SAM3X with the In-Circuit-Serial-Programming technique. The SPI of the Due has also advanced features that can be used with the [Extended SPI methods for Due](#).

+ CAN: CANRX and CANTX

These pins support the CAN communication protocol but are not yet supported by Arduino APIs.

+ "L" LED: 13

There is a built-in LED connected to digital pin 13. When the pin is HIGH, the LED is on, when the pin is LOW, it's off. It is also possible to dim the LED because the digital pin 13 is also a PWM output.

+ TWI 1: 20 (SDA) and 21 (SCL)

+ TWI 2: SDA1 and SCL1.

Support TWI communication using the [Wire library](#).

+ Analog Inputs: pins from A0 to A11

The Due has 12 analog inputs, each of which can provide 12 bits of resolution (i.e. 4096 different values). By default, the resolution of the readings is set at 10 bits, for compatibility with other Arduino boards. It is possible to change the resolution of the ADC with [analogReadResolution\(\)](#). The Due's analog inputs pins measure from ground to a maximum value of 3.3V. Applying more than 3.3V on the Due's pins will damage the SAM3X chip. The [analogReference\(\)](#) function is ignored on the Due.

The AREF pin is connected to the SAM3X analog reference pin through a resistor bridge. To use the AREF pin, resistor BR1 must be desoldered from the PCB.

+ DAC1 and DAC2

These pins provides true analog outputs with 12-bits resolution (4096 levels) with the [analogWrite\(\)](#) function. These pins can be used to create an audio output using the [Audio library](#).

Other pins on the board:

+ AREF

Reference voltage for the analog inputs. Used with [analogReference\(\)](#).

+ Reset

Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Communication

The Arduino Due has a number of facilities for communicating with a computer, another Arduino or other microcontrollers, and different devices like phones, tablets, cameras and so on. The SAM3X provides one hardware UART and three hardware USARTs for TTL (3.3V) serial communication.

The Programming port is connected to an ATmega16U2, which provides a virtual COM port to software on a connected computer (To recognize the device, Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically.). The 16U2 is also connected to the SAM3X hardware UART. Serial on pins RX0 and TX0 provides Serial-to-USB communication for programming the board through the ATmega16U2 microcontroller. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega16U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

The Native USB port is connected to the SAM3X. It allows for serial (CDC) communication over USB. This provides a serial connection to the Serial Monitor or other applications on your computer. It also enables the Due to emulate a USB mouse or keyboard to an attached computer. To use these features, see the [Mouse and Keyboard library reference pages](#).

The Native USB port can also act as a USB host for connected peripherals such as mice, keyboards, and smartphones. To use these features, see the [USBHost reference pages](#).

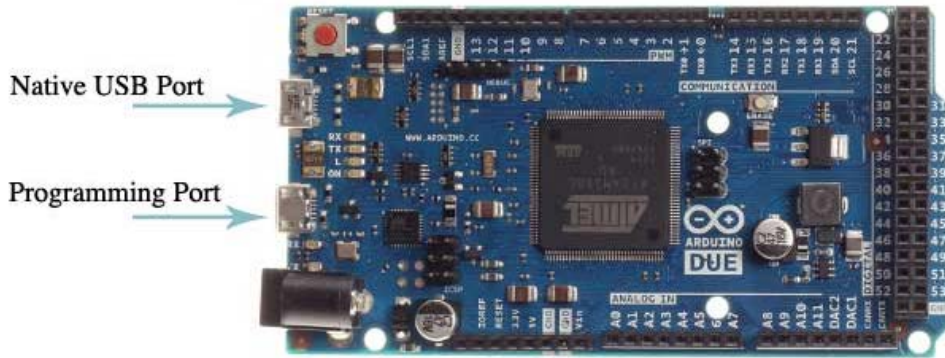
The SAM3X also supports TWI and SPI communication. The Arduino software includes a [Wire library](#) to simplify use of the TWI bus; see the [documentation](#) for details. For SPI communication, use the [SPI library](#).

Programming

The Arduino Due can be programmed with the Arduino software ([download](#)). For details, see the [reference](#) and [tutorials](#).

Uploading sketches to the SAM3X is different than the AVR microcontrollers found in other Arduino boards because the flash memory needs to be erased before being re-programmed. Upload to the chip is managed by ROM on the SAM3X,

which is run only when the chip's flash memory is empty.



Either of the USB ports can be used for programming the board, though it is recommended to use the Programming port due to the way the erasing of the chip is handled :

- ✦ Programming port: To use this port, select "Arduino Due (Programming Port)" as your board in the Arduino IDE. Connect the Due's programming port (the one closest to the DC power jack) to your computer. The programming port uses the 16U2 as a USB-to-serial chip connected to the first UART of the SAM3X (RX0 and TX0). The 16U2 has two pins connected to the Reset and Erase pins of the SAM3X. Opening and closing the Programming port connected at 1200bps triggers a "hard erase" procedure of the SAM3X chip, activating the Erase and Reset pins on the SAM3X before communicating with the UART. This is the recommended port for programming the Due. It is more reliable than the "soft erase" that occurs on the Native port, and it should work even if the main MCU has crashed.

- ✦ Native port: To use this port, select "Arduino Due (Native USB Port)" as your board in the Arduino IDE. The Native USB port is connected directly to the SAM3X. Connect the Due's Native USB port (the one closest to the reset button) to your computer. Opening and closing the Native port at 1200bps triggers a 'soft erase' procedure: the flash memory is erased and the board is restarted with the bootloader. If the MCU crashed for some reason it is likely that the soft erase procedure won't work as this procedure happens entirely in software on the SAM3X. Opening and closing the native port at a different baudrate will not reset the SAM3X.

Unlike other Arduino boards which use avrdude for uploading, the Due relies on bossac.

The ATmega16U2 firmware source code is available [in the Arduino repository](#). You can use the ISP header with an external programmer (overwriting the DFU bootloader). See [this user-contributed tutorial](#) for more information.

USB Overcurrent Protection

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

Physical Characteristics and Shield Compatibility

The maximum length and width of the Arduino Due PCB are 4 and 2.1 inches respectively, with the USB connectors 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 Arduino Due is designed to be compatible with most shields designed for the Uno, Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and "ICSP" (SPI) header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external

interrupts 0 and 1 (pins 2 and 3 respectively). Please note that I^2C is not located on the same pins on the Due (20 and 21) as the Duemilanove / Diecimila (analog inputs 4 and 5).

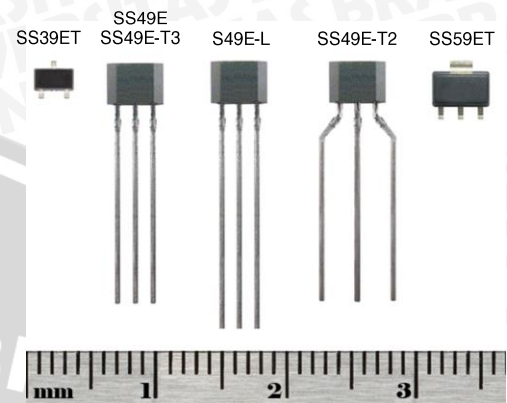
[+](#) Share | [✉](#) [★](#) [🖨](#) [f](#) [t](#)

[©Arduino](#) | [Edit Page](#) | [Page History](#) | [Printable View](#) | [All Recent Site Changes](#)



SS39ET/SS49E/SS59ET Series

Linear Hall-Effect Sensor ICs



DESCRIPTION

The SS39ET/SS49E/SS59ET Series Low-cost Linear Hall-effect sensors are small, versatile devices that are operated by the magnetic field from a permanent magnet or an electromagnet.

The linear sourcing output voltage is set by the supply voltage and varies in proportion to the strength of the magnetic field. Low voltage capability as low as 2.7 Vdc and reduced current consumption of only 6 mA typically at 5 Vdc help make this product energy efficient.

The integrated circuitry features low noise output, which makes it unnecessary to use external filtering. It also includes thin film resistors to provide increased temperature stability and accuracy. These linear Hall-effect sensors have an operating temperature range of -40 °C to 100 °C [-40 °F to 212 °F], appropriate for commercial, consumer, and industrial environments.

VALUE PROPOSITION

- Designed and manufactured for cost competitiveness
- All packages can be supplied on tape for automated, lower-cost assembly
- SS39ET's small footprint takes up less space on the PC board, typically allowing for more components
- Interfaces with many electrical components without buffering
- Thermally balanced integrated circuit for stable operation over a full temperature range

FEATURES AND BENEFITS

- **Miniature and subminiature construction:** Designed for compact designs with tight space requirements
- **Energy efficient:** Low current consumption of 6 mA at 5 Vdc
- **Easy PC board interface:** Single current sourcing output for common electronic circuits
- **Circuit design flexibility:** Voltage range of 2.7 Vdc to 6.5 Vdc
- **Low noise output:** Virtually eliminates the need for filtering
- **Stable output:** Thin film resistors improve accuracy
- **Wide range of environments:** Temperature range of -40 °C to 100 °C [-40 °F to 212 °F]
- **Application flexibility:** Responds to either positive or negative Gauss

POTENTIAL APPLICATIONS

- Industrial:**
- Basic current sensing for motor load monitoring, detection
 - Anti-tampering magnetic field sensor in smart remote utility meters
 - Pump control in heavy-duty equipment and household appliances
 - Simple linear or angular displacement sensing
 - Handlebar/throttle position sensing in e-bikes and scooters
 - Current sensing in appliances
 - Speed adjustment trigger in tools and appliances
 - Magnetic code reading in safes, security and building access control systems
- Medical:**
- Position sensing in infusion pumps

SS39ET/SS49E/SS59ET Series

Table 1. Operating Characteristics ($V_s = 5.0\text{ V}$, $T_A = -40\text{ }^\circ\text{C}$ to $85\text{ }^\circ\text{C}$ [$-40\text{ }^\circ\text{F}$ to $185\text{ }^\circ\text{F}$], except where noted.)

Characteristic	Condition	Min.	Typ.	Max.	Unit
Output type	–		linear, sourcing		–
Magnetics type	–		analog		–
Supply voltage	–	2.7	–	6.5	Vdc
Supply current	25 °C [77 °F]	–	6	10	mA
Output voltage	–	1.0	1.4	1.75	mV/Gauss
Output current	$V_s > 3.0\text{ V}$	1.0	1.5	–	mA
Null	0 Gauss, 25 °C	2.25	2.50	2.75	Vdc
Output voltage span	–	1.05 to ($V_s - 1.05$)	0.95 to ($V_s - 0.95$)	–	Vdc
Magnetic range	–	± 650	± 1000	–	Gauss
Sensitivity	25 °C	1.0	1.4	1.75	mV/Gauss
Operating temperature	–	-40 [-40]	–	100 [212]	°C [°F]
Temperature error:					
Null drift	–	-0.10	–	0.10	%/°C
Sensitivity drift	$\geq 25\text{ }^\circ\text{C}$	-0.15	–	0.05	
	$< 25\text{ }^\circ\text{C}$	-0.04	–	0.185	
Linearity	–	–	-0.7	–	% of span
Response time	–	–	3	–	μs

Table 2. Absolute Maximum Ratings

Characteristic	Parameter
Supply voltage (V_s)	-0.5 Vdc to 8.0 Vdc
Output current	10 mA
Storage temperature	-55 °C to 165 °C [-67 °F to 329 °F]

Figure 1. Current Sourcing Output Block Diagram

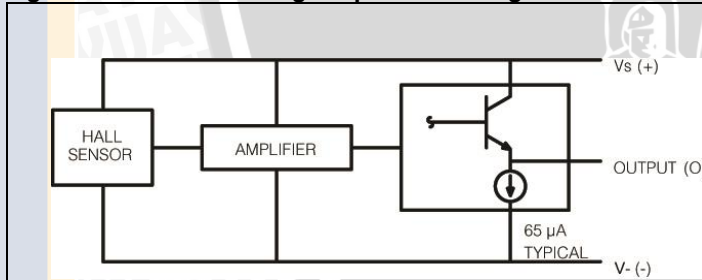
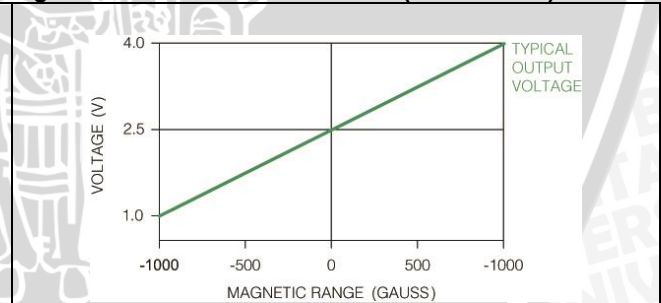


Figure 2. Transfer Characteristics ($V_s = 5.0\text{ Vdc}$)



NOTICE

Absolute maximum ratings are the extreme limits that the device will withstand without damage to the device. However, the electrical and mechanical characteristics are not guaranteed as the maximum limits (above recommended operating conditions) are approached, nor will the device necessarily operate at absolute maximum ratings.



Linear Hall-Effect Sensor ICs

Figure 3. Null Shift versus Temperature

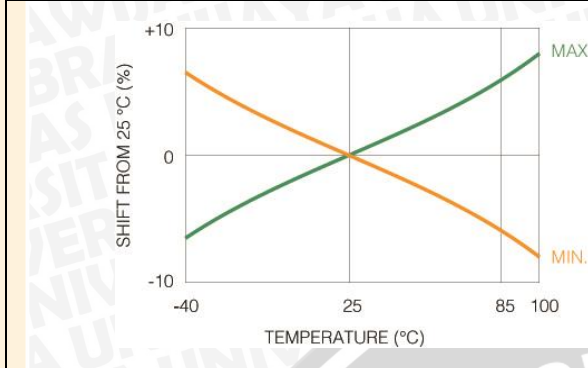


Figure 4. Sensitivity Shift versus Temperature

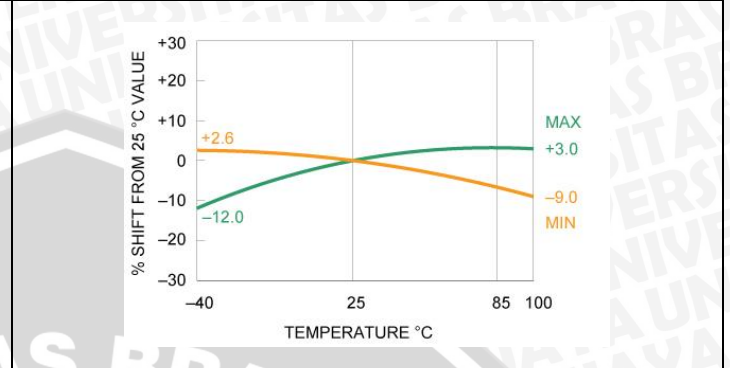


Figure 5. Typical Frequency Response

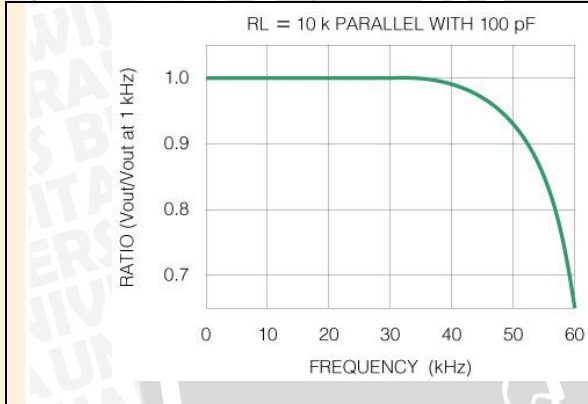


Figure 6. Supply Current versus Temperature

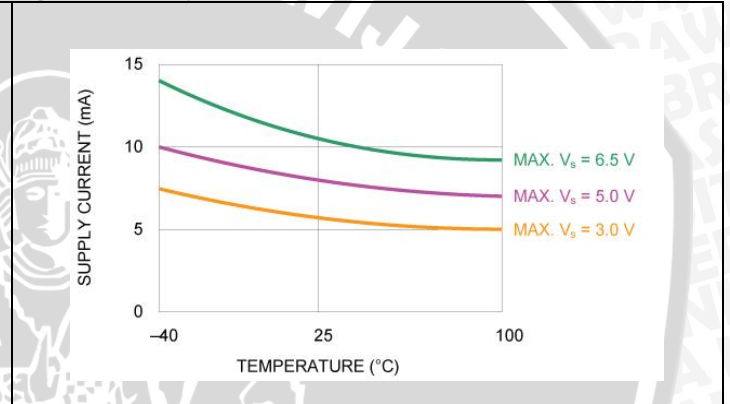
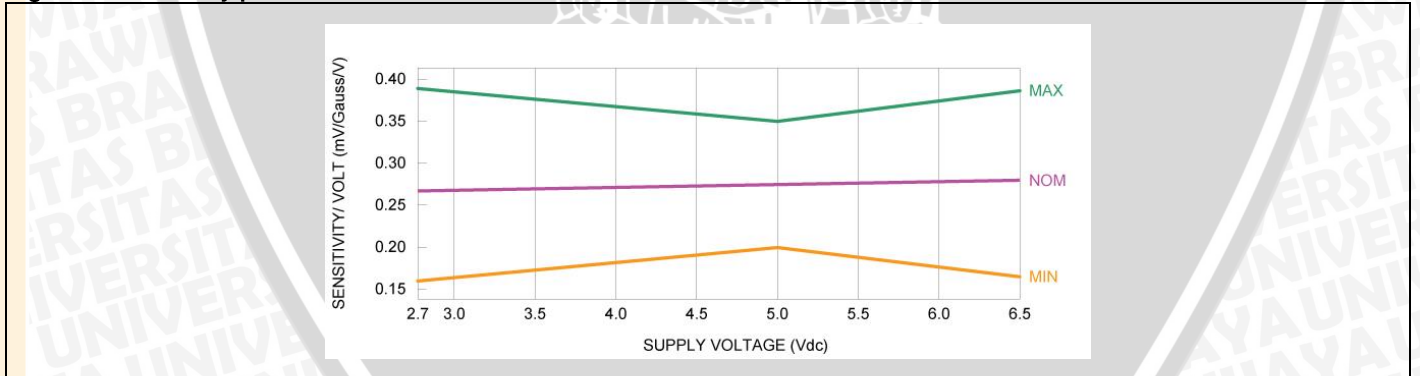


Figure 7. Sensitivity per Volt versus V_s



SS39ET/SS49E/SS59ET Series

Figure 8. SS39ET Mounting Dimensions (for reference only, mm/[in])

