



PROYEK AKHIR - VE 180626

ALAT PENDETEKSI KUALITAS UDARA

Heni Risa Aisyiah
NRP 10311500010045

Dosen Pembimbing
Ir. Josaphat Pramudijanto, M.Eng.

Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya
2019



FINAL PROJECT - VE 180626

AIR QUALITY DETECTOR

Heni Risa Aisyiah
NRP 10311500010045

Supervisor
Ir. Josaphat Pramudijanto, M.Eng.

*Electrical and Automation Engineering Department
Vocational Faculty
Sepuluh Nopember Institute of Technology
Surabaya 2019*

PERNYATAAN KEASLIAN PROYEK AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Proyek Akhir saya dengan judul "**Alat Pendeteksi Kualitas Udara**" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Januari 2019



Heni Risa Aisyiah
NRP 10311500010045

ALAT PENDETEKSI KUALITAS UDARA

PROYEK AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Memperoleh Gelar Ahli Madya Teknik
Pada
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember

Menyetujui :

Dosen Pembimbing



Ir. Josaphat Pramudijanto M.Eng
NIP. 19621005 199003 1 003

**SURABAYA
JANUARI, 2019**

-----Halaman ini sengaja dikosongkan-----

ALAT PENDETEKSI KUALITAS UDARA

Nama Mahasiswa : Heni Risa Aisyiah
NRP : 10311500010045
Dosen Pembimbing : Ir. Josaphat Pramudijanto, M.Eng
NIP : 19621005 199003 1 003

ABSTRAK

Selama ini, kualitas udara di suatu daerah tertentu belum dapat di monitoring, maka seseorang yang berada pada daerah itu tidak mengerti tentang apakah daerah itu tercemar dan dapat merugikan orang – orang disekitar atau tidak. Salah satu gas yang tidak berbahaya adalah O₂, sementara salah satu gas berbahaya adalah gas CO (Karbon Monoksida), gas CO₂ (Karbon Dioksida), gas NO, NO₂, SO, dan SO₂. Apabila manusia menghirup gas berbahaya tersebut dalam jumlah sedikit maka akan menyebabkan beberapa penyakit seperti sakit kepala, mual-mual. Jika manusia menghirup dalam jumlah yang banyak akan menyebabkan kematian.

Berdasarkan karakteristik gas tersebut maka akan merancang alat yang mendeteksi besarnya konsentrasi gas berbahaya serta dapat memberikan peringatan jika kandungan gas berbahayanya melebihi ambang batas aman bagi kesehatan manusia. Alat ini akan memberikan peringatan kepada masyarakat melalui indikator yang terdapat pada alat tersebut. Mikrokontroler yang digunakan adalah Arduino AT-Mega 2560 yang telah dilengkapi program serta telah dikonfigurasi dengan komponen perangkat keras seperti sensor MQ-135, MG-811 dan MQ-7 yang digunakan untuk mendeteksi gas O₂, CO₂ dan CO serta sensor *Thermocouple* max 6675 tipe-k yang mendeteksi temperatur. Dari sensor tersebut akan ditampilkan melalui LCD dan lampu indikator.

Hasil dari Proyek Akhir ini adalah sistem dapat memonitoring kadar gas dan temperatur di mana telah dilakukan *kalibrasi* dan perbandingan pengukuran dengan presentase *Error* : O₂ = 0,035% CO₂ = 0,27% CO = 0,28%. Pengaplikasian alat yang telah dirangkai dapat di-gunakan untuk memonitoring kadar gas O₂, CO₂ dan CO di daerah pabrik dan pemukiman warga.

Kata kunci : Kualitas udara, Arduino, Daerah Tercemar

-----Halaman ini sengaja dikosongkan-----

AIR QUALITY DETECTOR

Student Name : Heni Risa Aisyiah
Registration Number : 10311500010045
Supervisor : Ir. Josaphat Pramudijanto, M.Eng
ID : 19621005 199003 1 003

ABSTRACT

During this time, air quality in a certain area can not be monitored, then someone who is in the area do not understand about whether the area is polluted and can harm people - people around or not. One of the harmless gases is O₂, while one of the harmful gases is CO (Carbon Monoxide) gas, CO₂ (Carbon Dioxide) gas, NO, NO₂, SO, and SO₂ gas. If humans inhale the harmful gas in small amounts it will cause some diseases such as headache, nausea. If human inhalation in large amounts will cause death.

Based on the characteristics of the gas, it will design a device that detects the amount of dangerous gas concentration and can give a warning if the dangerous gas content exceeds the safe threshold for human health. This tool will provide attention to the community through the indicators found on the device. The microcontroller used is Arduino AT-Mega 2560 which has been equipped with the program and has been configured with hardware components such as MQ-135 sensors, MG-811 and MQ-7 which are used to detect O₂, CO₂ and CO gas and Thermo-couple sensors max 6675 type-k that detects temperature. The sensor will display via LCD and indicator lights.

The results of this Final Project are that the system can monitor gas levels and temperatures where calibration and comparison of measurements have been carried out with an Error percentage of : O₂ = 0.035% CO₂ = 0.27% CO = 0.28% . The application of tools that have been assembled can be used to monitor the levels of O₂, CO₂ and CO gas in the factory area and residential areas.

Keywords : Air Quality, Arduino, Polluted Area

-----Halaman ini sengaja dikosongkan-----

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT yang selalu memberikan rahmat dan hidayah-Nya sehingga Proyek Akhir ini dapat terselesaikan dengan baik. Shalawat serta salam semoga selalu dilimpahkan kepada Rasulullah Muhammad SAW, keluarga, sahabat, dan umat muslim yang senantiasa meneladani beliau.

Proyek Akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan Diploma-3 pada Program Studi Elektronika Industri, Departemen Teknik Elektro Otomasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember Surabaya.

Pada kesempatan ini penulis menyampaikan terima kasih atas segala doa, bantuan dan dukungannya yang telah diberikan selama proses pembuatan Proyek Akhir ini kepada :

1. Kedua orang tua atas dukungan baik spiritual maupun material yang tak ternilai harganya.
2. Bapak Ir. Joko Susila, MT. Ketua departemen Teknik Elektro Otomasi, Fakultas Vokasi- ITS Surabaya.
3. Bapak Ir. Josaphat Pramudijanto, M Eng. Sebagai dosen pembimbing yang telah banyak memberikan ilmu, pengarahan dan bimbingan selama penulis mengerjakan Proyek Akhir ini, serta membimbing saya dengan kesabaran yang tiada batasnya.
4. Semua pihak yang telah membantu saya dalam menyelesaikan Proyek Akhir ini yang tidak dapat kami sebutkan satu persatu.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Proyek Akhir ini. Akhir kata, semoga Proyek Akhir ini dapat bermanfaat dalam pengembangan keilmuan di kemudian hari.

Surabaya, Januari 2019

Henri Risa Aisyiah

-----Halaman ini sengaja dikosongkan-----

DAFTAR ISI

	HALAMAN
HALAMAN JUDUL	i
PERNYATAAN KEASLIAN PROYEK AKHIR	Error! Bookmark not defined.
LEMBAR PENGESAHAN	Error! Bookmark not defined.
ABSTRAK	ix
<i>ABSTRACT</i>	xi
KATA PENGANTAR	xiii
DAFTAR ISI	xv
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Permasalahan	2
1.3 Tujuan	3
1.4 Sistematika Laporan	3
1.5 Relevansi	4
BAB II TEORI DASAR	5
2.1 Pengertian Kualitas Udara	5
2.2 Sensor Gas O ₂ (MQ-135)	6
2.3 Sensor Gas CO ₂ (MG-811)	7
2.4 Sensor Gas CO	9
2.5 DC-DC <i>Converter</i>	10
2.6 LCD	11
2.7 Arduino Mega	11
2.8 LED RGB	13
2.9 Sensor Suhu DS-18B20	14
BAB III PERANCANGAN <i>HARDWARE</i> DAN <i>SOFTWARE</i>	16
3.1 Diagram Fungsional Sistem	16
3.2 Perancangan <i>Hardware</i>	17
3.2.1. Perancangan Kotak Kontrol	17
3.2.2. Perancangan Sensor Gas O ₂	19
3.2.3. Perancangan Sensor Gas CO	20
3.2.4. Perancangan Sensor Gas CO ₂	21