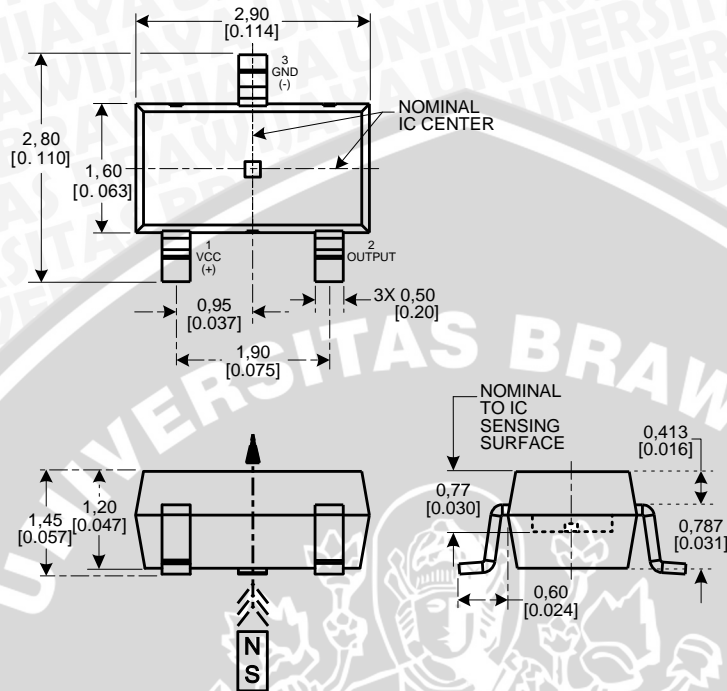
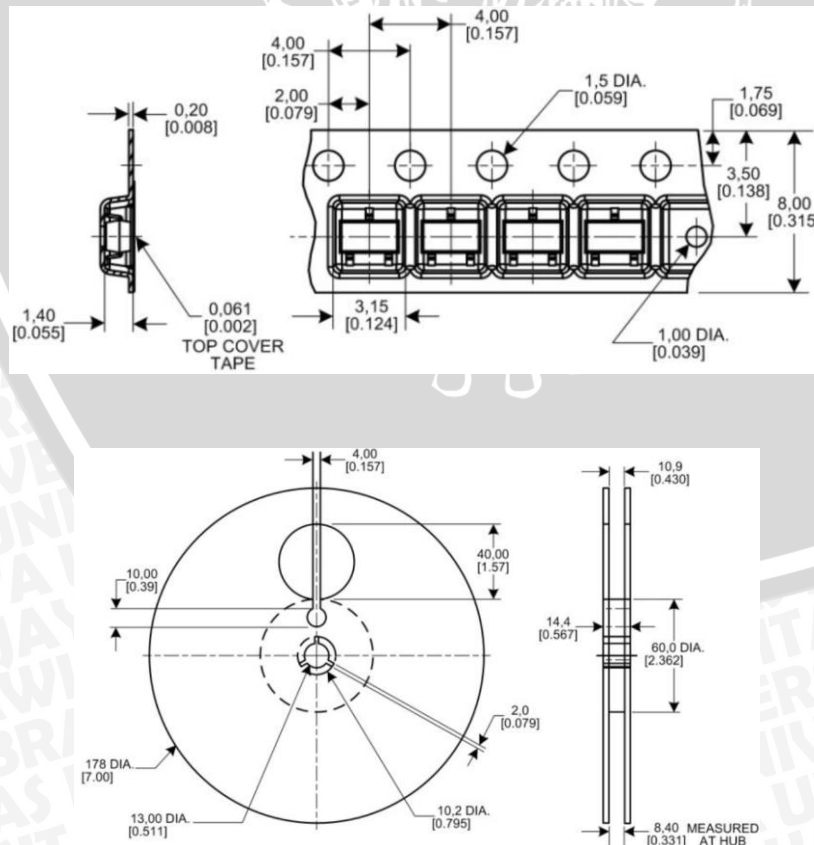
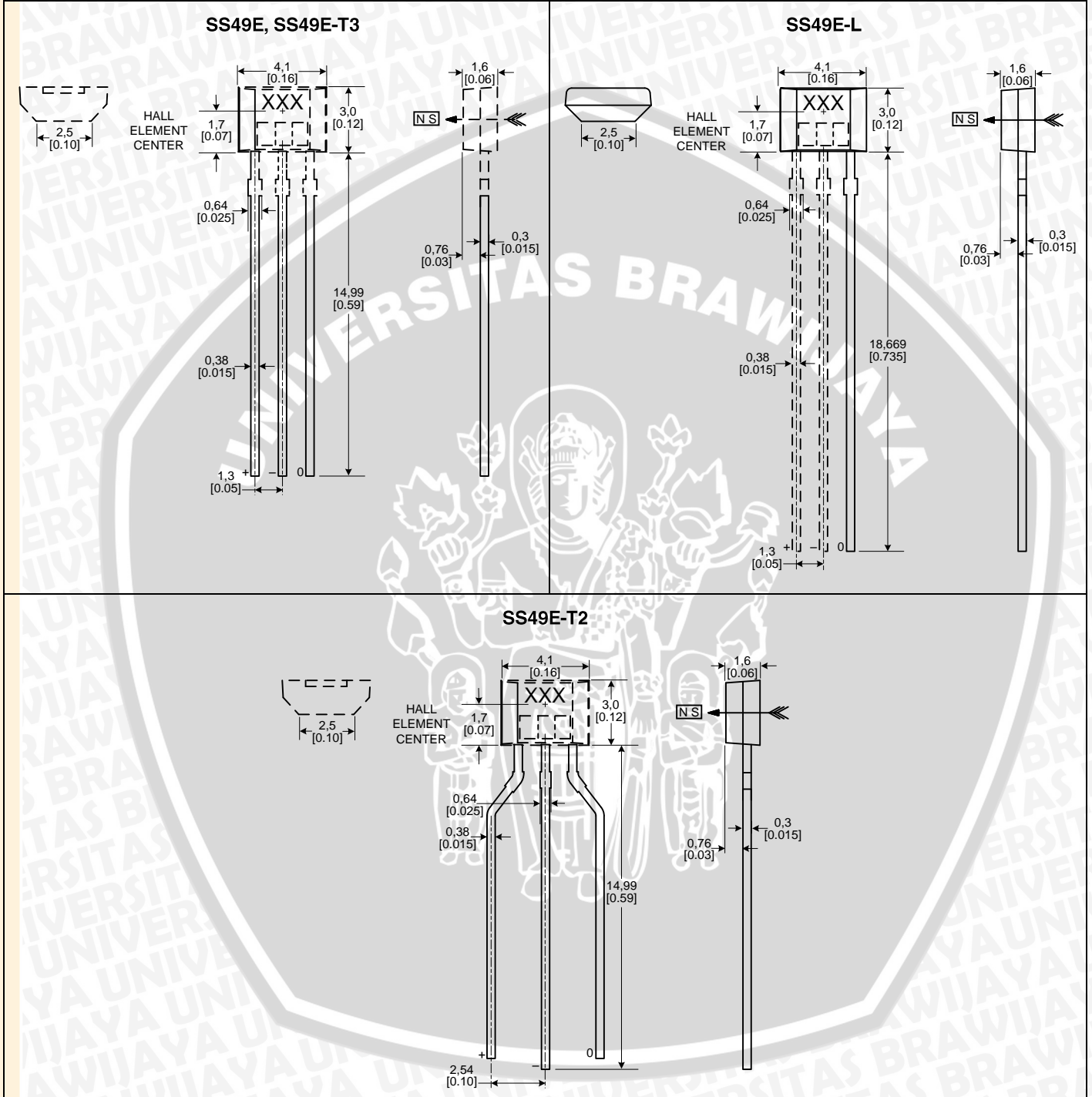


Figure 9. SS39ET Tape and Reel Dimensions (for reference only, mm/[in])



Linear Hall-Effect Sensor ICs

Figure 10. SS49E Mounting Dimensions (for reference only, mm/[in])



SS39ET/SS49E/SS59ET Series

Figure 11. Tape Style T2

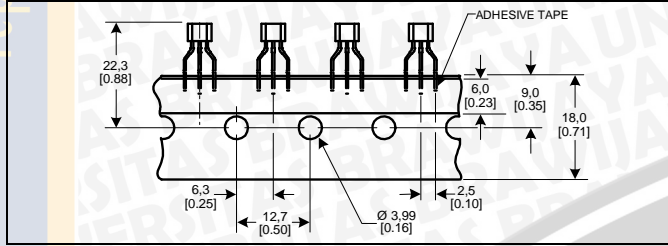


Figure 12. Tape Style T3

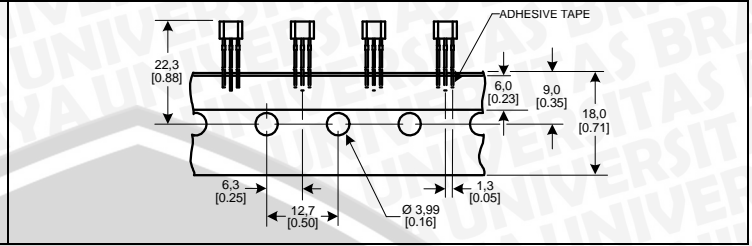


Figure 13. SS59ET Mounting Dimensions (for reference only, mm/[in])

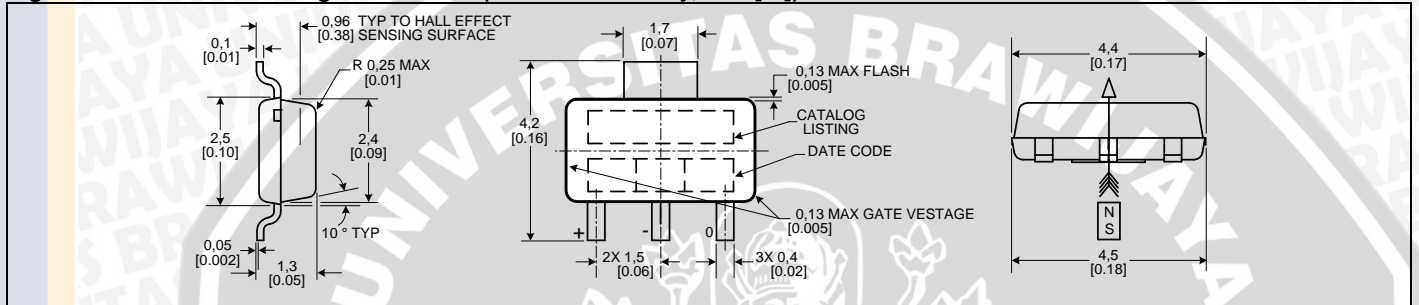
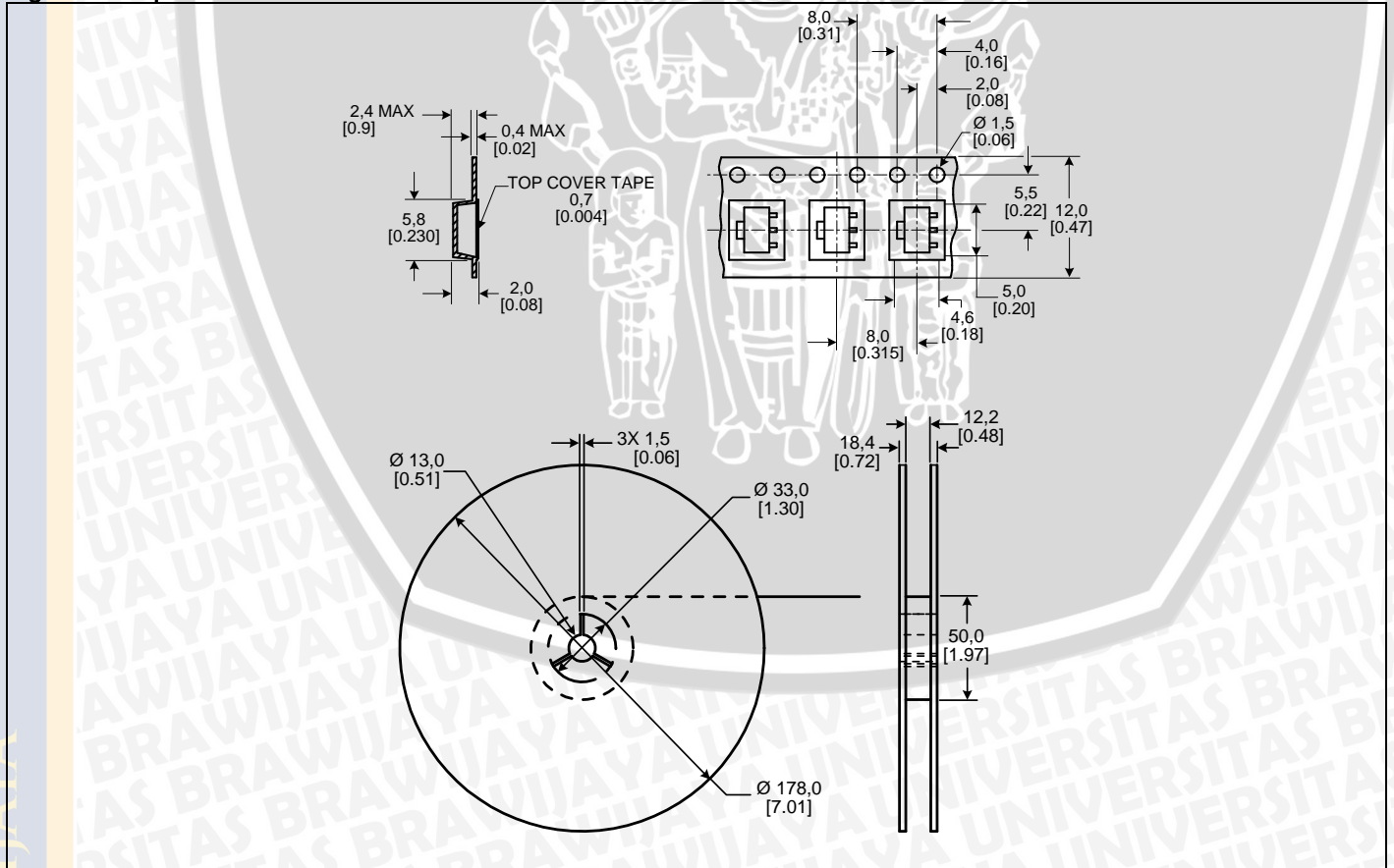


Figure 14. Tape and Reel Dimensions



Order Guide

Catalog Listing	Description
SS39ET	Linear Hall-effect sensor IC, SOT-23 tape and reel packaging (3000 units per reel)
SS49E	Linear Hall-effect sensor IC, straight leads, 14,99 mm [0.59 in] lead length, 1,30 mm [0.05 in] spacing, flat TO-92-style, bulk packaging (1000 units per bag)
SS49E-F	Linear Hall-effect sensor IC, formed leads, 14,99 mm [0.59 in] lead length, 2,54 mm [0.10 in] spacing, flat TO-92-style, bulk packaging (1000 units per bag)
SS49E-L	Linear Hall-effect sensor IC, straight leads, 18,67 mm [0.75 in] lead length, 1,30 mm [0.05 in] spacing, flat TO-92-style, bulk packaging (1000 units per bag)
SS49E-T2	Linear Hall-effect sensor IC, flat TO-92-style, tape-in-box (ammopack) version with formed leads (5000 units per box)
SS49E-T3	Linear Hall-effect sensor IC, flat TO-92-style, tape-in-box (ammopack) version with straight leads (5000 units per box)
SS59ET	Linear Hall-effect sensor IC, SOT-89 package, tape and reel packaging (1000 units per reel)

Note: Products ordered in bulk packaging (plastic bags) may not have perfectly straight leads as a result of normal handling and shipping operations. Please order a tape packaging option for applications with critical requirements for straight leads.

WARNING

PERSONAL INJURY

DO NOT USE these products as safety or emergency stop devices or in any other application where failure of the product could result in personal injury.

Failure to comply with these instructions could result in death or serious injury.

WARRANTY/REMEDY

Honeywell warrants goods of its manufacture as being free of defective materials and faulty workmanship. Honeywell's standard product warranty applies unless agreed to otherwise by Honeywell in writing; please refer to your order acknowledgement or consult your local sales office for specific warranty details. If warranted goods are returned to Honeywell during the period of coverage, Honeywell will repair or replace, at its option, without charge those items it finds defective. **The foregoing is buyer's sole remedy and is in lieu of all other warranties, expressed or implied, including those of merchantability and fitness for a particular purpose. In no event shall Honeywell be liable for consequential, special, or indirect damages.**

While we provide application assistance personally, through our literature and the Honeywell web site, it is up to the customer to determine the suitability of the product in the application.

Specifications may change without notice. The information we supply is believed to be accurate and reliable as of this printing. However, we assume no responsibility for its use.

WARNING

MISUSE OF DOCUMENTATION

- The information presented in this product sheet is for reference only. Do not use this document as a product installation guide.
- Complete installation, operation, and maintenance information is provided in the instructions supplied with each product.

Failure to comply with these instructions could result in death or serious injury.

SALES AND SERVICE

Honeywell serves its customers through a worldwide network of sales offices, representatives and distributors. For application assistance, current specifications, pricing or name of the nearest Authorized Distributor, contact your local sales office or:

E-mail: info.sc@honeywell.com

Internet: sensing.honeywell.com

Phone and Fax:

Asia Pacific	+65 6355-2828 +65 6445-3033 Fax
Europe	+44 (0) 1698 481481 +44 (0) 1698 481676 Fax
Latin America	+1-305-805-8188 +1-305-883-8257 Fax
USA/Canada	+1-800-537-6945 +1-815-235-6847 +1-815-235-6545 Fax

Sensing and Control

Honeywell

1985 Douglas Drive North

Golden Valley, MN 55422

www.honeywell.com/sensing

005850-3-EN
March 2013

© 2013 Honeywell International Inc. All rights reserved.

Honeywell



DS18B20

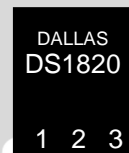
Programmable Resolution 1-Wire[®] Digital Thermometer

www.dalsemi.com

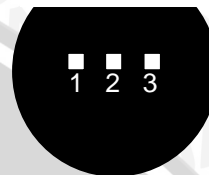
FEATURES

- Unique 1-Wire interface requires only one port pin for communication
- Multidrop capability simplifies distributed temperature sensing applications
- Requires no external components
- Can be powered from data line. Power supply range is 3.0V to 5.5V
- Zero standby power required
- Measures temperatures from -55°C to $+125^{\circ}\text{C}$. Fahrenheit equivalent is -67°F to $+257^{\circ}\text{F}$
- $\pm 0.5^{\circ}\text{C}$ accuracy from -10°C to $+85^{\circ}\text{C}$
- Thermometer resolution is programmable from 9 to 12 bits
- Converts 12-bit temperature to digital word in 750 ms (max.)
- User-definable, nonvolatile temperature alarm settings
- Alarm search command identifies and addresses devices whose temperature is outside of programmed limits (temperature alarm condition)
- Applications include thermostatic controls, industrial systems, consumer products, thermometers, or any thermally sensitive system

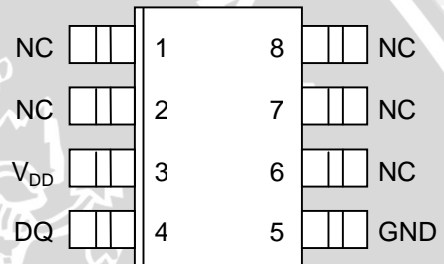
PIN ASSIGNMENT



BOTTOM VIEW



DS18B20 To-92
Package



DS18B20Z
8-Pin SOIC (150 mil)

PIN DESCRIPTION

- GND - Ground
 DQ - Data In/Out
 V_{DD} - Power Supply Voltage
 NC - No Connect

DESCRIPTION

The DS18B20 Digital Thermometer provides 9 to 12-bit (configurable) temperature readings which indicate the temperature of the device.

Information is sent to/from the DS18B20 over a 1-Wire interface, so that only one wire (and ground) needs to be connected from a central microprocessor to a DS18B20. Power for reading, writing, and performing temperature conversions can be derived from the data line itself with no need for an external power source.

Because each DS18B20 contains a unique silicon serial number, multiple DS18B20s can exist on the same 1-Wire bus. This allows for placing temperature sensors in many different places. Applications where this feature is useful include HVAC environmental controls, sensing temperatures inside buildings, equipment or machinery, and process monitoring and control.

DETAILED PIN DESCRIPTION Table 1

PIN 8PIN SOIC	PIN TO92	SYMBOL	DESCRIPTION
5	1	GND	Ground.
4	2	DQ	Data Input/Output pin. For 1-Wire operation: Open drain. (See “Parasite Power” section.)
3	3	V _{DD}	Optional V_{DD} pin. See “Parasite Power” section for details of connection. V _{DD} must be grounded for operation in parasite power mode.

DS18B20Z (8-pin SOIC): All pins not specified in this table are not to be connected.

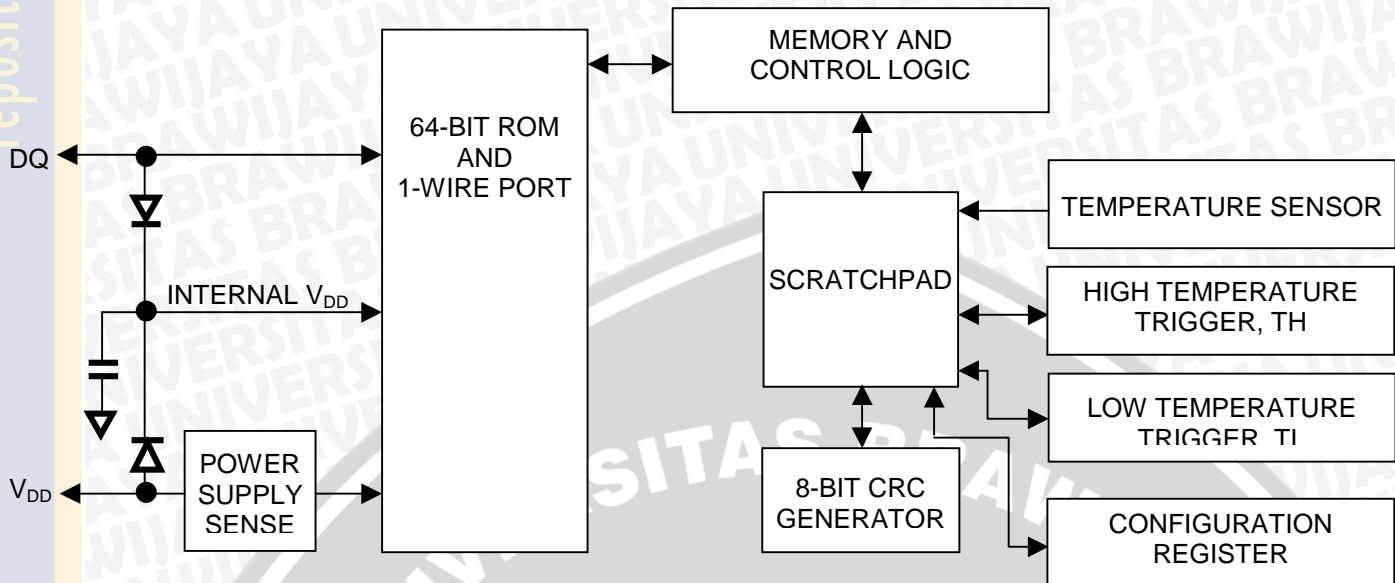
OVERVIEW

The block diagram of Figure 1 shows the major components of the DS18B20. The DS18B20 has four main data components: 1) 64-bit lasered ROM, 2) temperature sensor, 3) nonvolatile temperature alarm triggers TH and TL, and 4) a configuration register. The device derives its power from the 1-Wire communication line by storing energy on an internal capacitor during periods of time when the signal line is high and continues to operate off this power source during the low times of the 1-Wire line until it returns high to replenish the parasite (capacitor) supply. As an alternative, the DS18B20 may also be powered from an external 3 volt - 5.5 volt supply.

Communication to the DS18B20 is via a 1-Wire port. With the 1-Wire port, the memory and control functions will not be available before the ROM function protocol has been established. The master must first provide one of five ROM function commands: 1) Read ROM, 2) Match ROM, 3) Search ROM, 4) Skip ROM, or 5) Alarm Search. These commands operate on the 64-bit lasered ROM portion of each device and can single out a specific device if many are present on the 1-Wire line as well as indicate to the bus master how many and what types of devices are present. After a ROM function sequence has been successfully executed, the memory and control functions are accessible and the master may then provide any one of the six memory and control function commands.