3.3 Perancangan <i>Software</i>	22
BAB IV PENGUJIAN DAN ANALISA	25
4.1 Pengujian Sensor Karbon Dioksida (CO ₂)	25
4.2 Pengujian Sensor Oksigen (O ₂)	29
4.3 Pengujian Sensor Karbon Monoksida (CO)	33
4.4 Pengujian Sensor Suhu DS-18B20	37
4.5 Pengujian Keseluruhan.....	38
BAB V PENUTUP	42
5.1 Kesimpulan	42
5.2 Saran.....	42
DAFTAR PUSTAKA	44
LAMPIRAN A	A-1
LAMPIRAN B.....	B-1
DAFTAR RIWAYAT HIDUP	B-42

DAFTAR GAMBAR

HALAMAN

Gambar 1.1 Warga Purwarkarta Keracunan Gas Kimia	2
Gambar 2.1 Warga Desa Kutamekar Keracunan.....	6
Gambar 2.2 Sensor MQ-135.....	7
Gambar 2.3 Sensor MG-811.....	9
Gambar 2.4 Sensor MQ-7	10
Gambar 2.5 DC-DC <i>Converter</i>	10
Gambar 2.6 Konfigurasi Pin LCD	11
Gambar 2.7 Arduino Mega.....	13
Gambar 2.8 Lampu LED RGB	14
Gambar 2.9 Sensor Suhu DS-18B20	15
Gambar 3.1 Diagram Fungsional.....	16
Gambar 3.2 Perancangan <i>Box</i> Tampak Atas	18
Gambar 3.3 Perancangan <i>Box</i> Tampak Depan	18
Gambar 3.4 Rangkaian Sensor O ₂	19
Gambar 3.5 Rangkaian Sensor CO	20
Gambar 3.6 Rangkaian Sensor CO ₂	21
Gambar 3.7 <i>Setting Serial Port</i>	23
Gambar 3.8 <i>Flowchart Rangkaian Arduino</i>	24
Gambar 4.1 Program Sensor CO ₂	26
Gambar 4.2 Rangkaian Pengujian Sensor CO ₂	27
Gambar 4.3 Program Konversi Sensor CO ₂	28
Gambar 4.4 Program Sensor O ₂	30
Gambar 4.5 Rangkaian Pengujian Sensor O ₂	31
Gambar 4.6 Program Konversi pada Arduino	32
Gambar 4.7 Program Gas CO.....	34
Gambar 4.8 Rangkaian Pengujian Sensor CO.....	34
Gambar 4.9 Program Konversi Sensor CO	36
Gambar 4.10 Program Sensor Temperatur.....	37
Gambar 4.11 Kalibrasi Sensor Suhu	37
Gambar 4.12 Pengambilan Data Pada Alat	40

DAFTAR TABEL

HALAMAN

Tabel 2.1 Spesifikasi Arduino Mega	12
Tabel 3.1 Pin Pada Sensor O ₂	20
Tabel 3.2 Pin Sensor CO	21
Tabel 3.3 Pin Sensor CO ₂	22
Tabel 4.1 Hasil Kalibrasi Sensor CO ₂	27
Tabel 4.2 Hasil Kalibrasi Sensor CO ₂	29
Tabel 4.3 Hasil Kalibrasi Sensor O ₂	31
Tabel 4.4 Hasil Konversi Sensor O ₂	33
Tabel 4.5 Hasil Kalibrasi Sensor CO	35
Tabel 4.6 Hasil Konversi Sensor CO	36
Tabel 4.7 Hasil Kalibrasi Sensor Suhu	38
Tabel 4.8 Persentase <i>Error</i> Pada Sensor	38
Tabel 4.9 Hasil Pengujian di Area Parkiran Sepeda Motor	39

-----Halaman ini sengaja dikosongkan-----

BAB I

PENDAHULUAN

1.1 Latar Belakang

Udara merupakan salah satu unsur penting. Baik buruknya kualitas udara berdampak pada lingkungan dan dapat mempengaruhi kondisi atau kesehatan seseorang. Kualitas udara dibagi menjadi 2, yaitu *outdoor* dan *indoor*. Permasalahan yang sering muncul adalah *outdoor* karena polusi udara dari asap dan kendaraan serta sampah yang menumpuk dan tidak dibuang pada tempatnya menyebabkan kualitas udara menjadi menurun. Sementara untuk *indoor* karena ventilasi dan kelembapan udara. Jika udara yang kita hirup mengandung gas – gas berbahaya yang melebihi ambang batas, maka efek yang ditimbulkan akan sangat berbahaya, yaitu dapat mengganggu metabolisme kerja dari organ tubuh kita. Efek terburuknya yaitu menyebabkan kematian.

Terdapat beberapa gas yang terdapat pada udara, yaitu gas O₂ (Oksigen), gas CO₂ (Karbon Dioksida), dan gas CO (Karbon Monoksida). Ketiga gas tersebut sangat berpengaruh terhadap baik buruknya kualitas udara. Salah satu gas yang tidak berbahaya adalah O₂ (Oksigen), sementara salah satu gas berbahaya adalah gas CO (Karbon Monoksida), dan gas CO₂ (Karbon Dioksida). O₂ merupakan gas yang sangat diperlukan bagi kehidupan manusia, yang kita ketahui sebagai gas yang terdapat dalam sistem peredaran darah. Sementara gas CO₂ merupakan gas yang tak kalah penting dalam kehidupan sehari-hari. Gas CO₂ merupakan sebuah gas yang dihasilkan dari sebuah proses seperti misalnya proses pembakaran. Proses pembusukan yang terjadi pada bahan-bahan tertentu atau bahkan sampah sehari-hari juga dapat menghasilkan gas karbon dioksida. Yang terakhir adalah gas CO, gas ini juga merupakan gas yang berbahaya jika dihirup oleh seseorang. Gas CO dapat berikatan dengan hemoglobin di dalam tubuh sehingga pengikatan oksigen dalam tubuh akan terganggu. Jika CO dihirup dalam kadar yang tinggi akan menimbulkan kematian. Ketiga gas tersebut merupakan gas yang tidak berwarna dan tidak berbau sehingga akan sulit dilihat atau dirasakan apabila kandungannya melebihi batas aman manusia.

Dengan adanya aktifitas di daerah pabrik serta beberapa daerah disekitar perumahan warga, maka dapat dikatakan dapat membahayakan warga yang tidak tahu mengenai keadaan udara

didaerahnya. Maka dari itu, dengan adanya alat pendeteksi kualitas udara yang bersifat *portable* ini diharapkan dapat memudahkan warga yang tidak mengerti mengenai kualitas di daerahnya. Dengan begitu jika suatu saat kandungan di daerah di sekitarnya memburuk, maka alat tersebut akan memberikan peringatan.

Gas Kimia Bocor, Puluhan Warga Purwakarta Keracunan

Didin Jalaludin

Selasa, 1 November 2016 - 19:59 WIB



Gambar 1.1 Warga Purwakarta Keracunan Gas Kimia

Salah satu contoh pencemaran udara di daerah pemukiman warga dan daerah pabrik yang terjadi yaitu pada gambar 1.1. Puluhan warga kampung Ciroyom, RT 03/05 Desa Cicadas, Kecamatan Babakan Cikao, Kabupaten Purwakarta mengalami keracunan karena menghirup gas kimia yang bocor dari *PT South Pacific Viscos (SPV)* yang ada di sekitar tempat tinggal warga. Beberapa korban banyak diantaranya adalah anak-anak. [1]

Oleh karena itu, akan lebih baik jika terdapat alat untuk mendeteksi gas beracun yang dapat membuat masyarakat berjaga-jaga terhadap bahayanya gas beracun tersebut.