One control function command instructs the DS18B20 to perform a temperature measurement. The result of this measurement will be placed in the DS18B20's scratch-pad memory, and may be read by issuing a memory function command which reads the contents of the scratchpad memory. The temperature alarm triggers TH and TL consist of 1 byte EEPROM each. If the alarm search command is not applied to the DS18B20, these registers may be used as general purpose user memory. The scratchpad also contains a configuration byte to set the desired resolution of the temperature to digital conversion. Writing TH, TL, and the configuration byte is done using a memory function command. Read access to these registers is through the scratchpad. All data is read and written least significant bit first.

DS18B20 BLOCK DIAGRAM Figure 1

**PARASITE POWER**

The block diagram (Figure 1) shows the parasite-powered circuitry. This circuitry “steals” power whenever the DQ or V_{DD} pins are high. DQ will provide sufficient power as long as the specified timing and voltage requirements are met (see the section titled “1-Wire Bus System”). The advantages of parasite power are twofold: 1) by parasiting off this pin, no local power source is needed for remote sensing of temperature, and 2) the ROM may be read in absence of normal power.

In order for the DS18B20 to be able to perform accurate temperature conversions, sufficient power must be provided over the DQ line when a temperature conversion is taking place. Since the operating current of the DS18B20 is up to 1.5 mA, the DQ line will not have sufficient drive due to the 5k pullup resistor. This problem is particularly acute if several DS18B20s are on the same DQ and attempting to convert simultaneously.

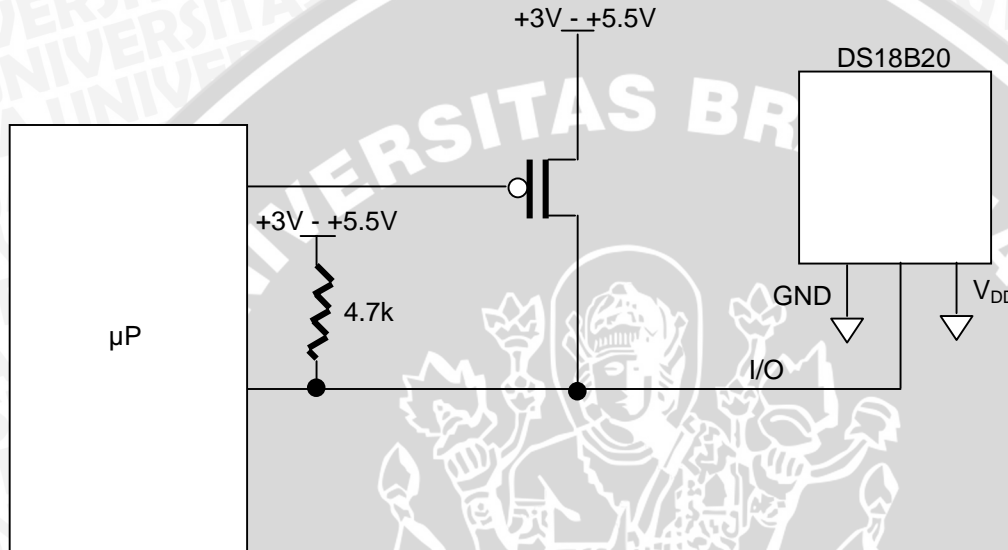
There are two ways to assure that the DS18B20 has sufficient supply current during its active conversion cycle. The first is to provide a strong pullup on the DQ line whenever temperature conversions or copies to the E^2 memory are taking place. This may be accomplished by using a MOSFET to pull the DQ line directly to the power supply as shown in Figure 2. The DQ line must be switched over to the strong pull-up within 10 μ s maximum after issuing any protocol that involves copying to the E^2 memory or initiates temperature conversions. When using the parasite power mode, the V_{DD} pin must be tied to ground.

Another method of supplying current to the DS18B20 is through the use of an external power supply tied to the V_{DD} pin, as shown in Figure 3. The advantage to this is that the strong pullup is not required on the DQ line, and the bus master need not be tied up holding that line high during temperature conversions. This allows other data traffic on the 1-Wire bus during the conversion time. In addition, any number of DS18B20s may be placed on the 1-Wire bus, and if they all use external power, they may all simultaneously perform temperature conversions by issuing the Skip ROM command and then issuing the Convert T command. Note that as long as the external power supply is active, the GND pin may not be floating.

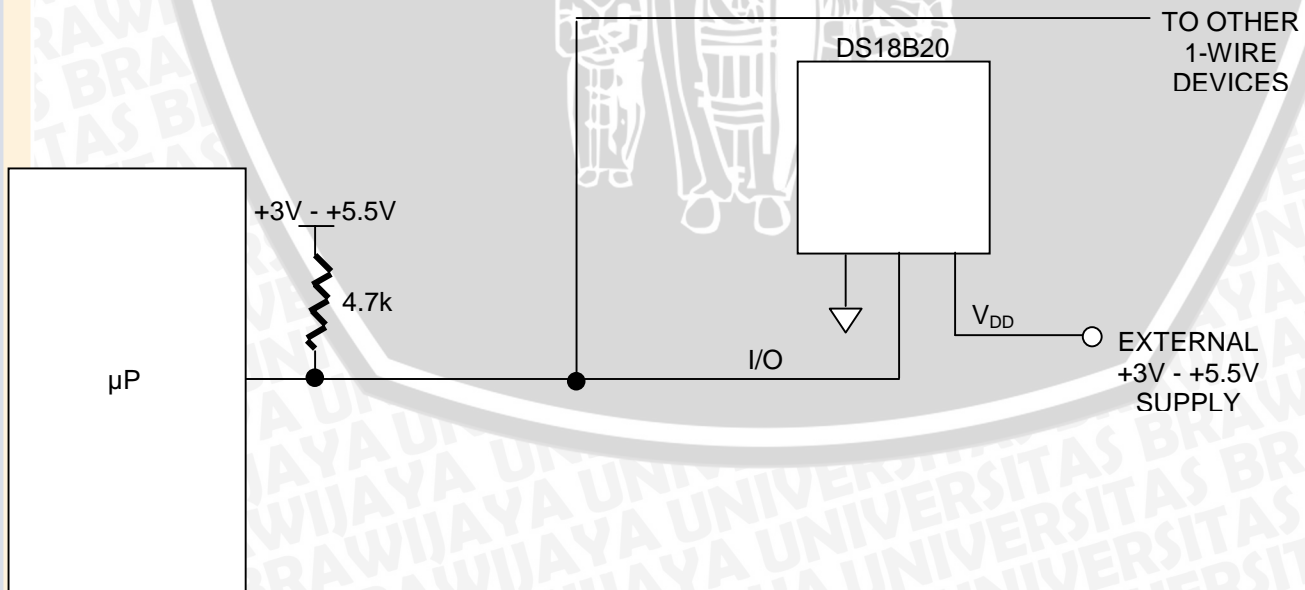
The use of parasite power is not recommended above 100°C, since it may not be able to sustain communications given the higher leakage currents the DS18B20 exhibits at these temperatures. For applications in which such temperatures are likely, it is strongly recommended that V_{DD} be applied to the DS18B20.

For situations where the bus master does not know whether the DS18B20s on the bus are parasite powered or supplied with external V_{DD} , a provision is made in the DS18B20 to signal the power supply scheme used. The bus master can determine if any DS18B20s are on the bus which require the strong pullup by sending a Skip ROM protocol, then issuing the read power supply command. After this command is issued, the master then issues read time slots. The DS18B20 will send back “0” on the 1-Wire bus if it is parasite powered; it will send back a “1” if it is powered from the V_{DD} pin. If the master receives a “0,” it knows that it must supply the strong pullup on the DQ line during temperature conversions. See “Memory Command Functions” section for more detail on this command protocol.

STRONG PULLUP FOR SUPPLYING DS18B20 DURING TEMPERATURE CONVERSION Figure 2



USING V_{DD} TO SUPPLY TEMPERATURE CONVERSION CURRENT Figure 3



OPERATION - MEASURING TEMPERATURE

The core functionality of the DS18B20 is its direct-to-digital temperature sensor. The resolution of the DS18B20 is configurable (9, 10, 11, or 12 bits), with 12-bit readings the factory default state. This equates to a temperature resolution of 0.5°C, 0.25°C, 0.125°C, or 0.0625°C. Following the issuance of the Convert T [44h] command, a temperature conversion is performed and the thermal data is stored in the scratchpad memory in a 16-bit, sign-extended two's complement format. The temperature information can be retrieved over the 1-Wire interface by issuing a Read Scratchpad [BEh] command once the conversion has been performed. The data is transferred over the 1-Wire bus, LSB first. The MSB of the temperature register contains the "sign" (S) bit, denoting whether the temperature is positive or negative.

Table 2 describes the exact relationship of output data to measured temperature. The table assumes 12-bit resolution. If the DS18B20 is configured for a lower resolution, insignificant bits will contain zeros. For Fahrenheit usage, a lookup table or conversion routine must be used.

Temperature/Data Relationships Table 2

2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	LSB
MSb				(unit = °C)				LSB
S	S	S	S	S	2^6	2^5	2^4	MSB

TEMPERATURE	DIGITAL OUTPUT (Binary)	DIGITAL OUTPUT (Hex)
+125°C	0000 0111 1101 0000	07D0h
+85°C	0000 0101 0101 0000	0550h*
+25.0625°C	0000 0001 1001 0001	0191h
+10.125°C	0000 0000 1010 0010	00A2h
+0.5°C	0000 0000 0000 1000	0008h
0°C	0000 0000 0000 0000	0000h
-0.5°C	1111 1111 1111 1000	FFF8h
-10.125°C	1111 1111 0101 1110	FF5Eh
-25.0625°C	1111 1110 0110 1111	FF6Fh
-55°C	1111 1100 1001 0000	FC90h

*The power on reset register value is +85°C.

OPERATION - ALARM SIGNALING

After the DS18B20 has performed a temperature conversion, the temperature value is compared to the trigger values stored in TH and TL. Since these registers are 8-bit only, bits 9-12 are ignored for comparison. The most significant bit of TH or TL directly corresponds to the sign bit of the 16-bit temperature register. If the result of a temperature measurement is higher than TH or lower than TL, an alarm flag inside the device is set. This flag is updated with every temperature measurement. As long as the alarm flag is set, the DS18B20 will respond to the alarm search command. This allows many DS18B20s to be connected in parallel doing simultaneous temperature measurements. If somewhere the temperature exceeds the limits, the alarming device(s) can be identified and read immediately without having to read non-alarming devices.

64-BIT LASERED ROM

Each DS18B20 contains a unique ROM code that is 64-bits long. The first 8 bits are a 1-Wire family code (DS18B20 code is 28h). The next 48 bits are a unique serial number. The last 8 bits are a CRC of the first 56 bits. (See Figure 4.) The 64-bit ROM and ROM Function Control section allow the DS18B20 to operate as a 1-Wire device and follow the 1-Wire protocol detailed in the section “1-Wire Bus System.” The functions required to control sections of the DS18B20 are not accessible until the ROM function protocol has been satisfied. This protocol is described in the ROM function protocol flowchart (Figure 5). The 1-Wire bus master must first provide one of five ROM function commands: 1) Read ROM, 2) Match ROM, 3) Search ROM, 4) Skip ROM, or 5) Alarm Search. After a ROM function sequence has been successfully executed, the functions specific to the DS18B20 are accessible and the bus master may then provide one of the six memory and control function commands.

CRC GENERATION

The DS18B20 has an 8-bit CRC stored in the most significant byte of the 64-bit ROM. The bus master can compute a CRC value from the first 56-bits of the 64-bit ROM and compare it to the value stored within the DS18B20 to determine if the ROM data has been received error-free by the bus master. The equivalent polynomial function of this CRC is:

$$\text{CRC} = X^8 + X^5 + X^4 + 1$$

The DS18B20 also generates an 8-bit CRC value using the same polynomial function shown above and provides this value to the bus master to validate the transfer of data bytes. In each case where a CRC is used for data transfer validation, the bus master must calculate a CRC value using the polynomial function given above and compare the calculated value to either the 8-bit CRC value stored in the 64-bit ROM portion of the DS18B20 (for ROM reads) or the 8-bit CRC value computed within the DS18B20 (which is read as a ninth byte when the scratchpad is read). The comparison of CRC values and decision to continue with an operation are determined entirely by the bus master. There is no circuitry inside the DS18B20 that prevents a command sequence from proceeding if the CRC stored in or calculated by the DS18B20 does not match the value generated by the bus master.

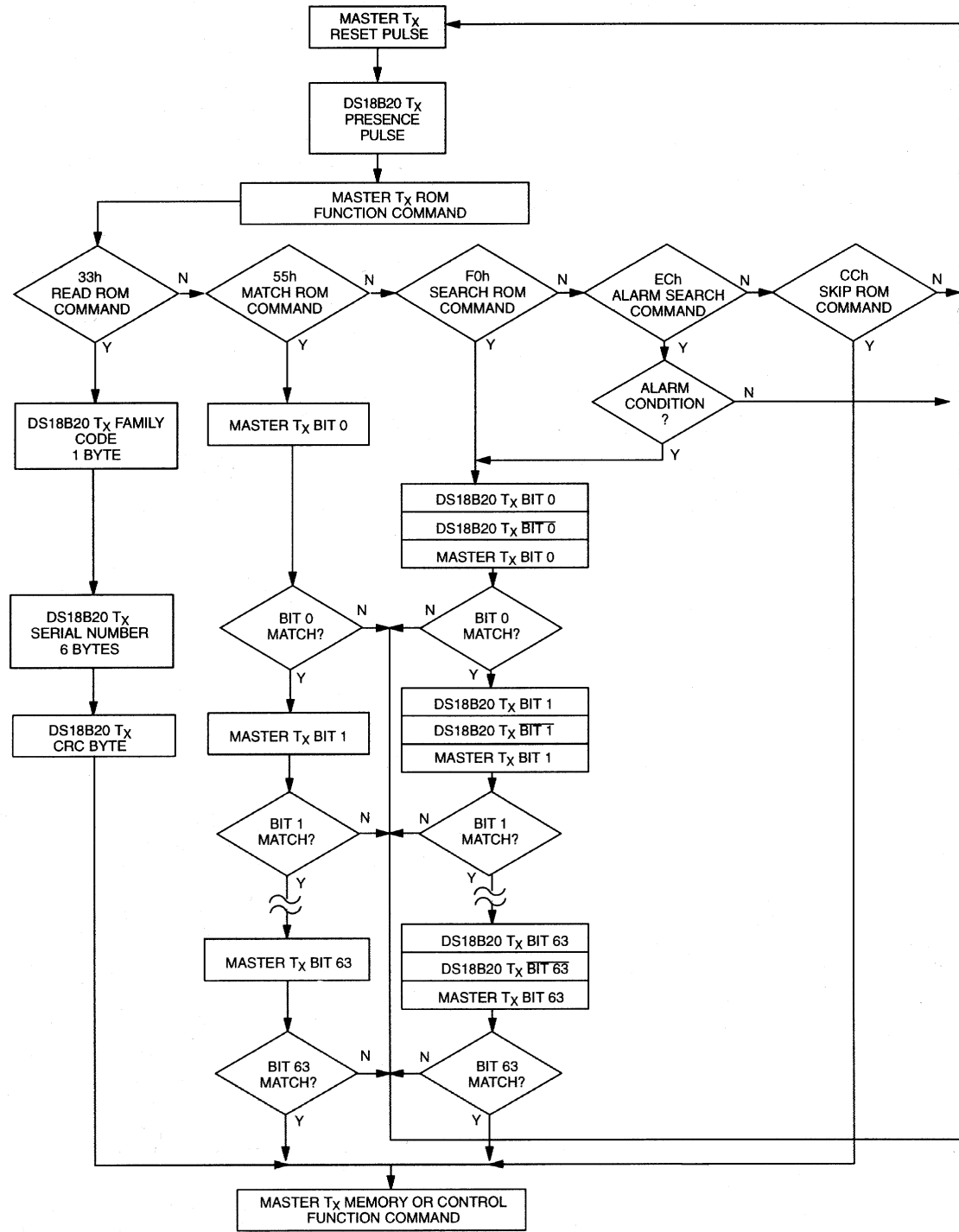
The 1-Wire CRC can be generated using a polynomial generator consisting of a shift register and XOR gates as shown in Figure 6. Additional information about the Dallas 1-Wire Cyclic Redundancy Check is available in Application Note 27 entitled “Understanding and Using Cyclic Redundancy Checks with Dallas Semiconductor Touch Memory Products.”

The shift register bits are initialized to 0. Then starting with the least significant bit of the family code, 1 bit at a time is shifted in. After the 8th bit of the family code has been entered, then the serial number is entered. After the 48th bit of the serial number has been entered, the shift register contains the CRC value. Shifting in the 8 bits of CRC should return the shift register to all 0s.

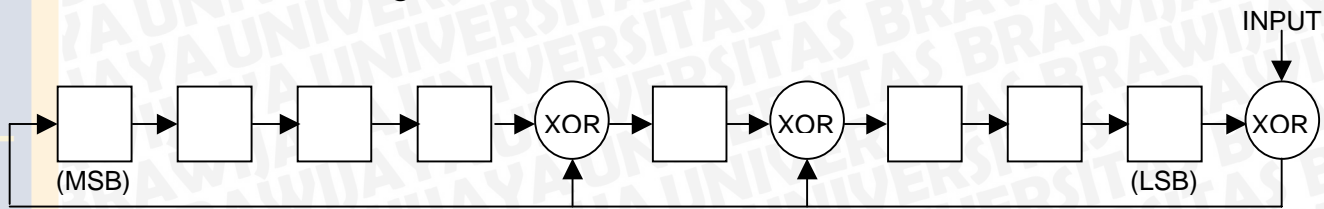
64-BIT LASERED ROM Figure 4

8-BIT CRC CODE		48-BIT SERIAL NUMBER				8-BIT FAMILY CODE (28h)	
MSB	LSB	MSB	LSB	MSB	LSB	MSB	LSB

ROM FUNCTIONS FLOW CHART Figure 5



1-WIRE CRC CODE Figure 6



MEMORY

The DS18B20's memory is organized as shown in Figure 8. The memory consists of a scratchpad RAM and a nonvolatile, electrically erasable (E²) RAM, which stores the high and low temperature triggers TH and TL, and the configuration register. The scratchpad helps insure data integrity when communicating over the 1-Wire bus. Data is first written to the scratchpad using the Write Scratchpad [4Eh] command. It can then be verified by using the Read Scratchpad [BEh] command. After the data has been verified, a Copy Scratchpad [48h] command will transfer the data to the nonvolatile (E²) RAM. This process insures data integrity when modifying memory. The DS18B20 EEPROM is rated for a minimum of 50,000 writes and 10 years data retention at T = +55°C.

The scratchpad is organized as eight bytes of memory. The first 2 bytes contain the LSB and the MSB of the measured temperature information, respectively. The third and fourth bytes are volatile copies of TH and TL and are refreshed with every power-on reset. The fifth byte is a volatile copy of the configuration register and is refreshed with every power-on reset. The configuration register will be explained in more detail later in this section of the datasheet. The sixth, seventh, and eighth bytes are used for internal computations, and thus will not read out any predictable pattern.

It is imperative that one writes TH, TL, and config in succession; i.e. a write is not valid if one writes only to TH and TL, for example, and then issues a reset. If any of these bytes must be written, all three must be written before a reset is issued.

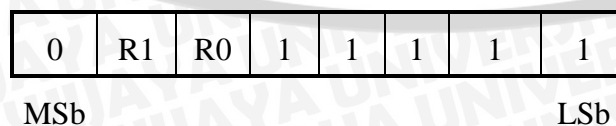
There is a ninth byte which may be read with a Read Scratchpad [BEh] command. This byte contains a cyclic redundancy check (CRC) byte which is the CRC over all of the eight previous bytes. This CRC is implemented in the fashion described in the section titled "CRC Generation".

Configuration Register

The fifth byte of the scratchpad memory is the configuration register.

It contains information which will be used by the device to determine the resolution of the temperature to digital conversion. The bits are organized as shown in Figure 7.

DS18B20 CONFIGURATION REGISTER Figure 7



Bits 0-4 are don't cares on a write but will always read out "1".