1.2 Permasalahan

Berdasarkan latar belakang diatas maka permasalahan utama pada Proyek akhir ini adalah pembuatan alat untuk mendeteksi kandungan gas yang berbahaya terutama gas O_2 (Oksigen), gas CO_2 (Karbon Dioksida), dan gas CO (Karbon Monoksida). Alat yang

dirancang harus mampu membaca seberapa besar gas yang terdapat dalam alam bebas agar terhindar dari keracunan gas. Banyaknya gas yang tercemar dapat mempengaruhi kualitas udara, sehingga perlu dibuat alat yang mampu untuk mendeteksi gas yang terdapat pada udara dengan tepat. Udara yang telah diberi sensor gas O₂, gas CO₂, dan gas CO. Gas yang diukur harus sesuai dengan konsentrasi gas yang terdapat pada daerah yang tercemar, pada alat ini yang dimaksud gas tercemar adalah udara yang terdapat kandungan CO₂ dan CO yang melebihi ambang batas aman bagi manusia. Dengan begitu sensor akan bekerja dan mengukur kandungan konsentrasi gas yang terdapat pada daerah tersebut melalui data ppm dari sensornya.

1.3 Tujuan

Pembuatan alat pendeteksi kualitas udara bertujuan untuk mengukur, merancang dan membuat alat untuk konsentrasi gas O₂, CO₂, dan CO agar dapat memudahkan manusia dalam menentukan kualitas udara disuatu ruangan

1.4 Sistematika Laporan

Pembahasan Proyek Akhir ini akan dibagi menjadi lima Bab dengan sistematika sebagai berikut:

Bab I Pendahuluan

Bab ini meliputi latar belakang, permasalahan, batasan masalah, tujuan, metodologi penelitian, sistematika laporan, dan relevansi.

Bab II Teori Dasar

Bab ini membahas teori dasar, teori penunjang dari peralatan-peralatan yang digunakan dalam pembuatan alat.

Bab III Perancangan Sistem

Membahas perencanaan dan pembuatan perangkat keras yang meliputi rangkaian-rangkaian, dan perangkat lunak yang akan digunakan untuk menjalankan alat tersebut.

Bab IV Pengujian dan Analisa Sistem

Bab ini membahas tentang pengukuran, pengujian alat dan analisa data yang didapat dalam pengujian alat.

Bab V Penutup

Bab ini berisi kesimpulan dan saran dari hasil pembahasan yang telah diperoleh.

1.5 Relevansi

Dengan adanya Proyek Akhir ini diharapkan dapat memonitoring kualitas udara unruk kandungan gas O₂, CO₂ dan CO di daerah yang tercemar.

BAB II

TEORI DASAR

Pada bab ini akan dibahas teori dasar dan teori penunjang terkait perangkat dan bahan yang digunakan dalam Proyek Akhir ini.

2.1 Pengertian Kualitas Udara [1]

Udara merupakan salah satu faktor penting sumber kehidupan bagi makhluk hidup di bumi selain air dan tanah. Kelestarian dan kebersihan udara di alam bebas ini perlu untuk dijaga dan dilindungi. Udara yang dihirup oleh makhluk hidup dikenal dengan istilah udara ambien, yang dalam pengertiannya adalah udara bebas di permukaan bumi yang dibutuhkan dan mempengaruhi kesehatan manusia, makhluk hidup dan unsur lingkungan hidup lainnya. Perkembangan teknologi dan pembangunan yang pesat, memberikan dampak negatif yaitu mengakibatkan kualitas udara semakin menurun akibat terkontaminasi oleh polutan yang berasal dari aktivitas pembakaran sampah, gas buang kendaraan bermotor, dan kegiatan industri yang menghasilkan polusi.

Terciptanya lingkungan yang layak terdapat dalam UUD Pasal 28 H ayat 1: “Setiap orang berhak hidup sejahtera lahir dan batin, bertempat tinggal dan mendapatkan lingkungan hidup yang baik dan sehat serta berhak memperoleh pelayanan kesehatan”. Namun, implementasi pada pasal tersebut belum sungguh-sungguh dilakukan oleh pemerintah daerah dan masyarakatnya sendiri. Hasil pemantauan kualitas udara yang dilakukan pada bulan Januari hingga Juni oleh sebuah organisasi lingkungan pada bulan Juli 2017 menunjukkan bahwa kualitas udara di Jabodetabek terindikasi memasuki level tidak sehat atau tercemar dengan skor lumayan parah. Sampel diambil dari di 21 lokasi sekitar Jabodetabek.

Jika kualitas udara di bawah ambang normal, maka kesehatan masyarakat di sekitarnya akan terganggu seperti pusing disertai dengan mual-mual dan badan lemas hingga akhirnya tubuh tidak kuat berdiri yang akhirnya roboh. Hal ini pernah terjadi di Desa Kutamekar, Kecamatan Ciampel, Kabupaten Karawang, Jawa Barat pada hari Rabu, 8 November 2017, belasan warga di desa tersebut mengalami keracunan akibat kebocoran gas milik PT Pidodeli. Pada Gambar 2.1 merupakan gambar dari seorang korban yang mengalami keracunan gas milik PT Pidodeli. [1]

Belasan Warga Karawang Keracunan Akibat Kebocoran Gas

Nila Kusuma

Rabu, 8 November 2017 - 12:34 WIB



[d/ackk?sa=L&ai=CO0jw7TkyX0iGk47ewwTX5ZiAbCW4NHU8_Gr8MEIsJAFaEgfsjkjWDpouKD4A2gAd3J4Z8DyAEdqQJCTrujP](https://www.detik.com/indonesia/bencana/d/ackk?sa=L&ai=CO0jw7TkyX0iGk47ewwTX5ZiAbCW4NHU8_Gr8MEIsJAFaEgfsjkjWDpouKD4A2gAd3J4Z8DyAEdqQJCTrujP)

Gambar 2.1 Warga Desa Kutamekar Keracunan

2.2 Sensor Gas O₂ (MQ-135) [2]

Oksigen atau zat asam adalah unsur kimia dalam sistem tabel periodik yang mempunyai lambang O dan nomor atom 8. Ia merupakan unsur golongan kalkogen dan dapat dengan mudah bereaksi dengan hampir semua unsur lainnya.

Oksigen adalah suatu komponen gas dan unsur vital dalam proses metabolisme untuk mempertahankan kelangsungan hidup seluruh sel-sel secara normal yang diperoleh dengan cara menghirup O₂ setiap hari. Oksigenasi adalah memenuhi kebutuhan oksigen dalam tubuh dengan cara melancarkan saluran masuknya oksigen sehingga konsentrasi oksigen dapat meningkat dalam tubuh.

Dalam kondisi normal oksigen adalah gas tak berwarna, tidak berbau, dan insipid. Oksigen adalah bagian dari kelompok kecil gas yang secara harfiah bersifat paramagnetik, dan oksigen adalah gas yang paling paramagnetik dari semuanya. Oksigen cair juga sedikit paramagnetik.

MQ-135 *Air Quality Sensor* adalah sensor yang memonitor kualitas udara untuk mendeteksi gas O₂ (Oksigen), gas amonia (NH₃), natrium-(di)oksida (NO_x), alkohol / ethanol (C₂H₅OH), benzena (C₆H₆), karbondioksida (CO₂), gas belerang / sulfur-hidroksida (H₂S) dan asap / gas-gas lainnya di udara.

Sensor ini melaporkan hasil deteksi kualitas udara berupa nilai resistensi analog di pin keluarannya. Pin keluaran ini bisa disambungkan dengan pin ADC (*analog-to-digital converter*) di mikrokontroler / pin *analog input* Arduino. Sensor MQ-135 dapat dilihat pada Gambar 2.2

Spesifikasi:

- Sensitivitas tinggi dengan area deteksi luas
- *Detection gas* : Oksigen, Amonia, Benze Steam, Sulfide
- *Concentration* : 10 - 10000 ppm
- *Loop Voltage (Vc)* : <24V
- *Heater Voltage (Vh)* : 5V
- *Load Resistance (RL)*: Adjustable
- *Heater resistance (Rh)* : 31 Ohm
- *Heater Consumption* : <900mW
- *Standard operating voltage* : 5V
- *Preheat time* : >48 jam



Gambar 2.2 Sensor MQ-135

2.3 Sensor Gas CO₂ (MG-811) [3]

Karbon dioksida atau karbon dioksida (CO₂) adalah jenis senyawa kimia yang terdiri dari dua atom oksigen yang terikat secara kovalen dengan sebuah atom karbon. CO₂ dalam bentuk gas pada suhu standar dan ada di atmosfer bumi.