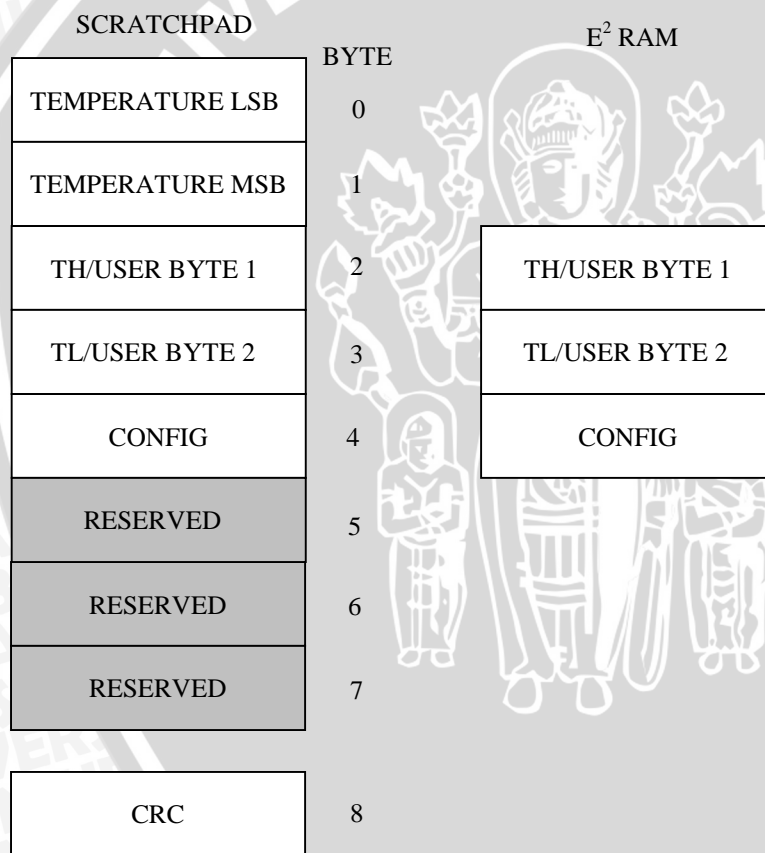
Bit 7 is a don't care on a write but will always read out "0".

R0, R1: Thermometer resolution bits. Table 3 below defines the resolution of the digital thermometer, based on the settings of these 2 bits. There is a direct tradeoff between resolution and conversion time, as depicted in the AC Electrical Characteristics. The factory default of these EEPROM bits is R0=1 and R1=1 (12-bit conversions).

Thermometer Resolution Configuration Table 3

R1	R0	Thermometer Resolution	Max Conversion Time
0	0	9 bit	93.75 ms ($t_{conv}/8$)
0	1	10 bit	187.5 ms ($t_{conv}/4$)
1	0	11 bit	375 ms ($t_{conv}/2$)
1	1	12 bit	750 ms (t_{conv})

DS18B20 MEMORY MAP Figure 8



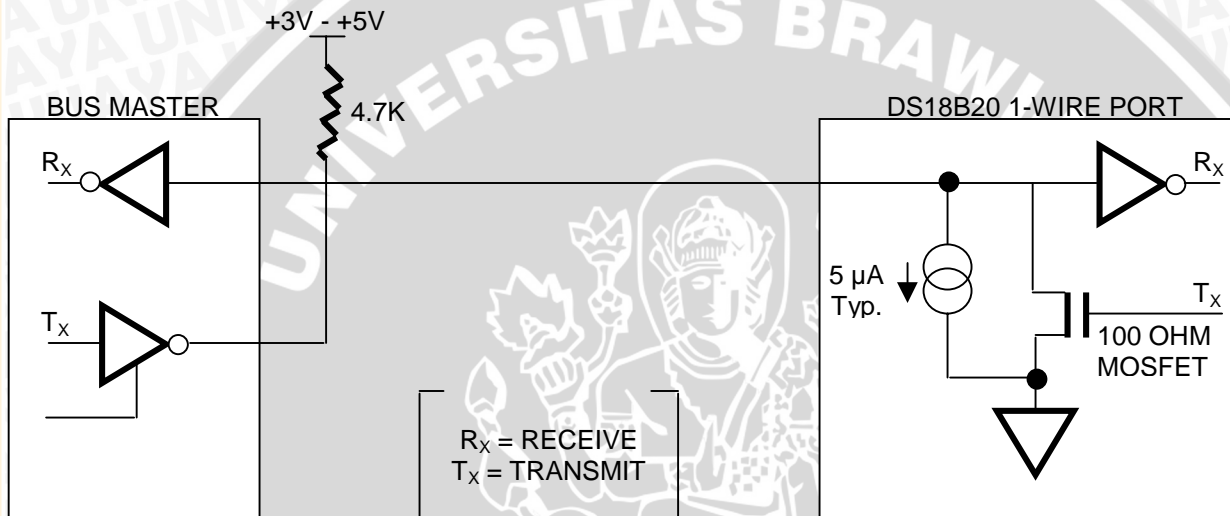
1-WIRE BUS SYSTEM

The 1-Wire bus is a system which has a single bus master and one or more slaves. The DS18B20 behaves as a slave. The discussion of this bus system is broken down into three topics: hardware configuration, transaction sequence, and 1-Wire signaling (signal types and timing).

HARDWARE CONFIGURATION

The 1-Wire bus has only a single line by definition; it is important that each device on the bus be able to drive it at the appropriate time. To facilitate this, each device attached to the 1-Wire bus must have open drain or 3-state outputs. The 1-Wire port of the DS18B20 (DQ pin) is open drain with an internal circuit equivalent to that shown in Figure 9. A multidrop bus consists of a 1-Wire bus with multiple slaves attached. The 1-Wire bus requires a pullup resistor of approximately 5 k Ω .

HARDWARE CONFIGURATION Figure 9



The idle state for the 1-Wire bus is high. If for any reason a transaction needs to be suspended, the bus MUST be left in the idle state if the transaction is to resume. Infinite recovery time can occur between bits so long as the 1-Wire bus is in the inactive (high) state during the recovery period. If this does not occur and the bus is left low for more than 480 μ s, all components on the bus will be reset.

TRANSACTION SEQUENCE

The protocol for accessing the DS18B20 via the 1-Wire port is as follows:

- Initialization
- ROM Function Command
- Memory Function Command
- Transaction/Data

INITIALIZATION

All transactions on the 1-Wire bus begin with an initialization sequence. The initialization sequence consists of a reset pulse transmitted by the bus master followed by presence pulse(s) transmitted by the slave(s).

The presence pulse lets the bus master know that the DS18B20 is on the bus and is ready to operate. For more details, see the “1-Wire Signaling” section.

ROM FUNCTION COMMANDS

Once the bus master has detected a presence, it can issue one of the five ROM function commands. All ROM function commands are 8 bits long. A list of these commands follows (refer to flowchart in Figure 5):

Read ROM [33h]

This command allows the bus master to read the DS18B20's 8-bit family code, unique 48-bit serial number, and 8-bit CRC. This command can only be used if there is a single DS18B20 on the bus. If more than one slave is present on the bus, a data collision will occur when all slaves try to transmit at the same time (open drain will produce a wired AND result).

Match ROM [55h]

The match ROM command, followed by a 64-bit ROM sequence, allows the bus master to address a specific DS18B20 on a multidrop bus. Only the DS18B20 that exactly matches the 64-bit ROM sequence will respond to the following memory function command. All slaves that do not match the 64-bit ROM sequence will wait for a reset pulse. This command can be used with a single or multiple devices on the bus.

Skip ROM [CCh]

This command can save time in a single drop bus system by allowing the bus master to access the memory functions without providing the 64-bit ROM code. If more than one slave is present on the bus and a Read command is issued following the Skip ROM command, data collision will occur on the bus as multiple slaves transmit simultaneously (open drain pulldowns will produce a wired AND result).

Search ROM [F0h]

When a system is initially brought up, the bus master might not know the number of devices on the 1-Wire bus or their 64-bit ROM codes. The search ROM command allows the bus master to use a process of elimination to identify the 64-bit ROM codes of all slave devices on the bus.

Alarm Search [ECh]

The flowchart of this command is identical to the Search ROM command. However, the DS18B20 will respond to this command only if an alarm condition has been encountered at the last temperature measurement. An alarm condition is defined as a temperature higher than TH or lower than TL. The alarm condition remains set as long as the DS18B20 is powered up, or until another temperature measurement reveals a non-alarming value. For alarming, the trigger values stored in EEPROM are taken into account. If an alarm condition exists and the TH or TL settings are changed, another temperature conversion should be done to validate any alarm conditions.

Example of a ROM Search

The ROM search process is the repetition of a simple three-step routine: read a bit, read the complement of the bit, then write the desired value of that bit. The bus master performs this simple, three-step routine on each bit of the ROM. After one complete pass, the bus master knows the contents of the ROM in one device. The remaining number of devices and their ROM codes may be identified by additional passes.

The following example of the ROM search process assumes four different devices are connected to the same 1-Wire bus. The ROM data of the four devices is as shown:

ROM1	00110101...
ROM2	10101010...
ROM3	11110101...
ROM4	00010001...

The search process is as follows:

1. The bus master begins the initialization sequence by issuing a reset pulse. The slave devices respond by issuing simultaneous presence pulses.
2. The bus master will then issue the Search ROM command on the 1-Wire bus.
3. The bus master reads a bit from the 1-Wire bus. Each device will respond by placing the value of the first bit of their respective ROM data onto the 1-Wire bus. ROM1 and ROM4 will place a 0 onto the 1-Wire bus, i.e., pull it low. ROM2 and ROM3 will place a 1 onto the 1-Wire bus by allowing the line to stay high. The result is the logical AND of all devices on the line, therefore the bus master sees a 0. The bus master reads another bit. Since the Search ROM data command is being executed, all of the devices on the 1-Wire bus respond to this second read by placing the complement of the first bit of their respective ROM data onto the 1-Wire bus. ROM1 and ROM4 will place a 1 onto the 1-Wire, allowing the line to stay high. ROM2 and ROM3 will place a 0 onto the 1-Wire, thus it will be pulled low. The bus master again observes a 0 for the complement of the first ROM data bit. The bus master has determined that there are some devices on the 1-Wire bus that have a 0 in the first position and others that have a 1.

The data obtained from the two reads of the three-step routine have the following interpretations:

00	There are still devices attached which have conflicting bits in this position.
01	All devices still coupled have a 0-bit in this bit position.
10	All devices still coupled have a 1-bit in this bit position.
11	There are no devices attached to the 1-Wire bus.

4. The bus master writes a 0. This deselects ROM2 and ROM3 for the remainder of this search pass, leaving only ROM1 and ROM4 connected to the 1-Wire bus.
5. The bus master performs two more reads and receives a 0-bit followed by a 1-bit. This indicates that all devices still coupled to the bus have 0s as their second ROM data bit.
6. The bus master then writes a 0 to keep both ROM1 and ROM4 coupled.
7. The bus master executes two reads and receives two 0-bits. This indicates that both 1-bits and 0-bits exist as the 3rd bit of the ROM data of the attached devices.

8. The bus master writes a 0-bit. This deselects ROM1, leaving ROM4 as the only device still connected.
9. The bus master reads the remainder of the ROM bits for ROM4 and continues to access the part if desired. This completes the first pass and uniquely identifies one part on the 1-Wire bus.
10. The bus master starts a new ROM search sequence by repeating steps 1 through 7.
11. The bus master writes a 1-bit. This decouples ROM4, leaving only ROM1 still coupled.
12. The bus master reads the remainder of the ROM bits for ROM1 and communicates to the underlying logic if desired. This completes the second ROM search pass, in which another of the ROMs was found.
13. The bus master starts a new ROM search by repeating steps 1 through 3.
14. The bus master writes a 1-bit. This deselects ROM1 and ROM4 for the remainder of this search pass, leaving only ROM2 and ROM3 coupled to the system.
15. The bus master executes two Read time slots and receives two 0s.
16. The bus master writes a 0-bit. This decouples ROM3 leaving only ROM2.
17. The bus master reads the remainder of the ROM bits for ROM2 and communicates to the underlying logic if desired. This completes the third ROM search pass, in which another of the ROMs was found.
18. The bus master starts a new ROM search by repeating steps 13 through 15.
19. The bus master writes a 1-bit. This decouples ROM2, leaving only ROM3.
20. The bus master reads the remainder of the ROM bits for ROM3 and communicates to the underlying logic if desired. This completes the fourth ROM search pass, in which another of the ROMs was found.

NOTE:

The bus master learns the unique ID number (ROM data pattern) of one 1-Wire device on each ROM Search operation. The time required to derive the part's unique ROM code is:

$$960 \mu\text{s} + (8 + 3 \times 64) 61 \mu\text{s} = 13.16 \text{ ms}$$

The bus master is therefore capable of identifying 75 different 1-Wire devices per second.

I/O SIGNALING

The DS18B20 requires strict protocols to insure data integrity. The protocol consists of several types of signaling on one line: reset pulse, presence pulse, write 0, write 1, read 0, and read 1. All of these signals, with the exception of the presence pulse, are initiated by the bus master.

The initialization sequence required to begin any communication with the DS18B20 is shown in Figure 11. A reset pulse followed by a presence pulse indicates the DS18B20 is ready to send or receive data given the correct ROM command and memory function command.

The bus master transmits (TX) a reset pulse (a low signal for a minimum of 480 μs). The bus master then releases the line and goes into a receive mode (RX). The 1-Wire bus is pulled to a high state via the 5k pullup resistor. After detecting the rising edge on the DQ pin, the DS18B20 waits 15-60 μs and then transmits the presence pulse (a low signal for 60-240 μs).

MEMORY COMMAND FUNCTIONS

The following command protocols are summarized in Table 4, and by the flowchart of Figure 10.

Write Scratchpad [4Eh]

This command writes to the scratchpad of the DS18B20, starting at the TH register. The next 3 bytes written will be saved in scratchpad memory at address locations 2 through 4. All 3 bytes must be written before a reset is issued.

Read Scratchpad [BEh]

This command reads the contents of the scratchpad. Reading will commence at byte 0 and will continue through the scratchpad until the ninth (byte 8, CRC) byte is read. If not all locations are to be read, the master may issue a reset to terminate reading at any time.

Copy Scratchpad [48h]

This command copies the scratchpad into the E² memory of the DS18B20, storing the temperature trigger bytes in nonvolatile memory. If the bus master issues read time slots following this command, the DS18B20 will output 0 on the bus as long as it is busy copying the scratchpad to E²; it will return a 1 when the copy process is complete. If parasite-powered, the bus master has to enable a strong pullup for at least 10 ms immediately after issuing this command. The DS18B20 EEPROM is rated for a minimum of 50,000 writes and 10 years data retention at T=+55°C.

Convert T [44h]

This command begins a temperature conversion. No further data is required. The temperature conversion will be performed and then the DS18B20 will remain idle. If the bus master issues read time slots following this command, the DS18B20 will output 0 on the bus as long as it is busy making a temperature conversion; it will return a 1 when the temperature conversion is complete. If parasite-powered, the bus master has to enable a strong pullup for a period greater than t_{conv} immediately after issuing this command.

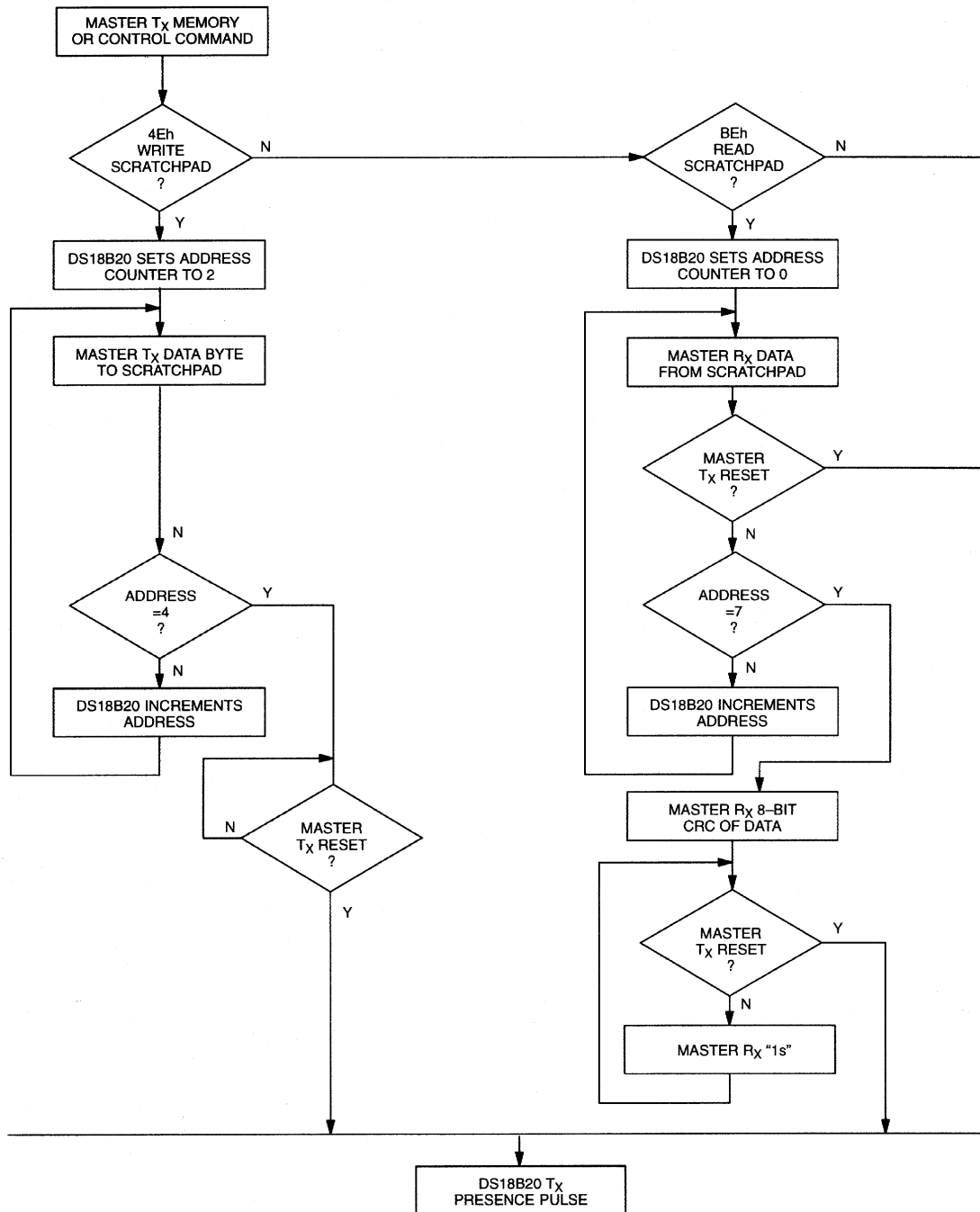
Recall E2 [B8h]

This command recalls the temperature trigger values and configuration register stored in E² to the scratchpad. This recall operation happens automatically upon power-up to the DS18B20 as well, so valid data is available in the scratchpad as soon as the device has power applied. With every read data time slot issued after this command has been sent, the device will output its temperature converter busy flag: 0=busy, 1=ready.

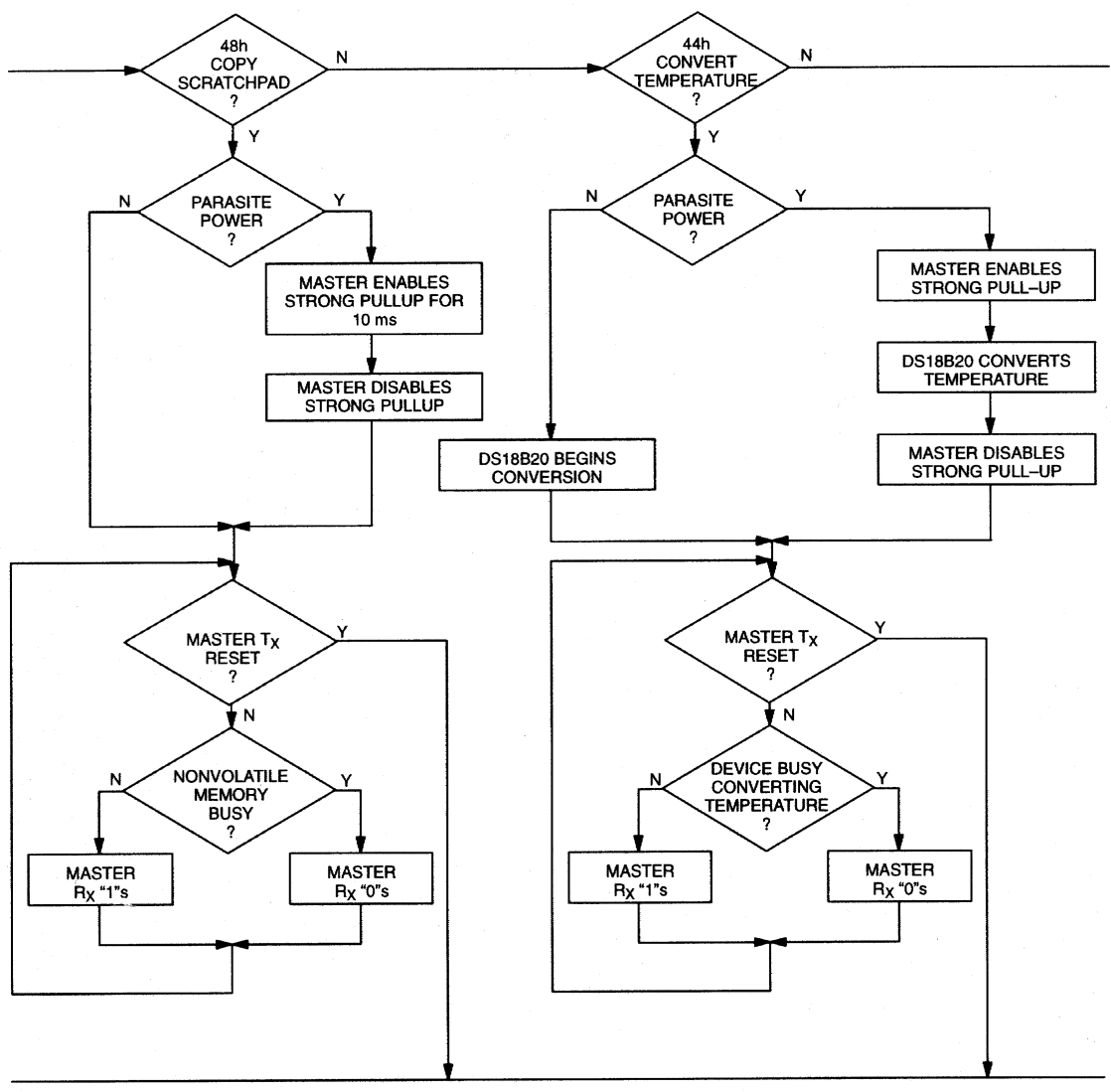
Read Power Supply [B4h]

With every read data time slot issued after this command has been sent to the DS18B20, the device will signal its power mode: 0=parasite power, 1=external power supply provided.