Karbon dioksida dihasilkan oleh semua hewan, tumbuhan, jamur dan mikroorganisme selama respirasi dan digunakan oleh tumbuhan selama fotosintesis. Oleh karena itu, karbon dioksida merupakan komponen penting dalam siklus karbon. Karbon dioksida juga diproduksi sebagai produk sampingan dari pembakaran bahan

bakar fosil. Karbon dioksida anorganik dikeluarkan dari gunung berapi dan proses geotermal lainnya seperti mata air panas.

Karbon dioksida adalah gas yang tidak berwarna dan tidak berbau. Ketika dihirup pada konsentrasi yang lebih tinggi dari konsentrasi karbon dioksida di atmosfer, ia akan terasa asam di mulut dan di hidung dan tenggorokan. Konsentrasi yang lebih besar dari 500 ppm tidak baik untuk kesehatan, sedangkan konsentrasi lebih besar dari 5000 ppm dapat membahayakan kehidupan hewan.

Sensor MG-811 merupakan modul sensor gas yang dapat digunakan untuk menentukan kadar karbon dioksida yang terdapat pada udara. Modul ini berbasis sensor MG-811 yang mampu melakukan pendeteksian gas karbon dioksida dengan range 350 - 10000 ppm. Sensor MG-811 cocok digunakan pada sistem pemantauan kualitas udara dalam ruangan, sistem kontrol proses fermentasi, dll. Sensor MG-811 dapat dilihat dari Gambar 2.3

Spesifikasi :

- Tegangan : 5 VDC.
- Target gas : karbon dioksida (CO₂).
- *Range* deteksi : 350 - 10000 ppm.
- Menggunakan ADC 10-bit untuk konversi data *analog* dari sensor.
- Memiliki *output* berupa data *digital* dengan nilai 0 - 1023 (hasil konversi ADC).
- Memiliki fitur kendali *on/off* dengan 2 mode kerja pilihan yaitu *hysteresis* dan *window*.
- Pin I/O yang kompatibel dengan level tegangan TTL dan CMOS.
- Memiliki 2 buah LED sebagai indikator.
- Dilengkapi dengan rangkaian EMI *filter* untuk mengurangi gangguan elektromagnetik.



Gambar 2.3 Sensor MG-811

2.4 Sensor Gas CO [4]

Merupakan gas yang tak berwarna, tak berbau dan bersifat sangat beracun. Karena gas ini tak berbau dan tak berwarna, orang dapat tertidur tanpa curiga bahwa ia sedang mulai keracunan. Karbon monoksida mencegah *hemoglobin* mengangkut oksigen untuk dipasok ke jaringan tubuh. Ikatan *hemoglobin* dan karbon monoksida lebih kuat daripada ikatan antara *hemoglobin* dan oksigen. Gejala akibat polusi karbon monoksida ialah napas pendek, sakit kepala, mudah lelah, mudah tersinggung, kejang dan koma, yang dapat berakhir dengan kematian apabila konsentrasi gas ini masuk ke dalam tubuh dalam jumlah yang sangat tinggi. Secara industri, gas ini dibuat dari arang, yang dipanaskan dengan karbon dioksida, oksigen (kuantitas terbatas) atau air.

MQ-7 merupakan sensor gas yang digunakan dalam peralatan untuk mendeteksi gas karbon monoksida (CO) dalam kehidupan sehari-hari, industri, atau mobil. Fitur dari sensor gas MQ-7 ini adalah mempunyai sensitivitas yang tinggi terhadap karbon monoksida (CO), stabil, dan berumur panjang. Sensor ini menggunakan catu daya *heater* : 5V AC/DC dan menggunakan catu daya rangkaian : 5VDC, jarak pengukuran : 20 - 2000ppm untuk ampu mengukur gas karbon monoksida. Sensor MQ-7 dapat dilihat pada Gambar 2.4

Spesifikasi:

- VC/(Tegangan Rangkaian) = $5V \pm 0,1$
- VH (H)/ Tegangan Pemanas (Tinggi) = $5V \pm 0,1$
- VH (L)/ Tegangan Pemanas (Rendah) = $1,4V \pm 0,1$
- RL/Resistansi Beban Dapat disesuaikan

- TH (H) Waktu Pemanasan (Tinggi) = 60 ± 1 seconds
- TH (L) Waktu Pemanasan (Rendah) = 90 ± 1 seconds
- Suhu Penggunaan = -20°C - 50°C
- Suhu Penyimpanan = -20°C - 50°C
- Kelembapan Relatif = kurang dari 95%RH
- O₂ Konsentrasi Oksigen = 21% (*stand condition* / Konsentrasi Oksigen dapat mempengaruhi sensitivitas



Gambar 2.4 Sensor MQ-7

2.5 DC-DC Converter [5]

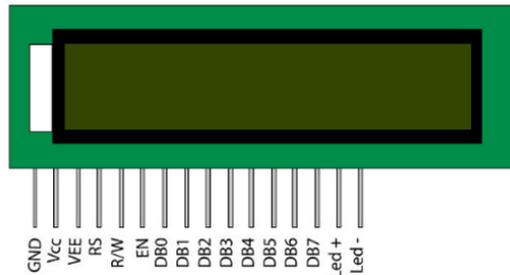
Konverter DC-DC berlaku seperti halnya trafo/transformer yang mengubah tegangan AC tertentu ke tegangan AC yang lebih tinggi atau lebih rendah. Tidak ada peningkatan ataupun pengurangan daya masukan selama pengkonversian bentuk energi listriknya. DC-DC Converter dapat dilihat pada Gambar 2.5



Gambar 2.5 DC-DC Converter

2.6 LCD [6]

LCD (*Liquid Crystal Display*) adalah salah satu komponen elektronika yang berfungsi sebagai tampilan suatu data, baik karakter, huruf ataupun grafik. Dipasaran tampilan LCD sudah tersedia dalam bentuk modul yaitu tampilan LCD beserta rangkaian pendukungnya. LCD mempunyai pin data, kontrol catu daya, dan pengatur kontras tampilan. Tampilan LCD dapat dilihat pada Gambar 2.6



Gambar 2.6 Konfigurasi Pin LCD

LCD juga merupakan perangkat *display* yang paling umum dipasangkan di Mikrokontroler, Mengingat ukurannya yang kecil dan kemampuannya menampilkan karakter atau grafik yang lebih dibandingkan *display seven-segmen*. CD mutlak diperlukan sebagai sumber pemberi informasi utama, misalnya alat pengukur kadar gula darah, penampil waktu jam, penampil counter putaran motor industri dan lain-lain.

2.7 Arduino Mega [7]

Arduino adalah pengendali mikro *single-board* yang bersifat *open-source*, diturunkan dari *wiring platform*, dirancang untuk memudahkan penggunaan elektronika dalam berbagai bidang. *Hardware* dalam arduino memiliki prosesor Atmel AVR dan menggunakan *software* dan bahasa sendiri.