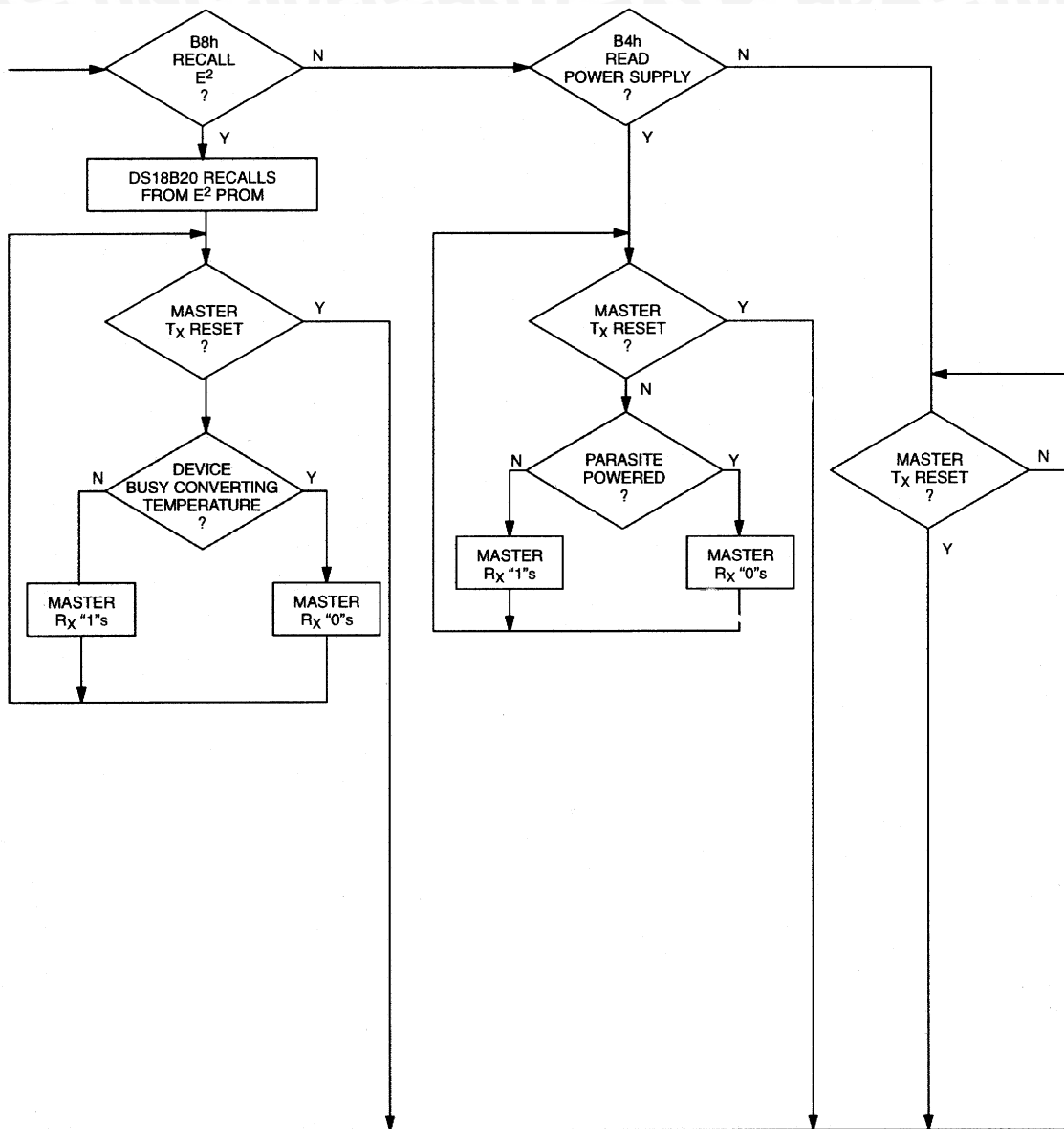
MEMORY FUNCTIONS FLOW CHART Figure 10



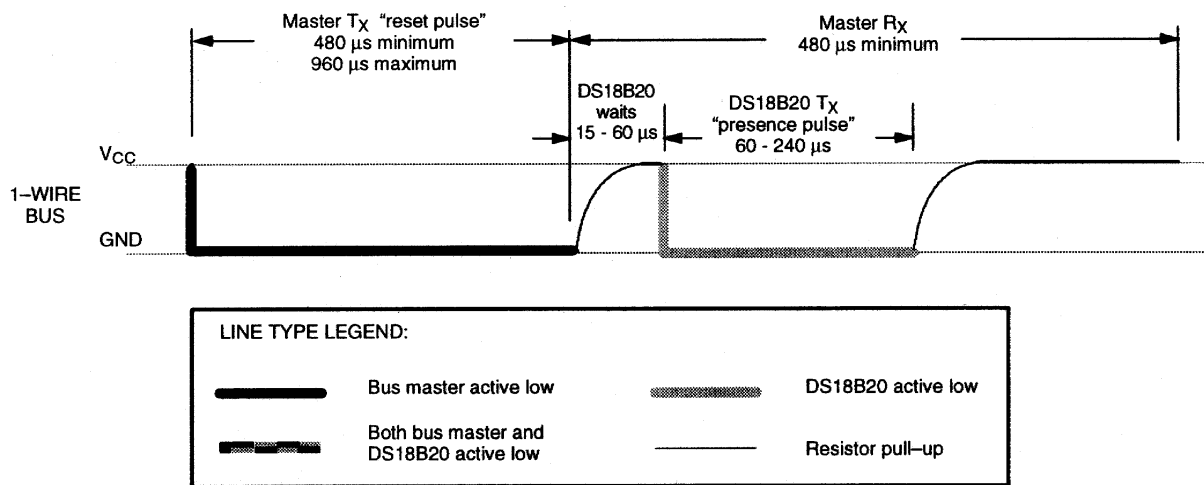
MEMORY FUNCTIONS FLOW CHART Figure 10 (cont'd)



MEMORY FUNCTIONS FLOW CHART Figure 10 (cont'd)



INITIALIZATION PROCEDURE "RESET AND PRESENCE PULSES" Figure 11



DS18B20 COMMAND SET Table 4

INSTRUCTION	DESCRIPTION	PROTOCOL	1-WIRE BUS AFTER ISSUING PROTOCOL	NOTES
TEMPERATURE CONVERSION COMMANDS				
Convert T	Initiates temperature conversion.	44h	<read temperature busy status>	1
MEMORY COMMANDS				
Read Scratchpad	Reads bytes from scratchpad and reads CRC byte.	BEh	<read data up to 9 bytes>	
Write Scratchpad	Writes bytes into scratchpad at addresses 2 through 4 (TH and TL temperature triggers and config).	4Eh	<write data into 3 bytes at addr. 2 through. 4>	3
Copy Scratchpad	Copies scratchpad into nonvolatile memory (addresses 2 through 4 only).	48h	<read copy status>	2
Recall E ²	Recalls values stored in nonvolatile memory into scratchpad (temperature triggers).	B8h	<read temperature busy status>	
Read Power Supply	Signals the mode of DS18B20 power supply to the master.	B4h	<read supply status>	

NOTES:

1. Temperature conversion takes up to 750 ms. After receiving the Convert T protocol, if the part does not receive power from the V_{DD} pin, the DQ line for the DS18B20 must be held high for at least a period greater than t_{conv} to provide power during the conversion process. As such, no other activity may take place on the 1-Wire bus for at least this period after a Convert T command has been issued.
2. After receiving the Copy Scratchpad protocol, if the part does not receive power from the V_{DD} pin, the DQ line for the DS18B20 must be held high for at least 10 ms to provide power during the copy process. As such, no other activity may take place on the 1-Wire bus for at least this period after a Copy Scratchpad command has been issued.
3. All 3 bytes must be written before a reset is issued.

READ/WRITE TIME SLOTS

DS18B20 data is read and written through the use of time slots to manipulate bits and a command word to specify the transaction.

Write Time Slots

A write time slot is initiated when the host pulls the data line from a high logic level to a low logic level. There are two types of write time slots: Write 1 time slots and Write 0 time slots. All write time slots must be a minimum of 60 μs in duration with a minimum of a 1- μs recovery time between individual write cycles.

The DS18B20 samples the DQ line in a window of 15 μs to 60 μs after the DQ line falls. If the line is high, a Write 1 occurs. If the line is low, a Write 0 occurs (see Figure 12).

For the host to generate a Write 1 time slot, the data line must be pulled to a logic low level and then released, allowing the data line to pull up to a high level within 15 μs after the start of the write time slot.

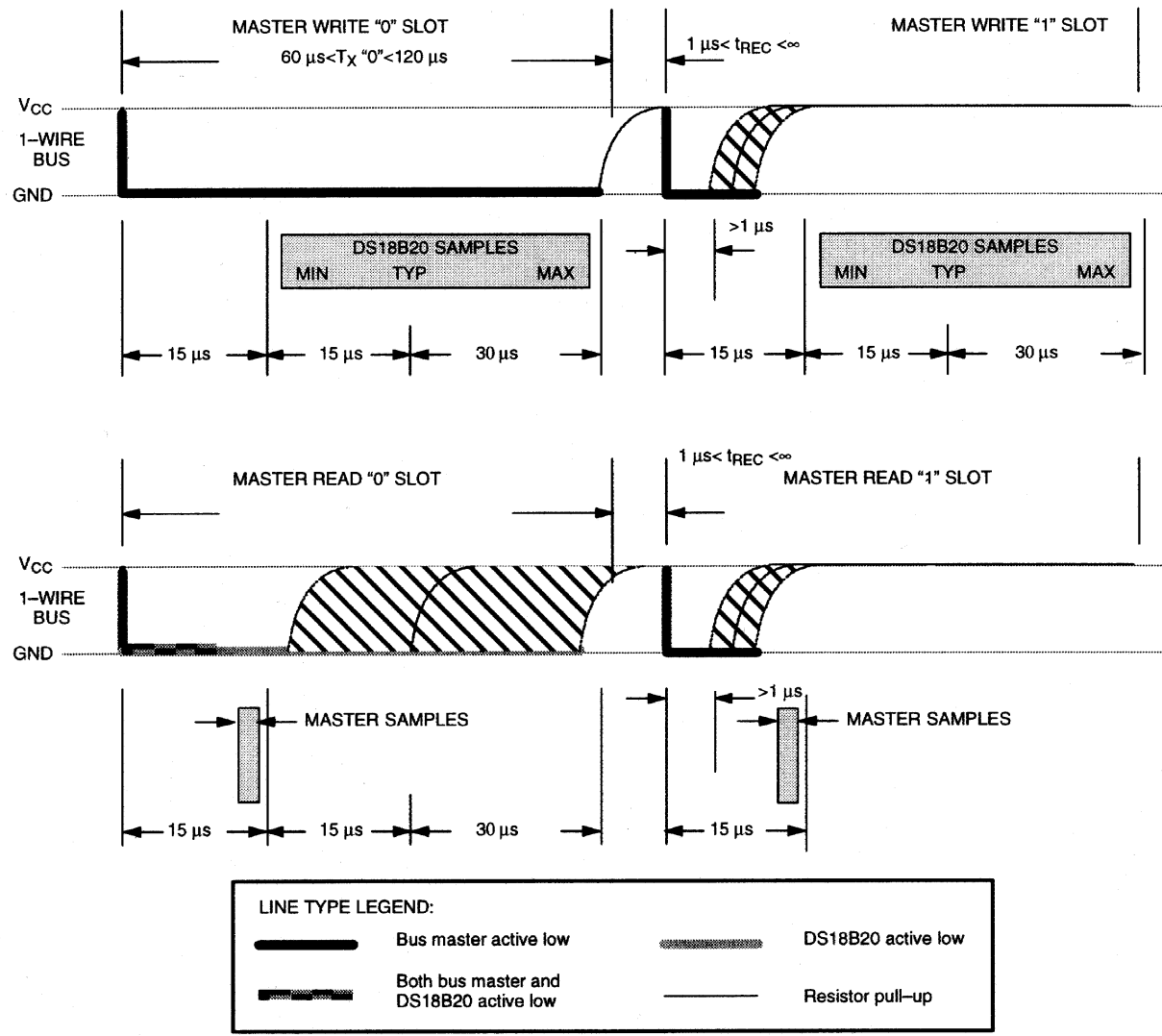
For the host to generate a Write 0 time slot, the data line must be pulled to a logic low level and remain low for 60 μs .

Read Time Slots

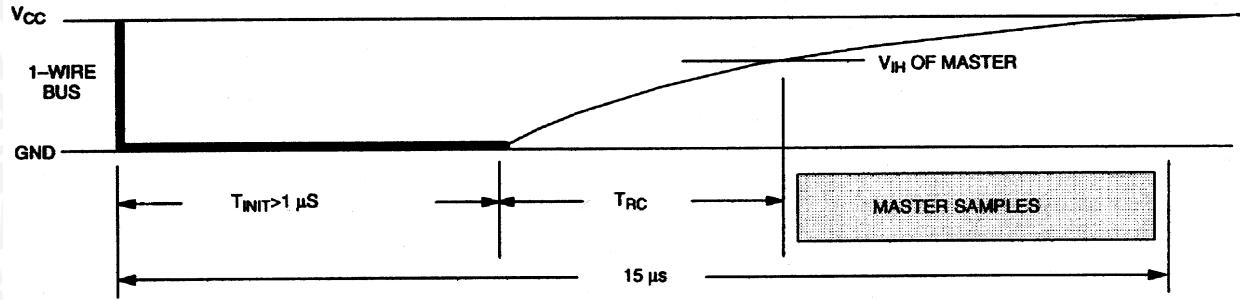
The host generates read time slots when data is to be read from the DS18B20. A read time slot is initiated when the host pulls the data line from a logic high level to logic low level. The data line must remain at a low logic level for a minimum of 1 μs ; output data from the DS18B20 is valid for 15 μs after the falling edge of the read time slot. The host therefore must stop driving the DQ pin low in order to read its state 15 μs from the start of the read slot (see Figure 12). By the end of the read time slot, the DQ pin will pull back high via the external pullup resistor. All read time slots must be a minimum of 60 μs in duration with a minimum of a 1- μs recovery time between individual read slots.

Figure 12 shows that the sum of T_{INIT} , T_{RC} , and T_{SAMPLE} must be less than 15 μs . Figure 14 shows that system timing margin is maximized by keeping T_{INIT} and T_{RC} as small as possible and by locating the master sample time towards the end of the 15- μs period.

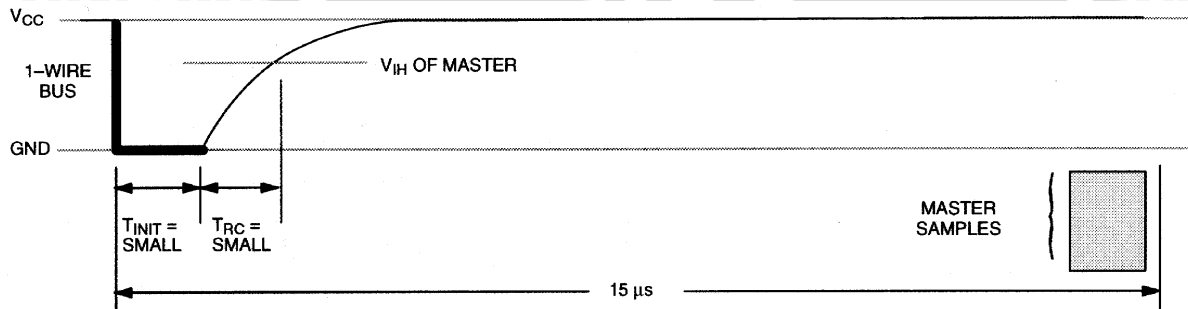
READ/WRITE TIMING DIAGRAM Figure 12



DETAILED MASTER READ 1 TIMING Figure 13



RECOMMENDED MASTER READ 1 TIMING Figure 14



LINE TYPE LEGEND:			
	Bus master active low		DS18B20 active low
	Both bus master and DS18B20 active low		Resistor pull-up



Related Application Notes

The following Application Notes can be applied to the DS18B20. These notes can be obtained from the Dallas Semiconductor “Application Note Book,” via our website at <http://www.dalsemi.com/>.

Application Note 27: “Understanding and Using Cyclic Redundancy Checks with Dallas Semiconductor Touch Memory Product”

Application Note 55: “Extending the Contact Range of Touch Memories”

Application Note 74: “Reading and Writing Touch Memories via Serial Interfaces”

Application Note 104: “Minimalist Temperature Control Demo”

Application Note 106: “Complex MicroLANs”

Application Note 108: “MicroLAN - In the Long Run”

Sample 1-Wire subroutines that can be used in conjunction with AN74 can be downloaded from the website or our Anonymous FTP Site.

MEMORY FUNCTION EXAMPLE Table 5

Example: Bus Master initiates temperature conversion, then reads temperature (parasite power assumed).

MASTER MODE	DATA (LSB FIRST)	COMMENTS
TX	Reset	Reset pulse (480-960 μ s).
RX	Presence	Presence pulse.
TX	55h	Issue “Match ROM” command.
TX	<64-bit ROM code>	Issue address for DS18B20.
TX	44h	Issue “Convert T” command.
TX	<I/O LINE HIGH>	I/O line is held high for at least a period of time greater than t_{conv} by bus master to allow conversion to complete.
TX	Reset	Reset pulse.
RX	Presence	Presence pulse.
TX	55h	Issue “Match ROM” command.
TX	<64-bit ROM code>	Issue address for DS18B20.
TX	BEh	Issue “Read Scratchpad” command.
RX	<9 data bytes>	Read entire scratchpad plus CRC; the master now recalculates the CRC of the eight data bytes received from the scratchpad, compares the CRC calculated and the CRC read. If they match, the master continues; if not, this read operation is repeated.
TX	Reset	Reset pulse.
RX	Presence	Presence pulse, done.

MEMORY FUNCTION EXAMPLE Table 6

Example: Bus Master writes memory (parasite power and only one DS18B20 assumed).

MASTER MODE	DATA (LSB FIRST)	COMMENTS
TX	Reset	Reset pulse.
RX	Presence	Presence pulse.
TX	CCh	Skip ROM command.
TX	4Eh	Write Scratchpad command.
TX	<3 data bytes>	Writes three bytes to scratchpad (TH, TL, and config).
TX	Reset	Reset pulse.
RX	Presence	Presence pulse.
TX	CCh	Skip ROM command.
TX	BEh	Read Scratchpad command.
RX	<9 data bytes>	Read entire scratchpad plus CRC. The master now recalculates the CRC of the eight data bytes received from the scratchpad, compares the CRC and the two other bytes read back from the scratchpad. If data match, the master continues; if not, repeat the sequence.
TX	Reset	Reset pulse.
RX	Presence	Presence pulse.
TX	CCh	Skip ROM command.
TX	48h	Copy Scratchpad command; after issuing this command, the master must wait 10 ms for copy operation to complete.
TX	Reset	Reset pulse.
RX	Presence	Presence pulse, done.

ABSOLUTE MAXIMUM RATINGS*

Voltage on Any Pin Relative to Ground	-0.5V to +6.0V
Operating Temperature	-55°C to +125°C
Storage Temperature	-55°C to +125°C
Soldering Temperature	See J-STD-020A specification

* This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operation sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods of time may affect reliability.