Arduino Mega yang merupakan papan pengembangan mikrokontroler yang berbasis Arduino dengan menggunakan *chip* ATmega2560. *Board* ini memiliki pin I/O yang cukup banyak, sejumlah 54 buah digital I/O pin (15 pin diantaranya adalah PWM), 16 pin *analog input*, 4 pin UART (*serial port hardware*). Dilengkapi

dengan sebuah *oscillator* 16 Mhz, sebuah *port USB*, *power jack DC*, *ICSP header*, dan tombol reset. Keterangan pada Arduino Mega dapat dilihat pada Tabel 2.1

Tabel 2.1 Spesifikasi Arduino Mega

Spesifikasi	Keterangan
<i>Chip</i> mikrokontroler	ATmega2560
Tegangan operasi	5V
Tegangan <i>input</i>	7V - 12V
Tegangan <i>input</i>	6V - 20V
<i>Digital I/O</i> pin	54 buah, 6 diantaranya menyediakan PWM output
<i>Analog Input</i> pin	16 buah
Arus DC per pin I/O	20 mA
Arus DC pin 3.3V	50 mA
<i>Memori Flash</i>	256 KB, 8 KB telah digunakan untuk <i>bootloader</i>
SRAM	8 KB
EEPROM	4 KB
<i>Clock speed</i>	16 Mhz

Arduino Mega 2560 memiliki jumlah pin terbanyak dari semua papan pengembangan Arduino. Mega 2560 memiliki 54 buah digital pin yang dapat digunakan sebagai input atau output, dengan menggunakan fungsi `pinMode()`, `digitalWrite()`, dan `digitalRead()`. Pin-pin tersebut bekerja pada tegangan 5V, dan setiap pin dapat menyediakan atau menerima arus sebesar 20mA, dan memiliki tahanan pull-up sekitar 20-50k Ohm (secara default dalam posisi disconnect). Nilai maximum adalah 40mA, yang sebisa mungkin dihindari untuk menghindari kerusakan *chip* mikrokontroler. Arduino Mega dapat dilihat pada Gambar 2.7



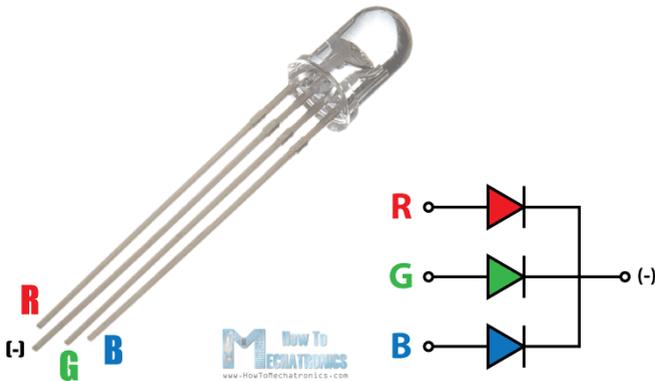
Gambar 2.7 Arduino Mega

2.8 LED RGB [8]

LED RGB adalah LED yang berisikan tiga warna LED yang terintegrasi menjadi satu lampu LED. LED RGB mengandung warna *Red* (merah), *Green* (hijau), dan *Blue* (biru). Ada dua jenis lampu LED RGB. Yang pertama, RGB jenis *flip-flop*, yang artinya lampu LED-nya akan berganti warna secara otomatis. Sedangkan lampu RGB yang kedua yaitu lampu yang dapat dikontrol, lampu ini warnanya dapat diatur sesuai dengan apa yang diinginkan. Jumlah kaki LED RGB ini ada empat kaki. Sementara LED RGB jenis *flip-flop* hanya mempunyai dua kaki saja. Lampu RGB dapat dilihat pada Gambar 2.8

Spesifikasi :

- Lensa : Bening
- Warna yg dipancarkan : Merah / Hijau / Biru
- Common : Katoda
- Ukuran : 5mm
- Tegangan : 2,5~3,0 (V) *forward voltage*
- Sudut Pancaran : 25 derajat
- Intensitas cahaya : 4.000 mcd (*millicandela*)



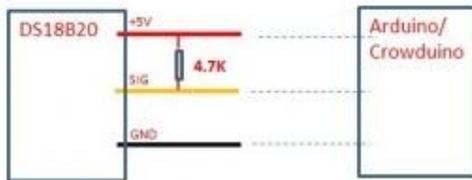
Gambar 2.8 Lampu LED RGB

2.9 Sensor Suhu DS-18B20 [9]

Sensor DS 1820 merupakan sensor suhu 9-12 bit yang memiliki fungsi seperti termometer. Sensor DS1820 memiliki kemampuan untuk mengukur suhu pada kisaran -55°C sampai 125°C dan bekerja secara akurat dengan kesalahan $\pm 0,5^{\circ}\text{C}$ pada kisaran -10°C sampai 85°C . Sensor suhu dapat dilihat pada Gambar 2.9

Spesifikasi:

- Hanya Memerlukan Satu *Port* Pin untuk Komunikasi
- Kemampuan Simplifies Distributed Temperature Sensing Aplikasi
- Tidak memerlukan Komponen Eksternal
- Power Supply berkisar 3,0V sampai 5,5V
- Suhu yang dapat diukur dari -55°C sampai 125°C (-67°F - 257°F)
- Keakuratan data dari -10°C sampai 85°C
- Resolusi termometer 9-Bit
- Kecepatan mengukur suhu dalam 750-800 ms (max)



Gambar 2.9 Sensor Suhu DS-18B20

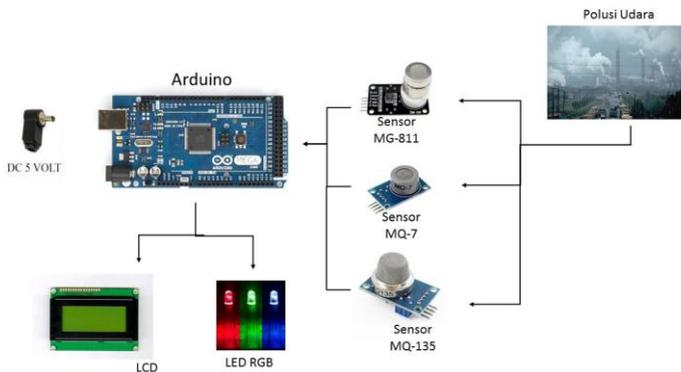
BAB III

PERANCANGAN *HARDWARE* DAN *SOFTWARE*

Pada bab ini akan dijelaskan mengenai perancangan dan pembuatan Alat Pendeteksi Kualitas Udara. Penjelasan diawali dengan penjelasan diagram fungsional sistem secara keseluruhan, kemudian perancangan perangkat keras dan diakhiri perencanaan perangkat lunak. Perancangan perangkat keras menjelaskan proses kerja alat dalam bentuk perancangan *prototype* simulasi alat pendeteksi kualitas udara beserta eletriknya serta perangkat lunak yang mengatur kerja alat.

3.1 Diagram Fungsional Sistem

Perancangan sistem secara keseluruhan pembuatan alat yang digunakan dapat diperlihatkan pada Gambar 3.1.



Gambar 3.1 Diagram Fungsional

Dari Gambar 3.1 dapat dipahami bahwa ketika polusi udara pada daerah tercemar (polusi pabrik misalnya) maka ketiga sensor akan bekerja, sensor yang bekerja meliputi sensor gas O_2 (MQ-135), sensor gas CO_2 (MG-811), dan sensor gas CO (MQ-7). Setelah sensor gas tersebut bekerja, maka data yang diperoleh oleh sensor tersebut akan masuk ke Arduino Mega dan akan ditampilkan pada LCD berupa besaran angka serta LED dengan warna yang berbeda (merah yang berarti bahaya, biru yang berarti hati-hati dan hijau

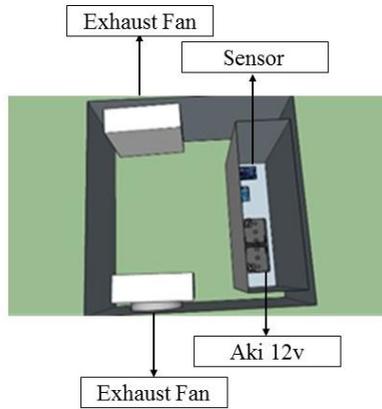
yang berarti aman). Dari data tersebut maka, orang akan mengerti seberapa besar polusi yang terdapat dalam suatu ruangan di pabrik tersebut.