RECOMMENDED DC OPERATING CONDITIONS

PARAMETER	SYMBOL	CONDITION	MIN	TYP	MAX	UNITS	NOTES
Supply Voltage	V _{DD}	Local Power	3.0		5.5	V	1
Data Pin	DQ		-0.3		+5.5	V	1
Logic 1	V _{IH}		2.2		V _{CC} +0.3	V	1,2
Logic 0	V _{IL}		-0.3		+0.8	V	1,3,7

DC ELECTRICAL CHARACTERISTICS (-55°C to +125°C; V_{DD}=3.0V to 5.5V)

PARAMETER	SYMBOL	CONDITION	MIN	TYP	MAX	UNITS	NOTES
Thermometer Error	t _{ERR}	-10°C to +85°C			±½	°C	
		-55°C to +125°C			±2		
Input Logic High	V _{IH}	Local Power	2.2		5.5	V	1,2
		Parasite Power	3.0			V	1,2
Input Logic Low	V _{IL}		-0.3		+0.8	V	1,3,7
Sink Current	I _L	V _{IO} =0.4V	-4.0			mA	1
Standby Current	I _{DDS}			750	1000	nA	6,8
Active Current	I _{DD}			1	1.5	mA	4
DQ-Input Load Current	I _{DQ}			5		µA	5

AC ELECTRICAL CHARACTERISTICS: NV MEMORY(-55°C to +125°C; V_{DD}=3.0V to 5.5V)

PARAMETER	SYMBOL	CONDITION	MIN	TYP	MAX	UNITS	NOTES
NV Write Cycle Time	t _{wr}			2	10	ms	
EEPROM Writes	N _{EEWR}	-55°C to +55°C	50k			writes	
EEPROM Data Retention	t _{EEDR}	-55°C to +55°C	10			years	

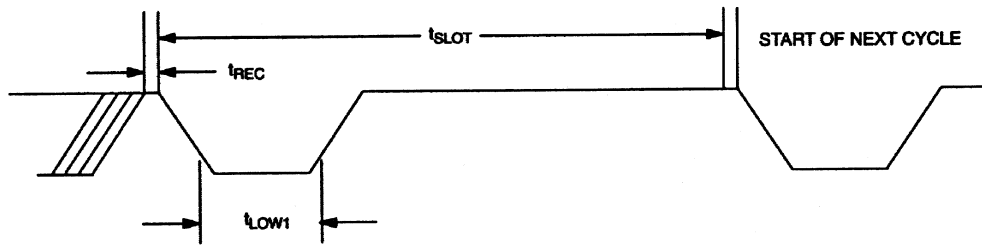
AC ELECTRICAL CHARACTERISTICS: (-55°C to +125°C; $V_{DD}=3.0V$ to 5.5V)

PARAMETER	SYMBOL	CONDITION	MIN	TYP	MAX	UNITS	NOTES
Temperature Conversion Time	t_{CONV}	9 bit			93.75	ms	
		10 bit			187.5		
		11 bit			375		
		12 bit			750		
Time Slot	t_{SLOT}		60		120	μs	
Recovery Time	t_{REC}		1			μs	
Write 0 Low Time	t_{LOW0}		60		120	μs	
Write 1 Low Time	t_{LOW1}		1		15	μs	
Read Data Valid	t_{RDV}				15	μs	
Reset Time High	t_{RSTH}		480			μs	
Reset Time Low	t_{RSTL}		480			μs	9
Presence Detect High	t_{PDHIGH}		15		60	μs	
Presence Detect Low	t_{PDLOW}		60		240	μs	
Capacitance	$C_{IN/OUT}$				25	pF	

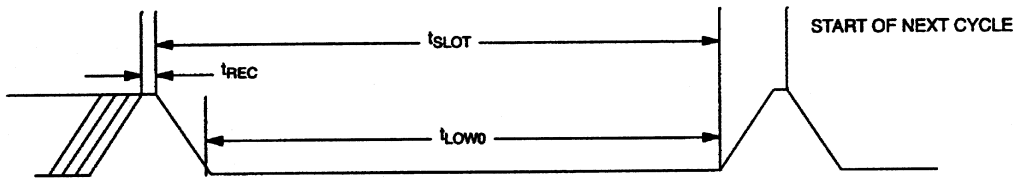
NOTES:

- All voltages are referenced to ground.
- Logic one voltages are specified at a source current of 1 mA.
- Logic zero voltages are specified at a sink current of 4 mA.
- Active current refers to either temperature conversion or writing to the E² memory. Writing to E² memory consumes approximately 200 μA for up to 10 ms.
- Input load is to ground.
- Standby current specified up to 70°C. Standby current typically is 3 μA at 125°C.
- To always guarantee a presence pulse under low voltage parasite power conditions, V_{ILMAX} may have to be reduced to as much as 0.5V.
- To minimize I_{DDs} , DQ should be: $GND \leq DQ \leq GND + 0.3V$ or $V_{DD} - 0.3V \leq DQ \leq V_{DD}$.
- Under parasite power, the max t_{RSTL} before a power on reset occurs is 960 μs .

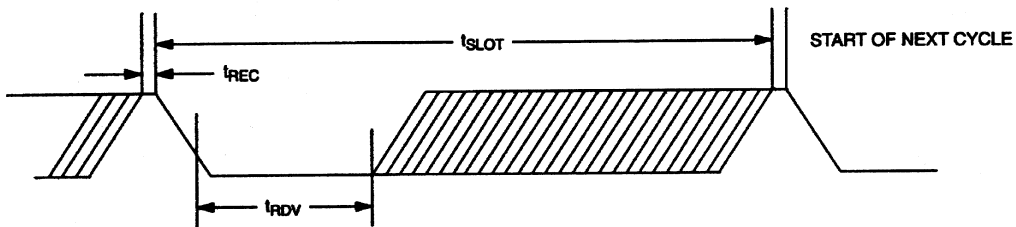
1-WIRE WRITE ONE TIME SLOT



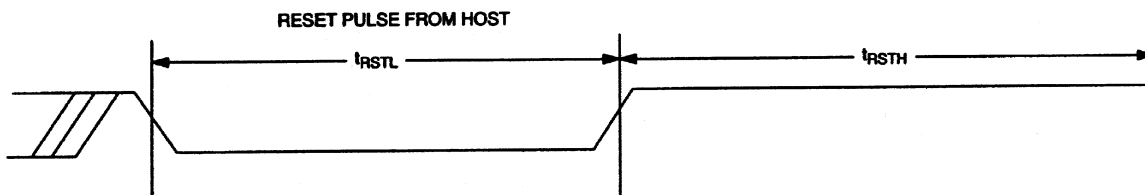
1-WIRE WRITE ZERO TIME SLOT



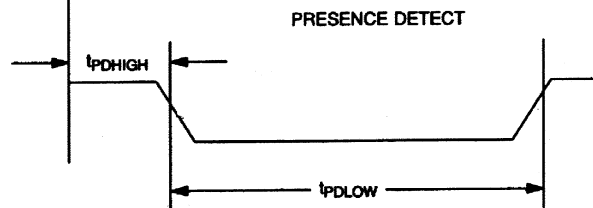
1-WIRE READ ZERO TIME SLOT



1-WIRE RESET PULSE

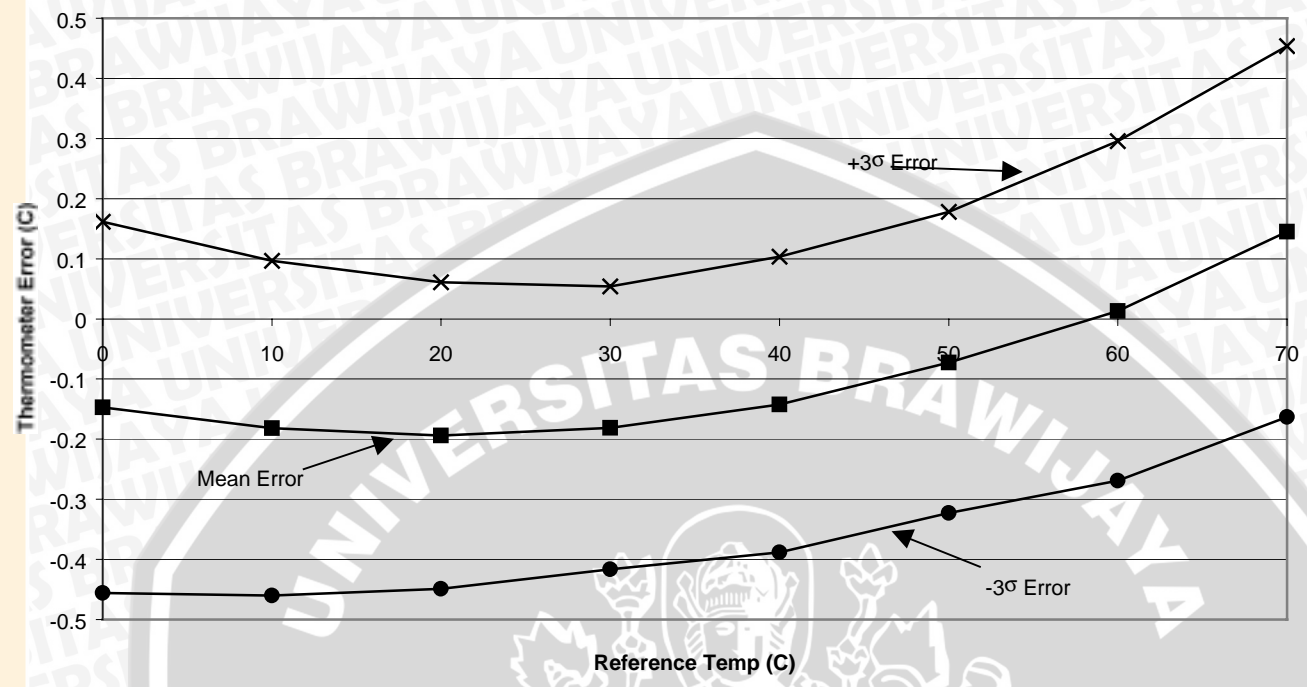


1-WIRE PRESENCE DETECT



TYPICAL PERFORMANCE CURVE

DS18B20 Typical Error Curve





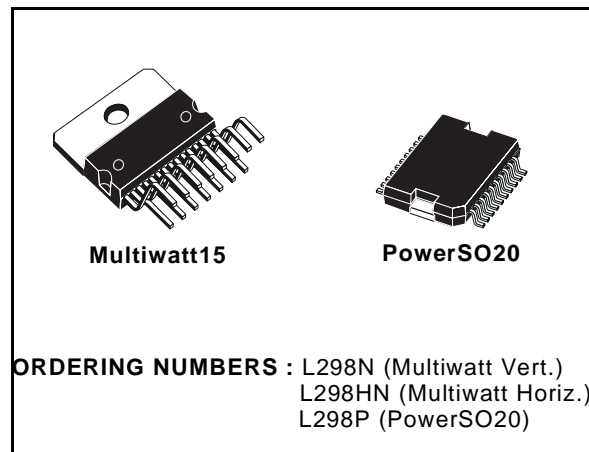
L298

DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)

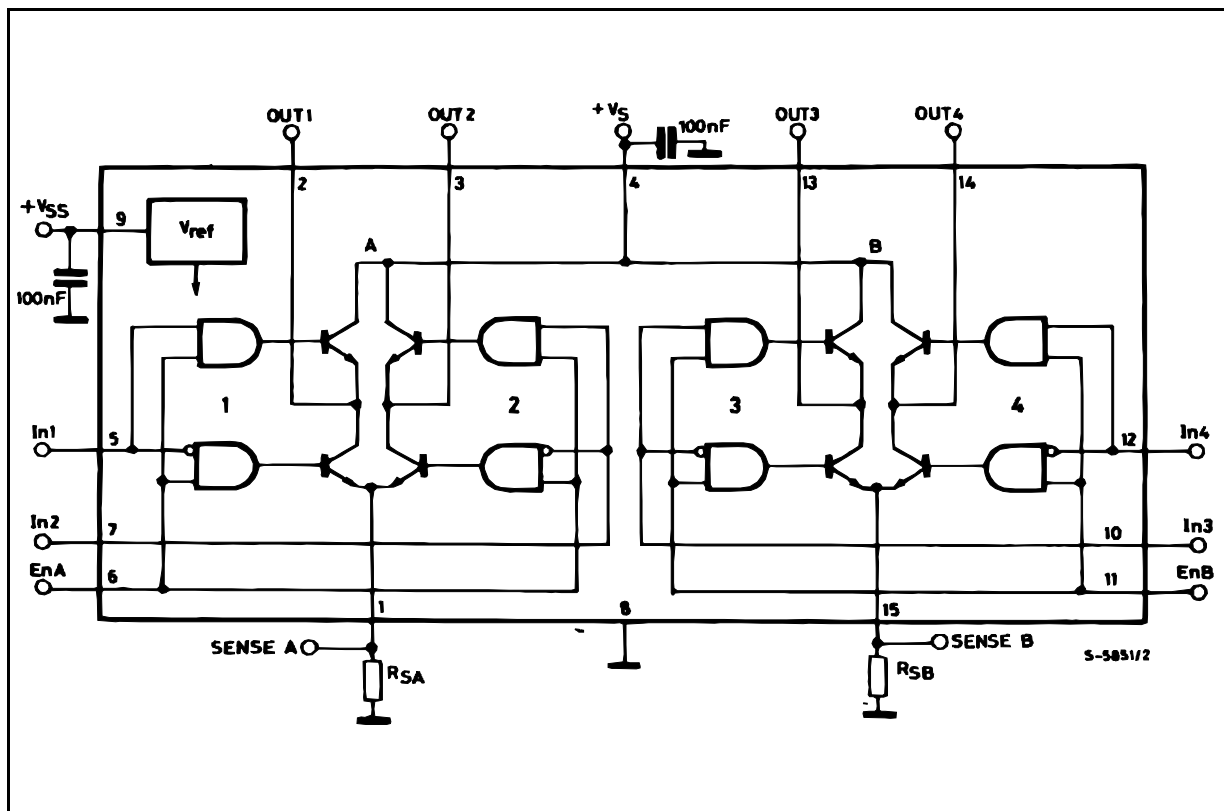
DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the con-



nection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

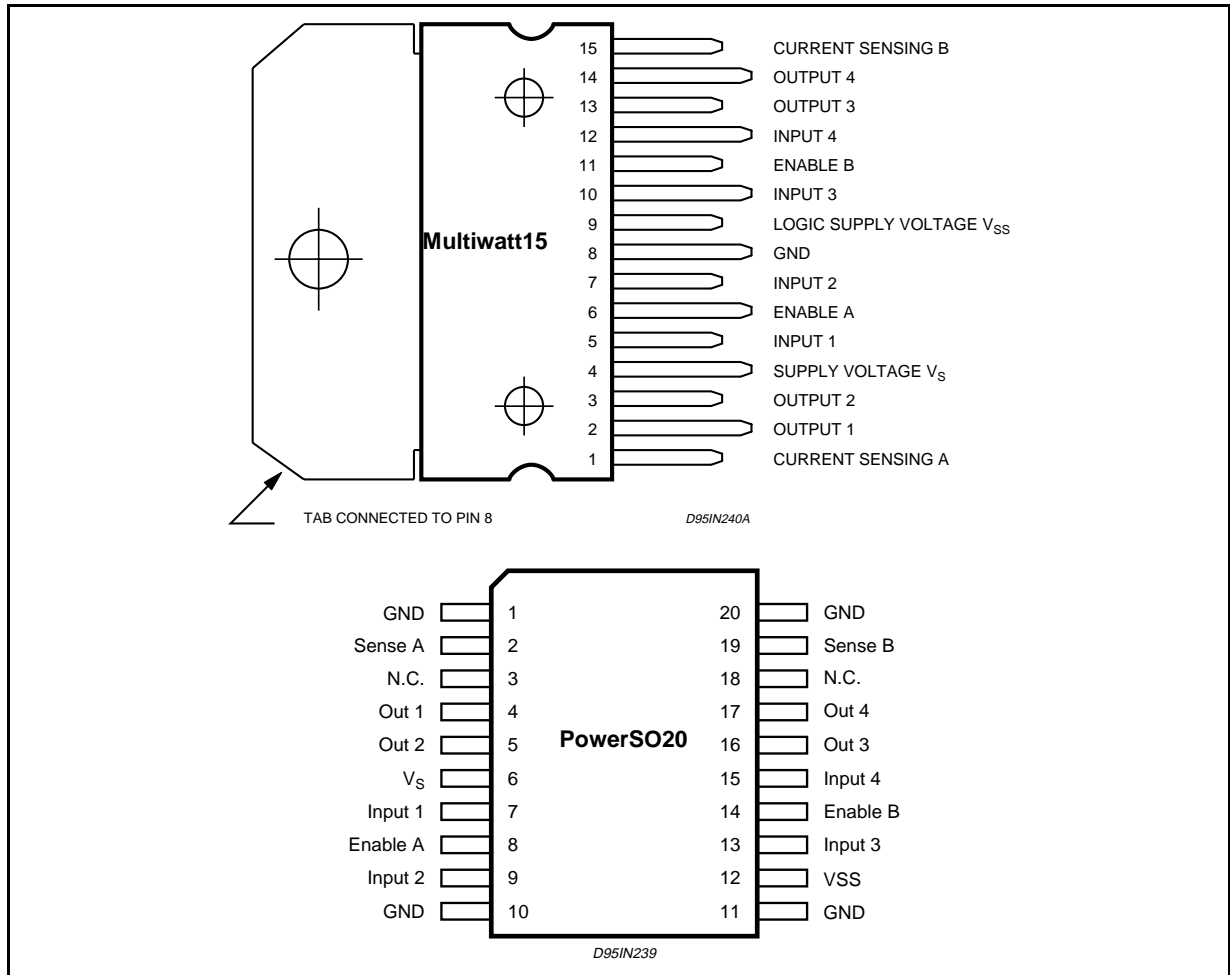
BLOCK DIAGRAM



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_S	Power Supply	50	V
V_{SS}	Logic Supply Voltage	7	V
V_I, V_{en}	Input and Enable Voltage	-0.3 to 7	V
I_O	Peak Output Current (each Channel)		
	- Non Repetitive ($t = 100\mu s$)	3	A
	- Repetitive (80% on -20% off; $t_{on} = 10ms$)	2.5	A
	-DC Operation	2	A
V_{sens}	Sensing Voltage	-1 to 2.3	V
P_{tot}	Total Power Dissipation ($T_{case} = 75^\circ C$)	25	W
T_{op}	Junction Operating Temperature	-25 to 130	$^\circ C$
T_{stg}, T_j	Storage and Junction Temperature	-40 to 150	$^\circ C$

PIN CONNECTIONS (top view)



THERMAL DATA

Symbol	Parameter		PowerSO20	Multiwatt15	Unit
$R_{th\ j-case}$	Thermal Resistance Junction-case	Max.	-	3	$^\circ C/W$
$R_{th\ j-amb}$	Thermal Resistance Junction-ambient	Max.	13 (*)	35	$^\circ C/W$

(*) Mounted on aluminum substrate



PIN FUNCTIONS (refer to the block diagram)

MW.15	PowerSO	Name	Function
1;15	2;19	Sense A; Sense B	Between this pin and ground is connected the sense resistor to control the current of the load.
2;3	4;5	Out 1; Out 2	Outputs of the Bridge A; the current that flows through the load connected between these two pins is monitored at pin 1.
4	6	V _s	Supply Voltage for the Power Output Stages. A non-inductive 100nF capacitor must be connected between this pin and ground.
5;7	7;9	Input 1; Input 2	TTL Compatible Inputs of the Bridge A.
6;11	8;14	Enable A; Enable B	TTL Compatible Enable Input: the L state disables the bridge A (enable A) and/or the bridge B (enable B).
8	1,10,11,20	GND	Ground.
9	12	V _{SS}	Supply Voltage for the Logic Blocks. A100nF capacitor must be connected between this pin and ground.
10; 12	13;15	Input 3; Input 4	TTL Compatible Inputs of the Bridge B.
13; 14	16;17	Out 3; Out 4	Outputs of the Bridge B. The current that flows through the load connected between these two pins is monitored at pin 15.
–	3;18	N.C.	Not Connected

ELECTRICAL CHARACTERISTICS (V_S = 42V; V_{SS} = 5V, T_j = 25°C; unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V _S	Supply Voltage (pin 4)	Operative Condition	V _{IH} +2.5		46	V
V _{SS}	Logic Supply Voltage (pin 9)		4.5	5	7	V
I _S	Quiescent Supply Current (pin 4)	V _{en} = H; I _L = 0	V _i = L V _i = H	13 50	22 70	mA mA
		V _{en} = L	V _i = X		4	mA
I _{SS}	Quiescent Current from V _{SS} (pin 9)	V _{en} = H; I _L = 0	V _i = L V _i = H	24 7	36 12	mA mA
		V _{en} = L	V _i = X		6	mA
V _{iL}	Input Low Voltage (pins 5, 7, 10, 12)		–0.3		1.5	V
V _{iH}	Input High Voltage (pins 5, 7, 10, 12)		2.3		V _{SS}	V
I _{iL}	Low Voltage Input Current (pins 5, 7, 10, 12)	V _i = L			–10	μA
I _{iH}	High Voltage Input Current (pins 5, 7, 10, 12)	V _i = H ≤ V _{SS} – 0.6V		30	100	μA
V _{en} = L	Enable Low Voltage (pins 6, 11)		–0.3		1.5	V
V _{en} = H	Enable High Voltage (pins 6, 11)		2.3		V _{SS}	V
I _{en} = L	Low Voltage Enable Current (pins 6, 11)	V _{en} = L			–10	μA
I _{en} = H	High Voltage Enable Current (pins 6, 11)	V _{en} = H ≤ V _{SS} – 0.6V		30	100	μA
V _{CEsat} (H)	Source Saturation Voltage	I _L = 1A I _L = 2A	0.95	1.35 2	1.7 2.7	V V
V _{CEsat} (L)	Sink Saturation Voltage	I _L = 1A (5) I _L = 2A (5)	0.85	1.2 1.7	1.6 2.3	V V
V _{CEsat}	Total Drop	I _L = 1A (5) I _L = 2A (5)	1.80		3.2 4.9	V V
V _{sens}	Sensing Voltage (pins 1, 15)		–1 (1)		2	V

ELECTRICAL CHARACTERISTICS (continued)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
T ₁ (V _i)	Source Current Turn-off Delay	0.5 V _i to 0.9 I _L (2); (4)		1.5		μs
T ₂ (V _i)	Source Current Fall Time	0.9 I _L to 0.1 I _L (2); (4)		0.2		μs
T ₃ (V _i)	Source Current Turn-on Delay	0.5 V _i to 0.1 I _L (2); (4)		2		μs
T ₄ (V _i)	Source Current Rise Time	0.1 I _L to 0.9 I _L (2); (4)		0.7		μs
T ₅ (V _i)	Sink Current Turn-off Delay	0.5 V _i to 0.9 I _L (3); (4)		0.7		μs
T ₆ (V _i)	Sink Current Fall Time	0.9 I _L to 0.1 I _L (3); (4)		0.25		μs
T ₇ (V _i)	Sink Current Turn-on Delay	0.5 V _i to 0.9 I _L (3); (4)		1.6		μs
T ₈ (V _i)	Sink Current Rise Time	0.1 I _L to 0.9 I _L (3); (4)		0.2		μs
f _c (V _i)	Commutation Frequency	I _L = 2A		25	40	KHz
T ₁ (V _{en})	Source Current Turn-off Delay	0.5 V _{en} to 0.9 I _L (2); (4)		3		μs
T ₂ (V _{en})	Source Current Fall Time	0.9 I _L to 0.1 I _L (2); (4)		1		μs
T ₃ (V _{en})	Source Current Turn-on Delay	0.5 V _{en} to 0.1 I _L (2); (4)		0.3		μs
T ₄ (V _{en})	Source Current Rise Time	0.1 I _L to 0.9 I _L (2); (4)		0.4		μs
T ₅ (V _{en})	Sink Current Turn-off Delay	0.5 V _{en} to 0.9 I _L (3); (4)		2.2		μs
T ₆ (V _{en})	Sink Current Fall Time	0.9 I _L to 0.1 I _L (3); (4)		0.35		μs
T ₇ (V _{en})	Sink Current Turn-on Delay	0.5 V _{en} to 0.9 I _L (3); (4)		0.25		μs
T ₈ (V _{en})	Sink Current Rise Time	0.1 I _L to 0.9 I _L (3); (4)		0.1		μs

- 1) Sensing voltage can be -1 V for t ≤ 50 μsec; in steady state V_{sens} min ≥ -0.5 V.
- 2) See fig. 2.
- 3) See fig. 4.
- 4) The load must be a pure resistor.

Figure 1 : Typical Saturation Voltage vs. Output Current.

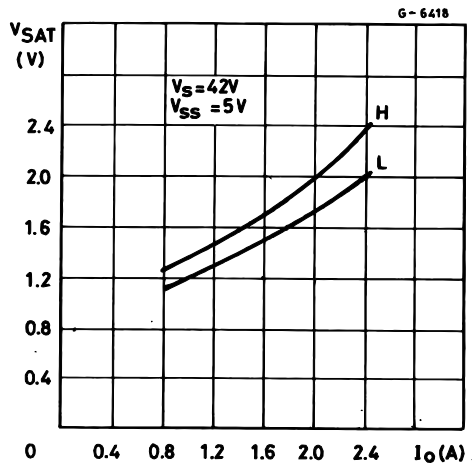
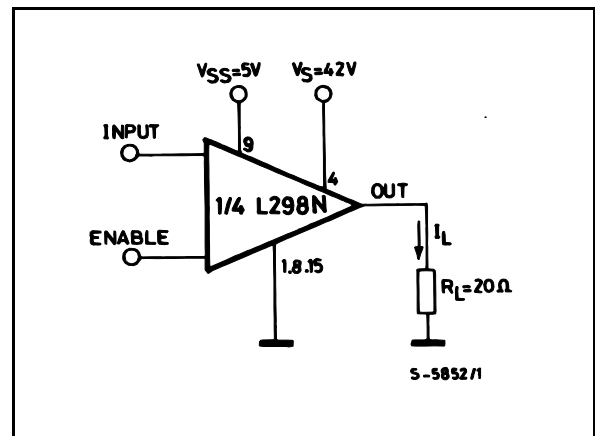


Figure 2 : Switching Times Test Circuits.



Note : For INPUT Switching, set EN = H
 For ENABLE Switching, set IN = H

Figure 3 : Source Current Delay Times vs. Input or Enable Switching.

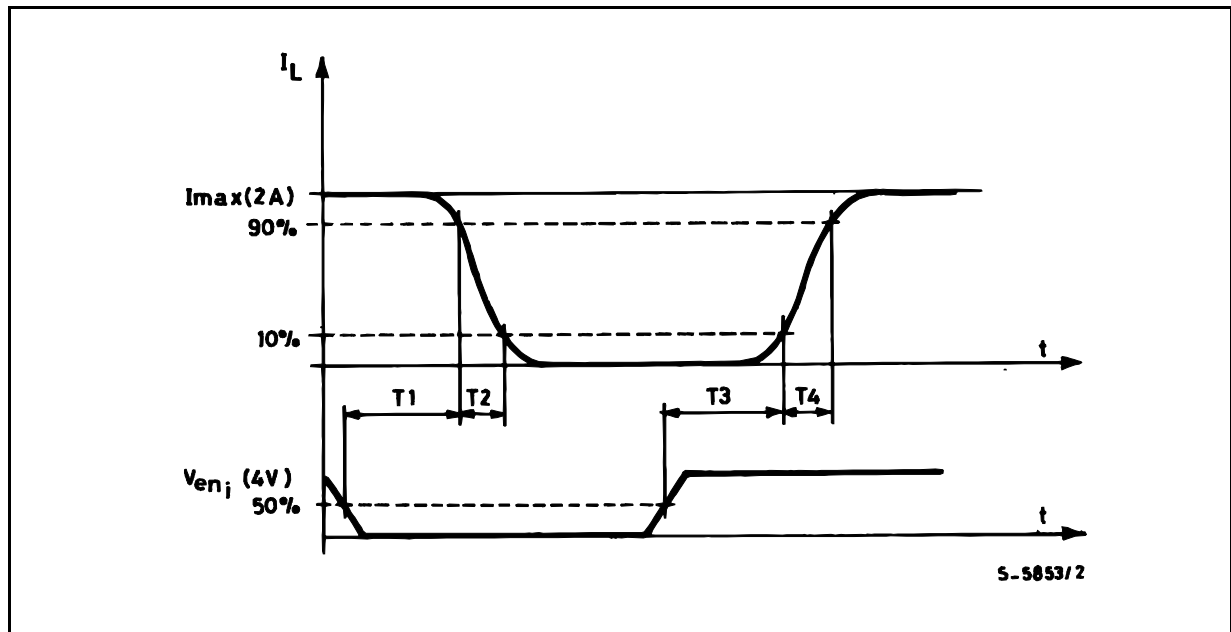
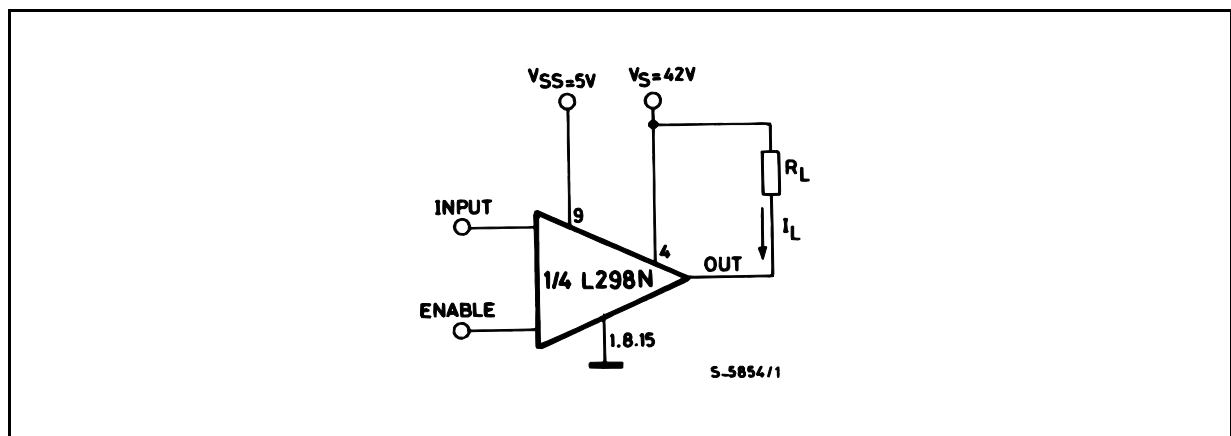


Figure 4 : Switching Times Test Circuits.



Note : For INPUT Switching, set EN = H
For ENABLE Switching, set IN = L

Figure 5 : Sink Current Delay Times vs. Input 0 V Enable Switching.

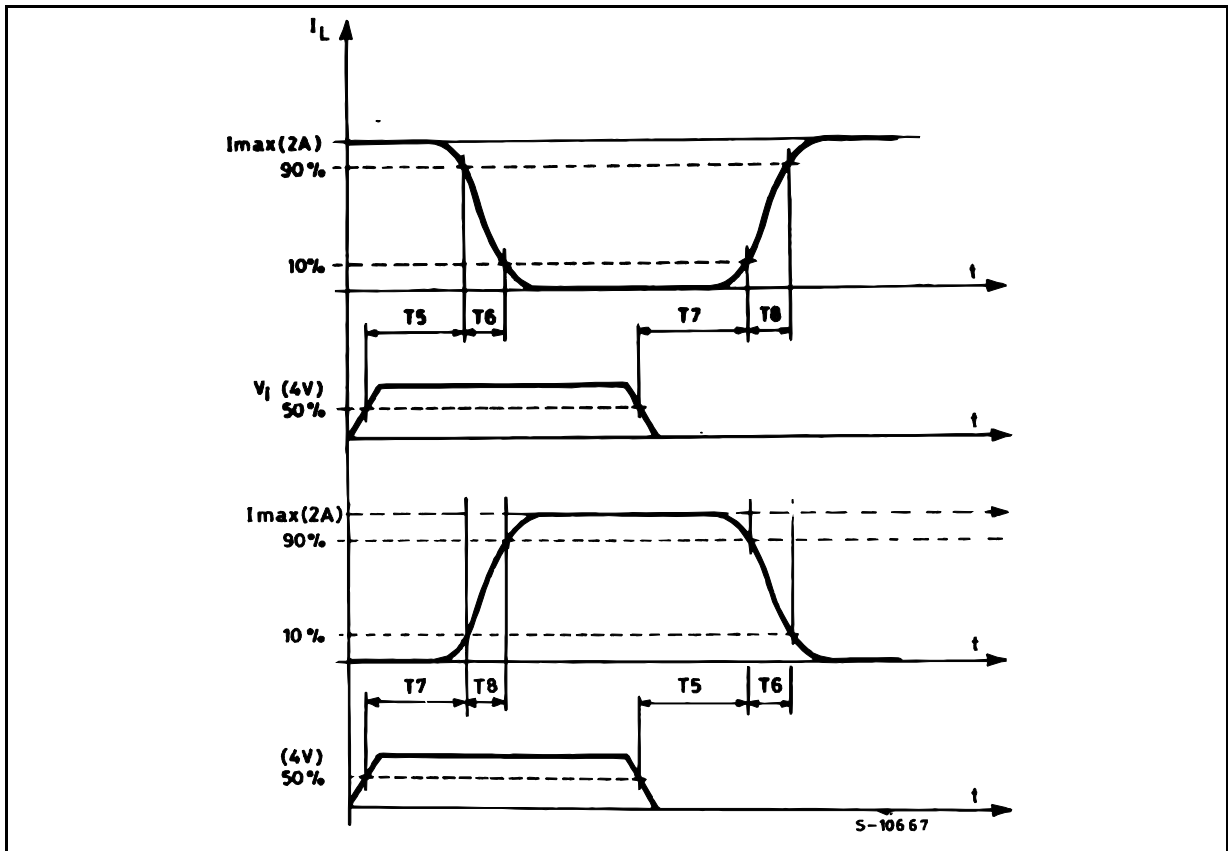


Figure 6 : Bidirectional DC Motor Control.

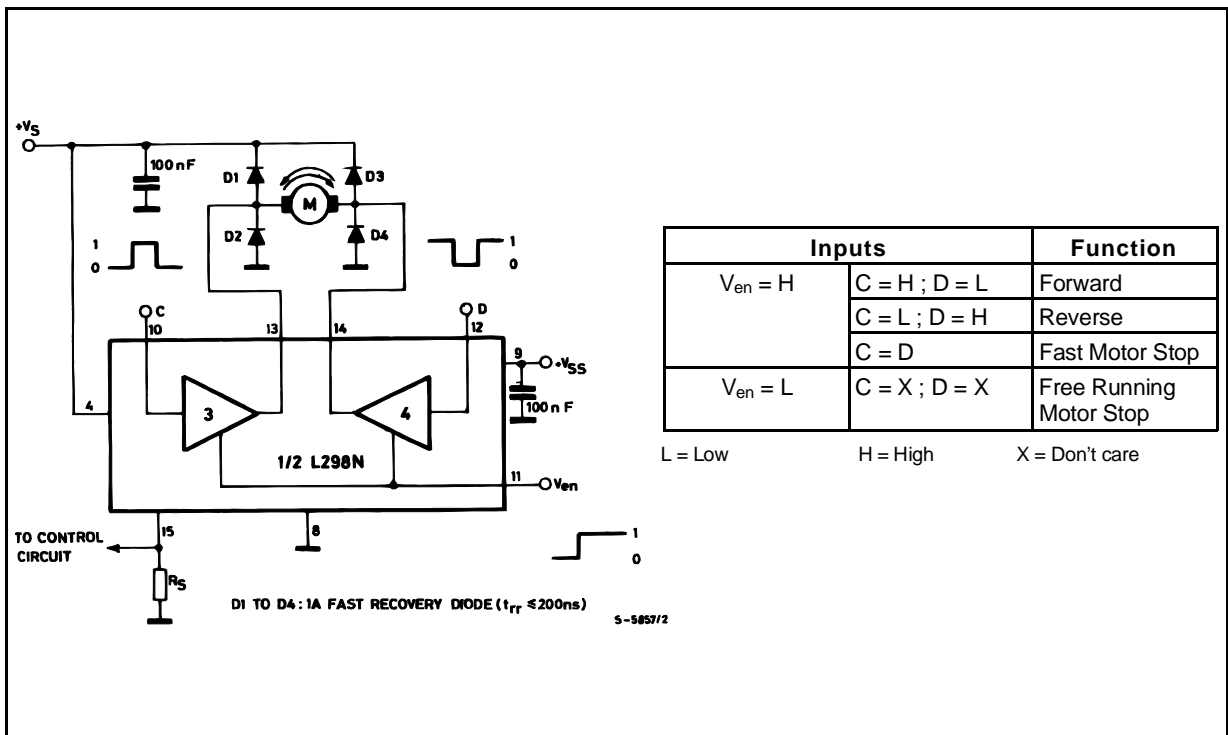
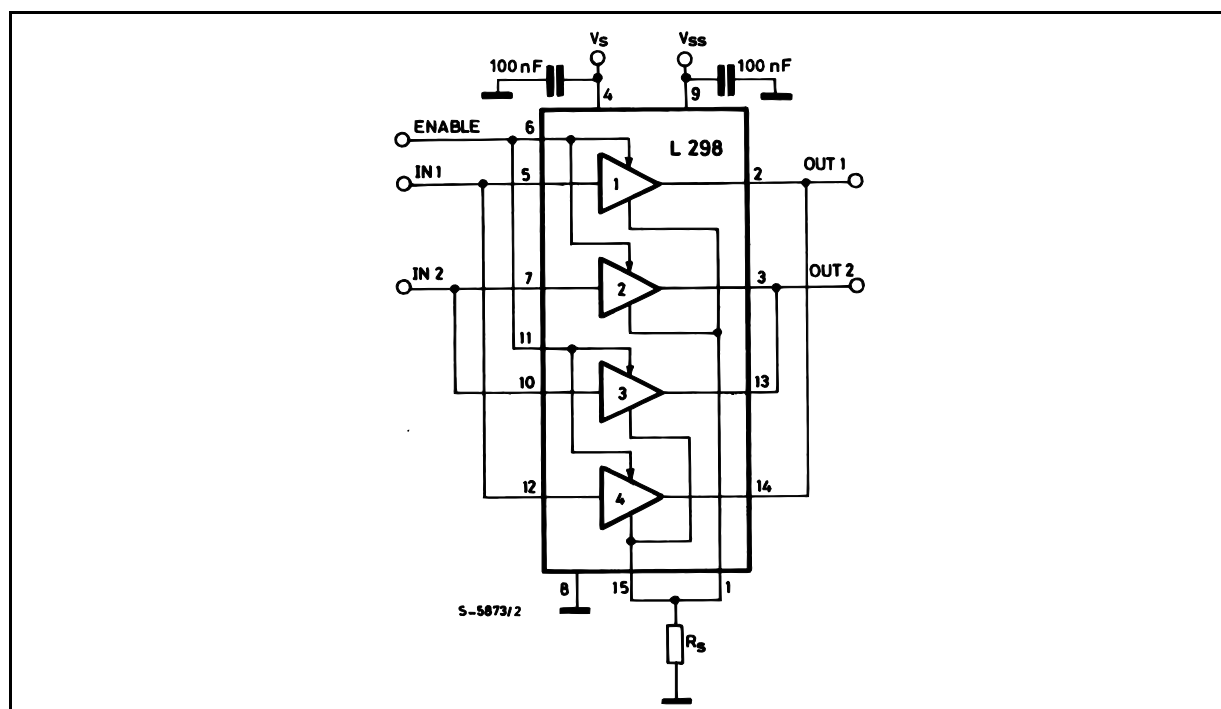


Figure 7 : For higher currents, outputs can be paralleled. Take care to parallel channel 1 with channel 4 and channel 2 with channel 3.



APPLICATION INFORMATION (Refer to the block diagram)

1.1. POWER OUTPUT STAGE

The L298 integrates two power output stages (A ; B). The power output stage is a bridge configuration and its outputs can drive an inductive load in common or differenzial mode, depending on the state of the inputs. The current that flows through the load comes out from the bridge at the sense output : an external resistor (R_{SA} ; R_{SB} .) allows to detect the intensity of this current.

1.2. INPUT STAGE

Each bridge is driven by means of four gates the input of which are In1 ; In2 ; EnA and In3 ; In4 ; EnB. The In inputs set the bridge state when The En input is high ; a low state of the En input inhibits the bridge. All the inputs are TTL compatible.

2. SUGGESTIONS

A non inductive capacitor, usually of 100 nF, must be foreseen between both Vs and Vss, to ground, as near as possible to GND pin. When the large capacitor of the power supply is too far from the IC, a second smaller one must be foreseen near the L298.

The sense resistor, not of a wire wound type, must be grounded near the negative pole of Vs that must be near the GND pin of the I.C.

Each input must be connected to the source of the driving signals by means of a very short path.

Turn-On and Turn-Off : Before to Turn-ON the Supply Voltage and before to Turn it OFF, the Enable input must be driven to the Low state.

3. APPLICATIONS

Fig 6 shows a bidirectional DC motor control Schematic Diagram for which only one bridge is needed. The external bridge of diodes D1 to D4 is made by four fast recovery elements ($t_{tr} \leq 200$ nsec) that must be chosen of a VF as low as possible at the worst case of the load current.

The sense output voltage can be used to control the current amplitude by chopping the inputs, or to provide overcurrent protection by switching low the enable input.

The brake function (Fast motor stop) requires that the Absolute Maximum Rating of 2 Amps must never be overcome.

When the repetitive peak current needed from the load is higher than 2 Amps, a paralleled configuration can be chosen (See Fig.7).

An external bridge of diodes are required when inductive loads are driven and when the inputs of the IC are chopped ; Schottky diodes would be preferred.

L298

This solution can drive until 3 Amps In DC operation and until 3.5 Amps of a repetitive peak current.

On Fig 8 it is shown the driving of a two phase bipolar stepper motor ; the needed signals to drive the inputs of the L298 are generated, in this example, from the IC L297.

Fig 9 shows an example of P.C.B. designed for the application of Fig 8.

Figure 8 : Two Phase Bipolar Stepper Motor Circuit.

This circuit drives bipolar stepper motors with winding currents up to 2 A. The diodes are fast 2 A types.

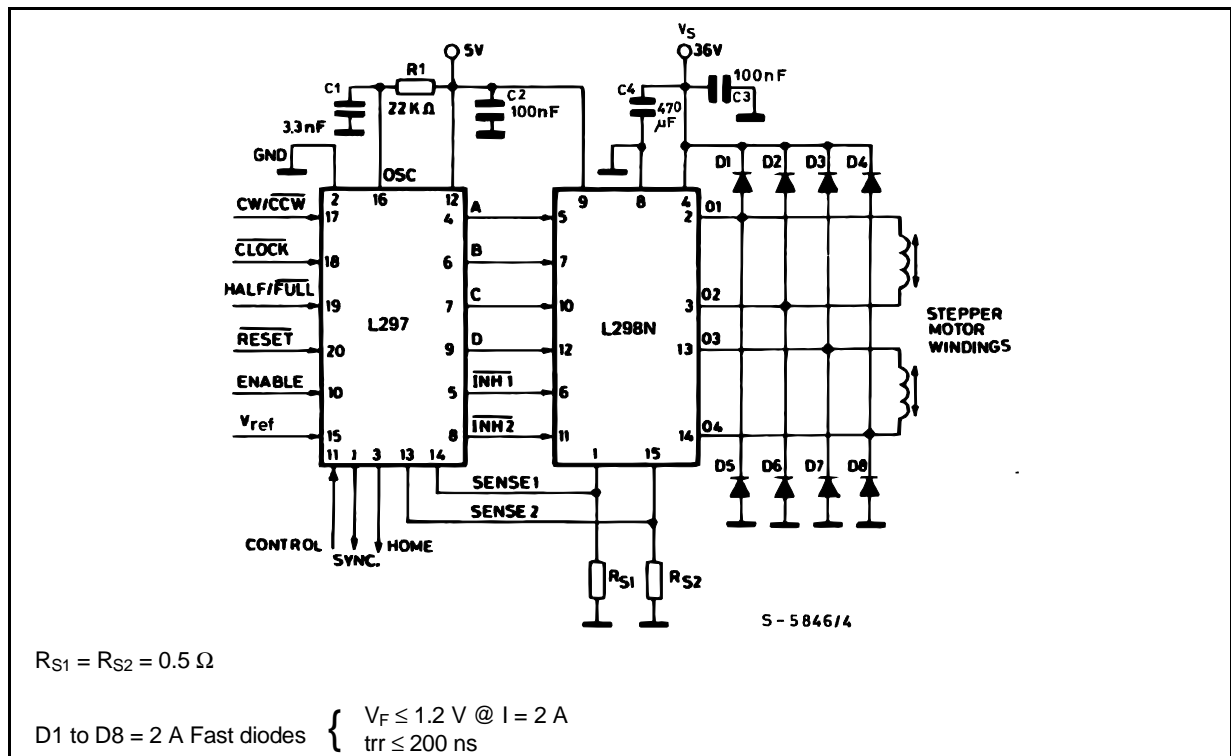


Fig 10 shows a second two phase bipolar stepper motor control circuit where the current is controlled by the I.C. L6506.