3.2 Perancangan *Hardware*

Pada perancangan *hardware* pada Proyek akhir ini yang dibahas terdiri dari perancangan kotak kontrol, rangkaian sensor gas O₂ (MQ-135), sensor gas CO₂ (MG-811), sensor gas CO (MQ-7), LCD dan juga lampu indikator.

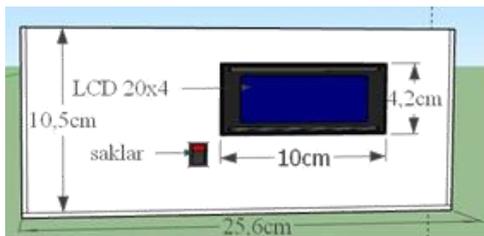
3.2.1. Perancangan Kotak Kontrol

Perancangan kotak kontrol ini sebagai tempat atau wadah dari beberapa rangkaian kontroler yang disusun dalam satu rangkaian, agar *interface* dari rangkaian tersebut terlihat lebih rapi. Kotak kontrol ini dibagi menjadi dua bagian yaitu bagian kontroler dan sensor. Di dalam kotak kontrol terdiri dari Arduino Mega dan *power supply*. Sementara untuk kotak udara terdiri dari ketiga sensor dan *Exhaust Fan*. *Box* ini dirancang berbentuk *box* dengan panjang 28 cm, lebar 20 cm dan panjang 15 cm. *Box* ini sama dengan *box* yang nantinya dibuat kotak udara, hanya perbedaannya ada sekat di pertengahan *box* yang memisahkan kotak kontrol dan kotak udara. Bahan yang digunakan untuk pembuatan *box* ini adalah bahan yang terbuat dari kayu. Pada Gambar 3.2 terlihat rancangan kotak kontrol tampak atas. Kotak kontrol dapat terlihat pula tata letak kotak kontrol. Yang mana paling kiri dari sudut *box* adalah arduino. Kemudian pada sudut kanan *box* terlihat sebuah *accu* 12V sebagai *power supply*. *Accu* digunakan karena sebagai sumber untuk menghidupkan alat ini karena alat ini bersifat *Portable*. Terlihat pula di samping (yang telah diberi sekat) *box* terdapat *exhaust fan* dan juga sensor O₂, CO₂, dan CO. *Exhaust fan* pada *box* kontrol ini akan disambungkan pada kotak udara yang akan diletakkan di sisi kosong *box* kontrol ini, antara *accu* dan pasangan *Arduino*.



Gambar 3.2 Perancangan *Box* Tampak Atas

Pada Gambar 3.3 rancangan *box* tampak depan terlihat sebuah LCD dan LED sebagai indikator bahwa sistem telah menyala dan penampil data dari konsentrasi gas pada udara sekitar yang telah dideteksi oleh sensor. Di dalam Gambar 3.2 juga terdapat I2C LCD dan saklar yang terhubung dengan Arduino pada kotak kontrol yang akan menampilkan kadar gas yang terbaca dari pembacaan sensor gas MQ-7, MG-811, dan MQ-135. Lalu pada bagian atas pada kotak kontrol terdapat kotak sensor yang terdiri dari dua buah *exhaust fan* yang berfungsi untuk menyedot dan membuang udara agar masuk kedalam kotak dan dideteksi oleh ketiga sensor. *Exhaust fan* disini menyedot dan mengeluarkan udara selama 5 detik lalu mati secara bergantian.

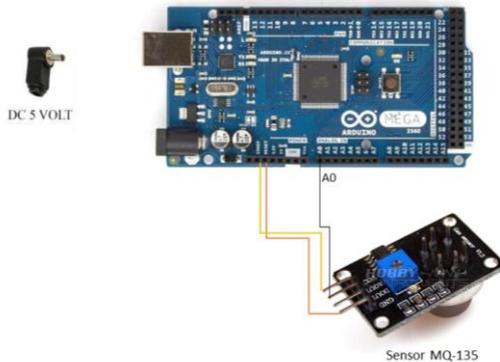


Gambar 3.3 Perancangan *Box* Tampak Depan

Pada Gambar 3.3 terlihat bahwa terdapat LCD dan tombol *on/off* yang akan memudahkan dalam pembacaan nilai PPM dari pembacaan ketiga sensor serta memudahkan dalam menyalakan dan mematikan alat tersebut. Perancangan pembuatan kotak kontrol ini dibuat sedemikian rupa agar berfungsi secara maksimal. Pada pembuatan kotak kontrol ini agar gas yang ada dilingkungan sekitar pabrik/rumah penduduk dapat ditarik ke dalam kotak dan selanjutnya kandungan gas yang terdapat di daerah tersebut dapat diukur.

3.2.2. Perancangan Sensor Gas O₂

Perancangan rangkaian sensor O₂ digunakan sebagai rangkaian untuk mendeteksi kadar gas O₂ pada area yang dideteksi oleh sensor gas MQ-135. Sensor gas ini merupakan sensor untuk mendeteksi gas-gas yang termasuk dalam gas yang menentukan kualitas udara itu sendiri. Salah satunya gas O₂. Kapasitas gas O₂ yang dapat dideteksi yaitu 10-10000 PPM. Keluaran dari sensor gas MQ-135 ini kemudian akan dibaca oleh Arduino Mega.



Gambar 3.4 Rangkaian Sensor O₂

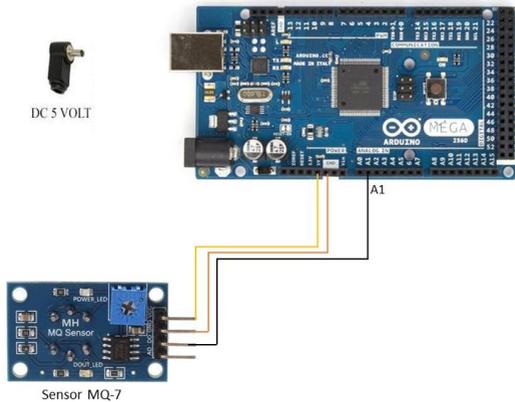
Keterangan rangkaian sensor O₂ dengan Arduino pada Gambar 3.4 dapat dilihat pada Tabel 3.1. Rangkaian pengirim dari *module* O₂ yaitu dilakukan pengiriman dari Arduino ke sensor MQ-135 dengan pin *out* VCC dihubungkan dengan pin 5 Volt Arduino, GND dihubungkan dengan pin GND Arduino, pin Vout dihubungkan dengan pin A0 Arduino.

Tabel 3.1 Pin Pada Sensor O₂

Pin Arduino	Pin sensor
VCC	5 Volt
Aout	A0
Ground	Ground

3.2.3. Perancangan Sensor Gas CO

Perancangan rangkaian sensor gas CO digunakan sebagai rangkaian untuk mengukur mendeteksi kadar gas CO pada area yang akan dideteksi. Sensor gas CO yang digunakan adalah sensor gas MQ-7 dengan menggunakan *modul DT-Sense*. Kapasitas gas CO yang dapat dideteksi yaitu mulai dari 20-2000 PPM. Keluaran dari sensor gas CO ini kemudian dibaca oleh Arduino Mega.



Gambar 3.5 Rangkaian Sensor CO

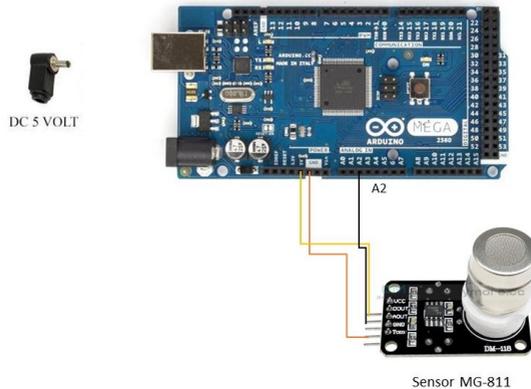
Keterangan rangkaian sensor CO dengan Arduino pada Gambar 3.6 dapat dilihat pada Tabel 3.3. Rangkaian pengirim dari *module* CO yaitu akan dilakukan pengiriman dari Arduino ke sensor MQ-7 dengan pin *out* VCC dihubungkan dengan pin 5 Volt Arduino, GND dihubungkan dengan pin GND Arduino, pin Vout dihubungkan dengan pin A1 Arduino.

Tabel 3.2 Pin Sensor CO

Pin Arduino	Pin sensor
VCC	5 Volt
Aout	A1
Ground	Ground

3.2.4. Perancangan Sensor Gas CO₂

Perancangan rangkaian sensor gas CO₂ digunakan sebagai rangkaian untuk mendeteksi kadar gas CO₂ pada area yang akan dideteksi. Sensor gas CO₂ yang digunakan adalah sensor gas MG-811 dengan menggunakan modul DT-Sense. Kapasitas gas CO₂ yang dapat dideteksi yaitu mulai dari 350-10000 PPM. Keluaran dari sensor gas CO₂ ini kemudian dibaca oleh Arduino Mega.



Gambar 3.6 Rangkaian Sensor CO₂

Keterangan rangkaian sensor CO₂ dengan Arduino pada Gambar 3.5 dapat dilihat pada Tabel 3.2. Rangkaian pengirim dari *module* O₂ yaitu dilakukan pengiriman dari Arduino ke sensor MG-811 dengan pin *out* VCC dihubungkan dengan pin 5 Volt Arduino, GND dihubungkan dengan pin GND Arduino, pin Vout dihubungkan dengan pin A2 Arduino.

Tabel 3.3 Pin Sensor CO₂

Pin Arduino	Pin sensor
VCC	5 Volt
Aout	A3
<i>Ground</i>	<i>Ground</i>

3.3 Perancangan *Software*

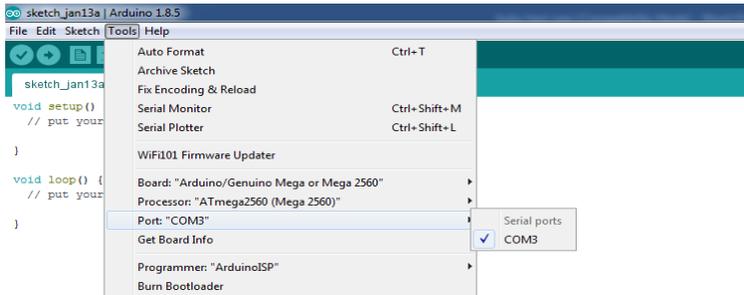
Pada perancangan perangkat *software* pada Proyek akhir ini yang dibahas terdiri dari pemrograman pengukuran ADC dan tegangan pada sensor pada *software* Arduino IDE.

Dalam perancangan program pada *software* arduino dengan fungsi terkait yang dibutuhkan diperlukan beberapa tahapan yang harus dilakukan terlebih dahulu. Tahapan tersebut adalah membuat *flowchart* agar alat lebih sederhana. Setelah tahapan tersebut terselesaikan barulah kita memprogram fungsi terkait yang dikodingkan dalam bahasa C.

Untuk memprogram arduino juga harus dilakukan beberapa tahapan sebagai berikut :

1. **Setting Board Arduino.** Dalam pemrograman *software* Arduino harus di *setting* terlebih dahulu *board* arduino agar penggunaan Arduino cocok. Dalam purwarupa kali ini Arduino menggunakan Arduino AT-Mega 2560. Untuk *setting board* arduino bisa masuk ke *tools – board* – setelah itu pilihlah *board* Arduino yang sesuai.
2. **Setting Serial.** *Serial* ini merupakan kabel Arduino yang dihubungkan kepada komputer atau laptop. *Serial* ini mempunyai dua fungsi yang bisa digunakan. Pertama *serial port* digunakan untuk mengunduh program dari Arduino yang kedua *serial* digunakan sebagai komunikasi *serial* pada Arduino dengan komputer. *Setting serial* bisa masuk *tools – serial* - lalu pilih COM yang sesuai dengan Arduino yang terpasang. Untuk lebih jelasnya dapat dilihat pada Gambar 3.7.
3. Apabila *program* tidak dapat diunduh karena *serial port*, maka cek terlebih dahulu *serial* yang benar pada *device manager*. Lalu dalam *software* Arduino untuk memilih *serial port*-nya samakan dengan *serial port* untuk arduino dalam *device manager* tersebut. Untuk masuk

ke *device manager* dapat masuk *start windows* – lalu ketika *device manager* klik dua kali dan masuk ke COM.



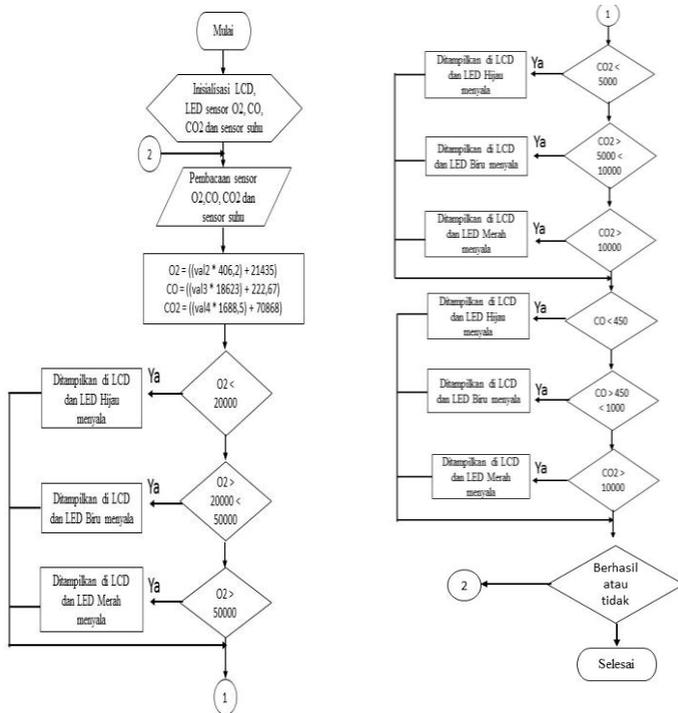
Gambar 3.7 *Setting Serial Port*

Pemrograman *software* Arduino dirancang dengan menggunakan software yang bernama Arduino IDE dengan menggunakan bahasa pemrograman C. Arduino sangatlah berbeda sekali dengan ATmega lainnya. Arduino merupakan sebuah kit mikrokontroler AVR yang dibuat dalam sebuah *board* (papan PCB). Dikembangkan di Italia sejak tahun 2005. Dalam 1 board sudah terdapat mikrokontroler lengkap dengan *pin/port* untuk koneksi serta sudah dilengkapi dengan *downloader*. Dalam segi bahasa pemrograman, Arduino memiliki bahasa pemrograman yang lebih mudah dan sederhana terutama bagi pemula.

Alasan bahasa pemrograman Arduino lebih mudah dan sederhana adalah karena didalam Arduino sudah terdapat beberapa

library yang dapat digunakan untuk merancang pemrograman yang diinginkan.

Berikut ini merupakan *flowchart* yang digunakan pada Proyek Akhir ini.



Gambar 3.8 Flowchart Rangkaian Arduino

Pada Gambar 3.8 dijelaskan bahwa pada saat alat menyala, maka sensor akan membaca nilai PPM menggunakan rumus yang terdapat pada *slope* dan *intercept*. Setelah pembacaan sensor maka nilai yang keluar akan ditampilkan pada LCD dan LED RGB, dimana ada lampu berwarna hijau, biru dan merah yang menyala sesuai dengan nilai sensor tersebut.

BAB IV

PENGUJIAN DAN ANALISA

Dalam membuat suatu sistem, pengujian dan analisa sangat diperlukan. Pengujian dan analisa bertujuan untuk mengetahui sistem bekerja sesuai rencana atau belum. Dari hasil pengujian dan analisa dapat diketahui kelemahan-kelemahan dari sistem, sehingga dapat dilakukan perbaikan, pengembangan, dan penyempurnaan sistem.