Figure 9 : Suggested Printed Circuit Board Layout for the Circuit of fig. 8 (1:1 scale).

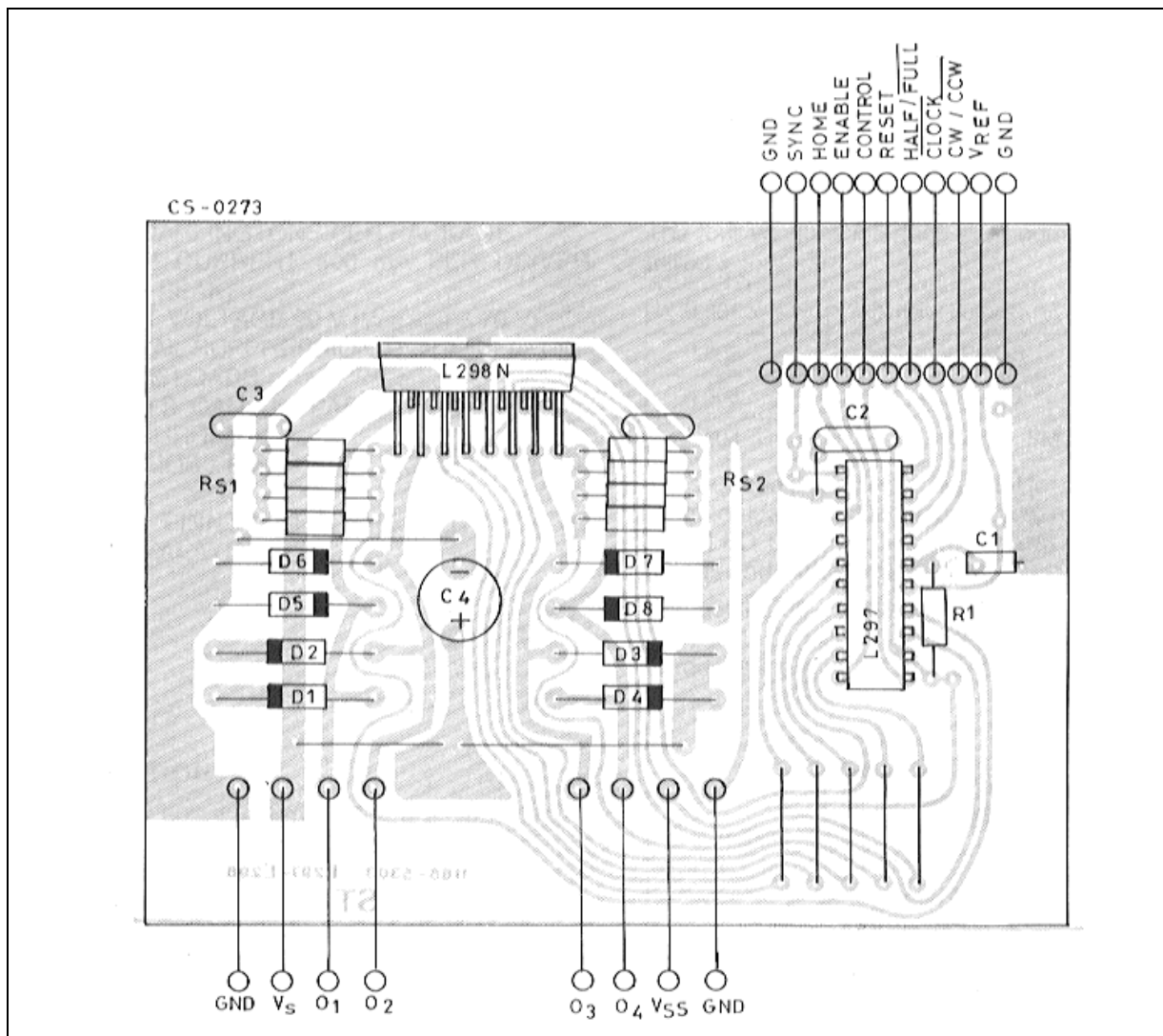
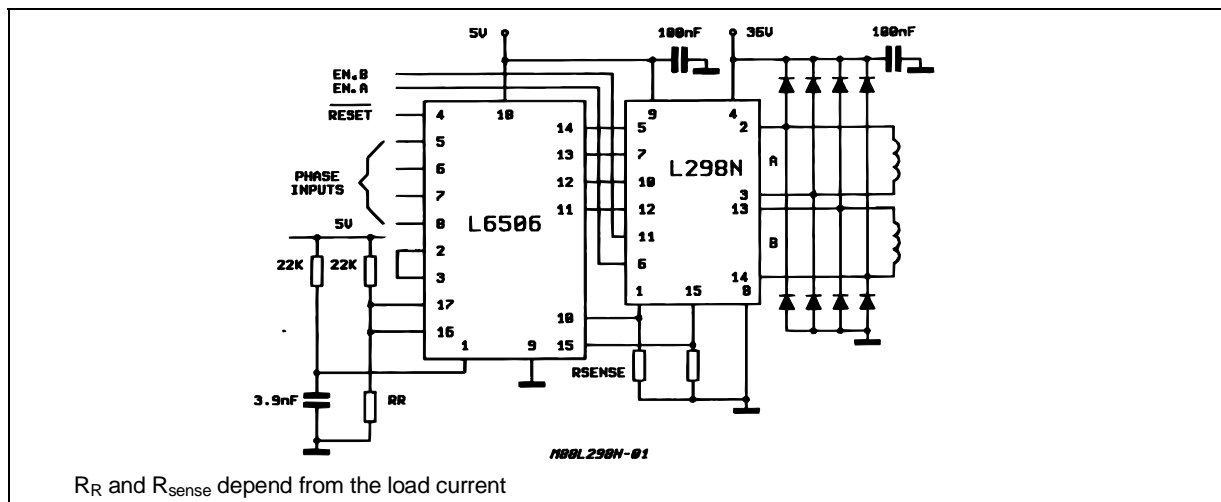


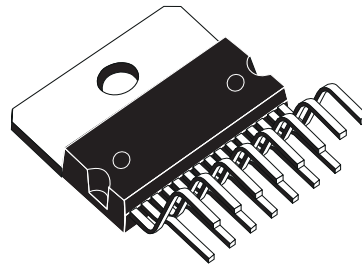
Figure 10 : Two Phase Bipolar Stepper Motor Control Circuit by Using the Current Controller L6506.



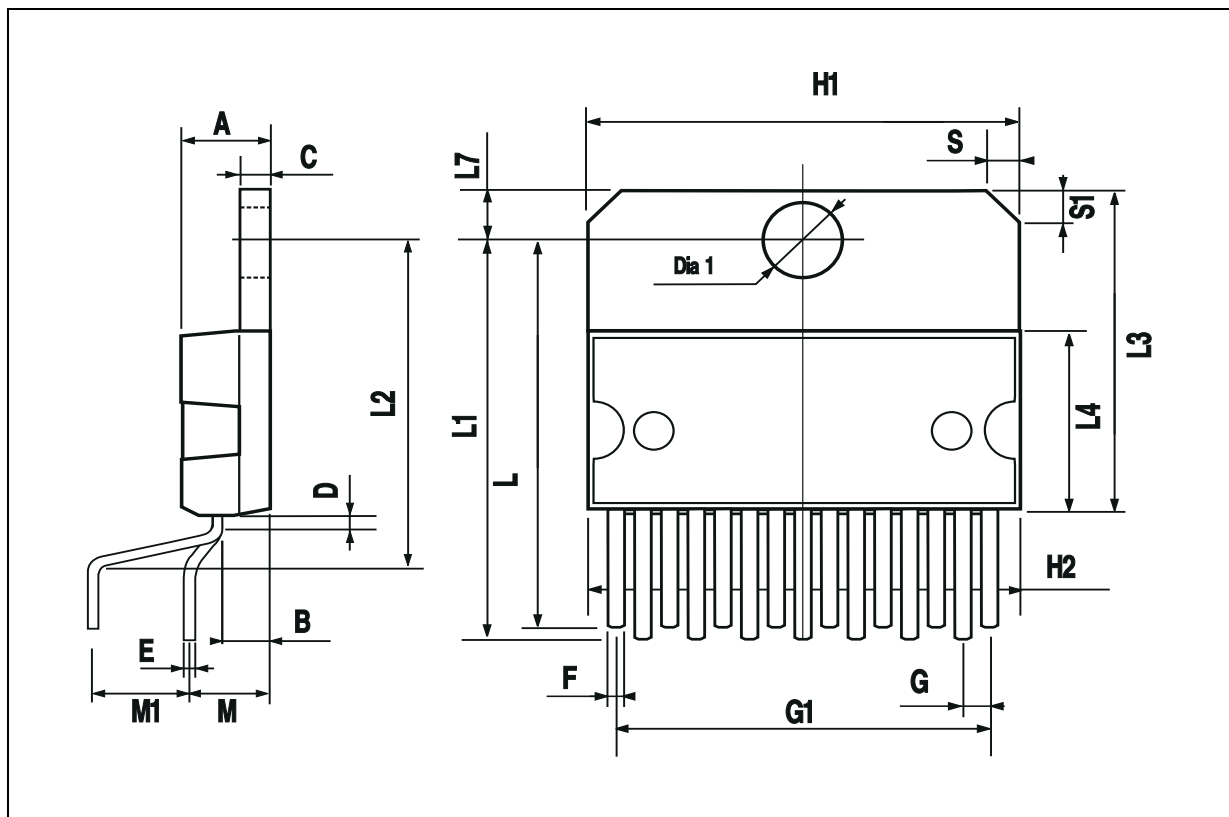
L298

DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			5			0.197
B			2.65			0.104
C			1.6			0.063
D		1			0.039	
E	0.49		0.55	0.019		0.022
F	0.66		0.75	0.026		0.030
G	1.02	1.27	1.52	0.040	0.050	0.060
G1	17.53	17.78	18.03	0.690	0.700	0.710
H1	19.6			0.772		
H2			20.2			0.795
L	21.9	22.2	22.5	0.862	0.874	0.886
L1	21.7	22.1	22.5	0.854	0.870	0.886
L2	17.65		18.1	0.695		0.713
L3	17.25	17.5	17.75	0.679	0.689	0.699
L4	10.3	10.7	10.9	0.406	0.421	0.429
L7	2.65		2.9	0.104		0.114
M	4.25	4.55	4.85	0.167	0.179	0.191
M1	4.63	5.08	5.53	0.182	0.200	0.218
S	1.9		2.6	0.075		0.102
S1	1.9		2.6	0.075		0.102
Dia1	3.65		3.85	0.144		0.152

OUTLINE AND MECHANICAL DATA

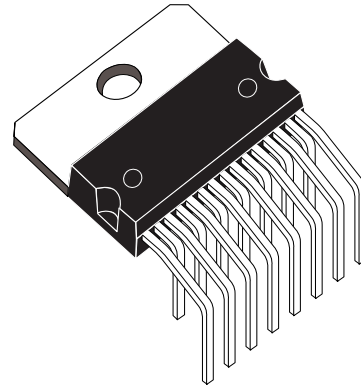


Multiwatt15 V

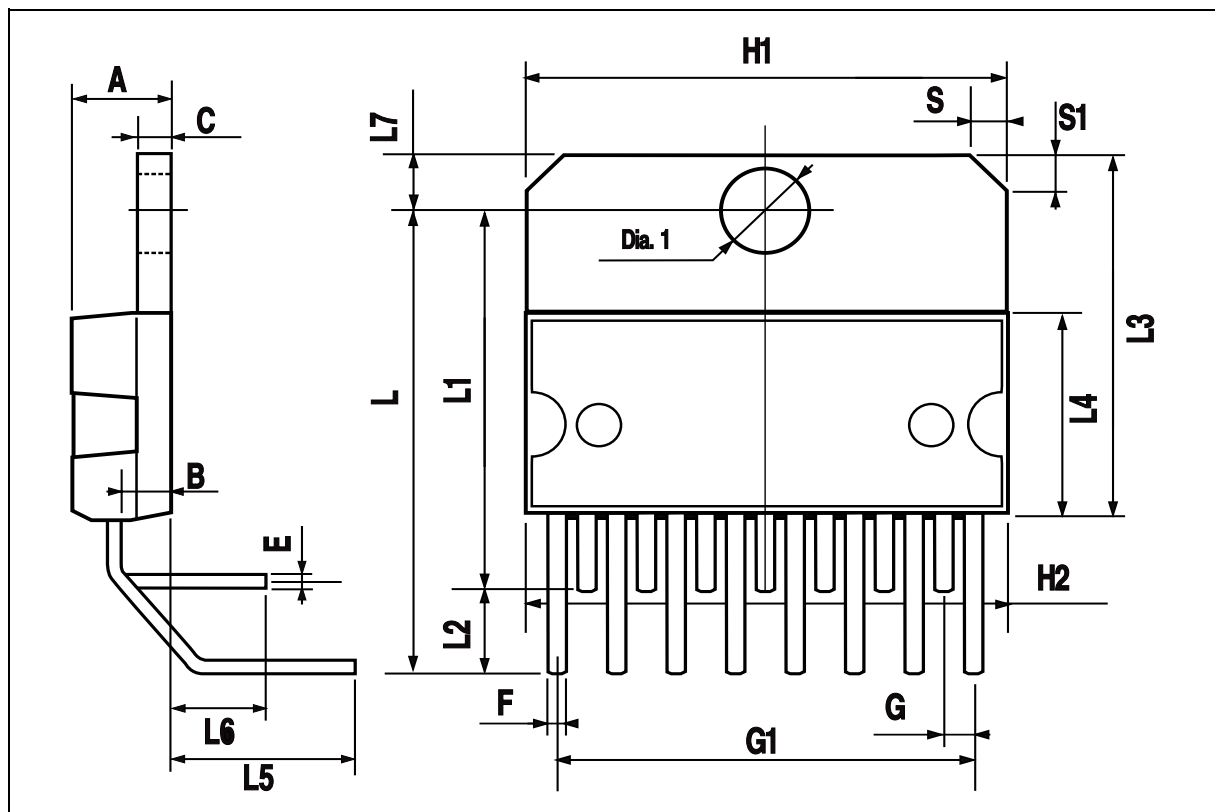


DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			5			0.197
B			2.65			0.104
C			1.6			0.063
E	0.49		0.55	0.019		0.022
F	0.66		0.75	0.026		0.030
G	1.14	1.27	1.4	0.045	0.050	0.055
G1	17.57	17.78	17.91	0.692	0.700	0.705
H1	19.6			0.772		
H2			20.2			0.795
L		20.57			0.810	
L1		18.03			0.710	
L2		2.54			0.100	
L3	17.25	17.5	17.75	0.679	0.689	0.699
L4	10.3	10.7	10.9	0.406	0.421	0.429
L5		5.28			0.208	
L6		2.38			0.094	
L7	2.65		2.9	0.104		0.114
S	1.9		2.6	0.075		0.102
S1	1.9		2.6	0.075		0.102
Dia1	3.65		3.85	0.144		0.152

OUTLINE AND MECHANICAL DATA



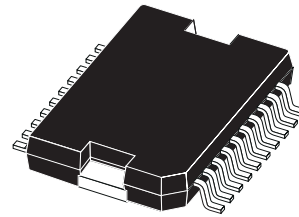
Multiwatt15 H



DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			3.6			0.142
a1	0.1		0.3	0.004		0.012
a2			3.3			0.130
a3	0		0.1	0.000		0.004
b	0.4		0.53	0.016		0.021
c	0.23		0.32	0.009		0.013
D (1)	15.8		16	0.622		0.630
D1	9.4		9.8	0.370		0.386
E	13.9		14.5	0.547		0.570
e		1.27			0.050	
e3		11.43			0.450	
E1 (1)	10.9		11.1	0.429		0.437
E2			2.9			0.114
E3	5.8		6.2	0.228		0.244
G	0		0.1	0.000		0.004
H	15.5		15.9	0.610		0.626
h			1.1			0.043
L	0.8		1.1	0.031		0.043
N	10° (max.)					
S	8° (max.)					
T		10			0.394	

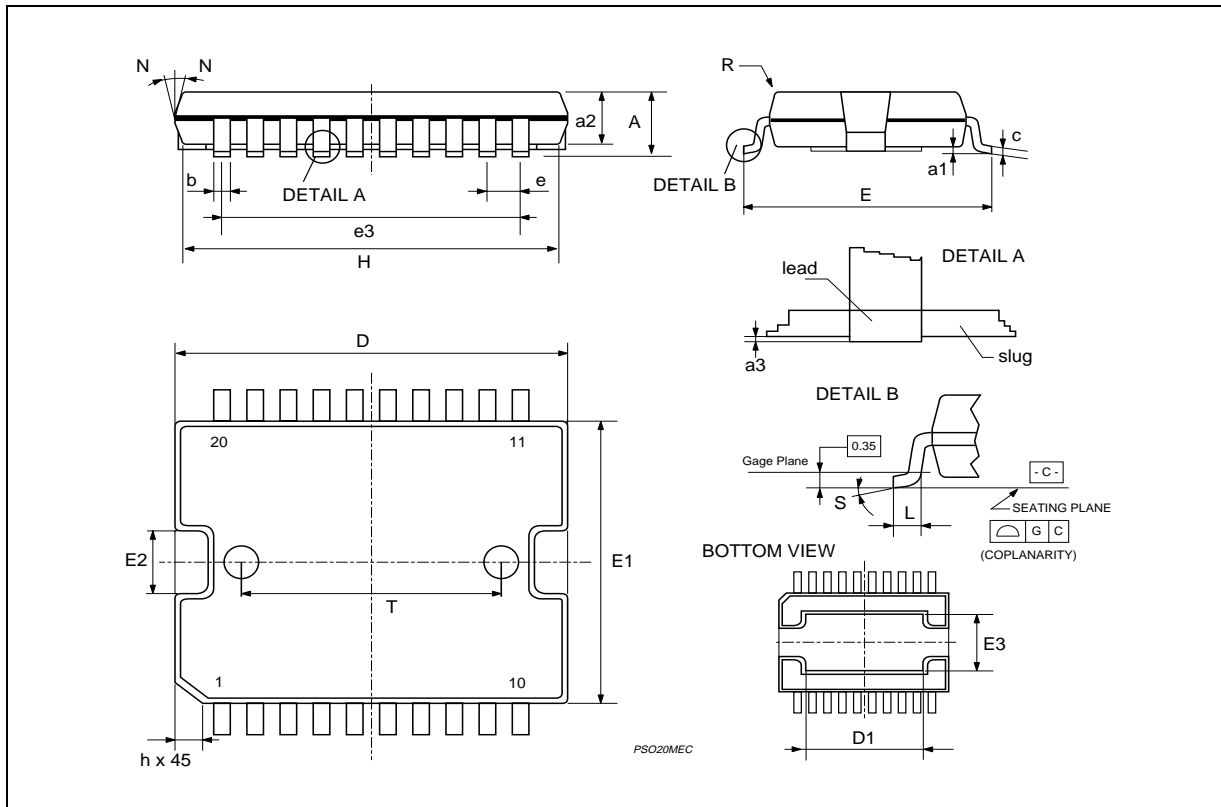
(1) "D and F" do not include mold flash or protrusions.
 - Mold flash or protrusions shall not exceed 0.15 mm (0.006").
 - Critical dimensions: "E", "G" and "a3"

OUTLINE AND MECHANICAL DATA



JEDEC MO-166

PowerSO20



Information furnished is believed to be accurate and reliable. However, STMicroelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of STMicroelectronics. Specification mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. STMicroelectronics products are not authorized for use as critical components in life support devices or systems without express written approval of STMicroelectronics.

The ST logo is a registered trademark of STMicroelectronics
© 2000 STMicroelectronics – Printed in Italy – All Rights Reserved
STMicroelectronics GROUP OF COMPANIES

Australia - Brazil - China - Finland - France - Germany - Hong Kong - India - Italy - Japan - Malaysia - Malta - Morocco -
Singapore - Spain - Sweden - Switzerland - United Kingdom - U.S.A.

<http://www.st.com>