Dalam bab ini dibahas pengujian serta analisa dari perancangan sistem yang telah dibuat pada Proyek akhir. Pengujian dilakukan secara parsial terlebih dahulu, tujuannya adalah untuk mengetahui kinerja *hardware* setiap unit. Setelah mengetahui respon kinerja *hardware* setiap unit, kemudian akan dilakukan pengujian integrasi sistem secara keseluruhan. Adapun beberapa pengujian yang telah dilakukan adalah sebagai berikut.

4.1 Pengujian Sensor Karbon Dioksida (CO₂)

Pengujian sensor CO₂ ini bertujuan untuk mengetahui apakah sensor sudah dapat bekerja dengan baik dan benar dalam mendeteksi gas CO₂. Sensor CO₂ yang digunakan pada Proyek Akhir ini yaitu MG8-11. Sensor MG8-11 memiliki kepekaan yang baik terhadap gas karbon dioksida. Sensor MG-811 akan disambungkan dengan modul DT-Sense gas sensor sebagai *driver* sensor MG-811.

Agar sensor mendeteksi sesuai yang kita inginkan, Arduino akan diberi sebuah program yang telah dibuat di *software* IDE seperti pada Gambar 4.1

```

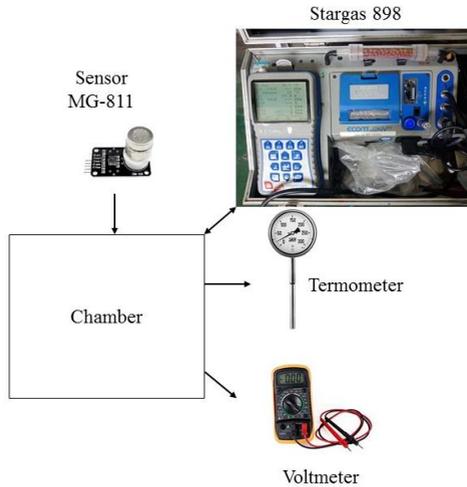
int sensorValue4 = analogRead(A4);
Serial.print("CO2 = ");
Serial.print(sensorValue4);
Serial.println();
lcd.clear();
lcd.setCursor(10,0);
lcd.print("");
lcd.print(sensorValue4);
float voltage4 = sensorValue4 *5/1023.0;
lcd.setCursor(10,1);
lcd.print("\tvv:");
lcd.print(voltage4, 2);
lcd.println();
Serial.print("\tvv:");
Serial.print(voltage4, 3);
Serial.println();
delay(2000);

```

Gambar 4.1 Program Sensor CO₂

Setelah pemrograman sensor CO₂ selesai, maka sensor dapat digunakan. Untuk menguji apakah program yang ada pada sensor dapat bekerja, setelah itu kami mencoba mengambil data dari asap kendaraan bermotor seperti yang dilakukan saat kalibrasi sensor, yaitu memasukkan asap kendaraan bermotor di tempat tertutup yang sudah terdapat sensor CO₂ di dalamnya.

Sebelum pengambilan data dilakukan, maka diperlukan kalibrasi sensor terlebih dahulu. Kalibrasi merupakan suatu proses untuk mengetahui apakah sensor tersebut masih linear atau tidak. Sensor yang masih linear akan memberikan hasil yang linear pula. Ketika kalibrasi diperlukam rangkaian pengujian dan juga alat untuk pengujian. Rangkaian pengujian untuk kalibrasi sensor CO₂ seperti pada Gambar 4.2



Gambar 4.2 Rangkaian Pengujian Sensor CO₂

Dari rangkaian tersebut akan didapatkan data kalibrasi menggunakan alat ukur sensor. Alat ukur sensor ini berguna untuk mendeteksi kandungan gas CO₂. Untuk percobaan kali ini, kami menggunakan sepeda motor berbahan bakar pertamax. Pada data hasil kalibrasi gas CO₂ didapatkan hasil seperti Tabel 4.1

Tabel 4.1 Hasil Kalibrasi Sensor CO₂

No	Temperatur (°C)	Tegangan (Volt)	Ecom (PPM)	LCD (PPM)
1	30,7	0,274	15000	10923,2
2	30,5	0,274	16000	15817,4
3	30,7	0,249	18000	17448,8
4	30,3	0,274	19000	23974,4
5	30,1	0,274	20000	25605,8
6	30,2	0,298	29000	30500
7	29,9	0,23	31000	32131,4
8	30,0	0,254	37000	35394,2
9	30,1	0,274	41000	40288,4
10	29,9	0,393	48000	43551,2

Pada hasil data kalibrasi tersebut didapatkan dengan sumber emisis yang berasal dari kendaraan sepeda motor yang berbahan bakar pertamax dan dari pengkalibrasian didapatkan hasil temperatur, output tegangan, PPM pada *ECOM* dan PPM pada LCD.

Pada Tabel 4.1 dijelaskan bahwa didapatkan hasil dari konsentrasi yang tampil pada LCD dan konsentrasi yang tampil pada alat ukur *ECOM*, pembacaan sensor MG-811 dalam mendeteksi gas CO₂ masih dalam bentuk %. Maka dari itu, perlu adanya rumus konversi untuk merubah data yang masih dalam bentuk % ke bentuk satuan gas CO₂ yaitu PPM. Untuk mendapatkan rumus konversi tersebut, yaitu dengan memasukkan data % yang telah dikalibrasi ke *Microsoft excel*. Setelah itu, masukkan rumus *Slope* dan *Intercept*. *Slope* secara sistematis merupakan ukuran kemiringan dari suatu garis atau suatu nilai yang menunjukkan seberapa besar kontribusi yang diberikan suatu variabel X terhadap variabel Y. Sementara *Intercept* merupakan suatu titik perpotongan antara suatu garis dengan sumbu Y. Pada Tabel 4.1 didapat hasil *intercept* sebesar 16688,5 sedangkan hasil dari *slope* sebesar 70868.

Kemudian dari hasil *slope* dan *intercept* yang terdapat pada Tabel 4.1 akan dimasukkan kedalam program Arduino sebagai rumus konversi seperti pada Gambar 4.3

```
int val4 = analogRead(A2);
CO2 = ((val4 * 1688,5) + 70868);
int sensorVal4 = val4;
//mb.Ireg(sencalc4, sensorVal4);
Serial.print("CO2 :");
Serial.print(CO2);
Serial.print(" % ");
Serial.println(" ");
```

Gambar 4.3 Program Konversi Sensor CO₂

Setelah dikonversi dengan rumus yang ada pada program seperti Gambar 4.3 maka pembacaan yang semula dalam bentuk ADC sudah berubah dalam bentuk persentase. Hasil pembacaan dapat dilihat pada Tabel 4.3 dengan data tersebut dapat dibandingkan dengan hasil pengukuran yang terdapat di alat ukur CO₂..

Tabel 4.2 Hasil Kalibrasi Sensor CO₂

No. Pengukuran	Pembacaan Pada <i>Ecom</i> (PPM)	Pembacaan Pada LCD (PPM)	% <i>Error</i>
1	15000	10923,2	0,271786667
2	16000	15817,4	0,0114125
3	18000	17448,8	0,030622222
4	19000	23974,4	0,261810526
5	20000	25605,8	0,28029
6	29000	30500	0,051724138
7	31000	32131,4	0,036496774
8	37000	35394,2	0,0434
9	41000	40288,4	0,017356098
10	48000	43551,2	0,092683333

Dari perhitungan konversi O₂ serta *slope* dan *intercept* didapatkan kesalahan (*Error*) yang sudah di paparkan pada Tabel 4.2. Untuk mendapatkan hasil % *error* pada Tabel 4.2 didapatkan dari rumus yang terdapat pada *Microsoft Excel* =ABS(nilai pada *Ecom* – nilai pada LCD) : nilai pada *Ecom*. Contoh : =ABS(15000-10923,2)/15000.

4.2 Pengujian Sensor Oksigen (O₂)

Pengujian sensor O₂ ini bertujuan mengetahui apakah sensor sudah dapat bekerja dengan baik dan benar dalam mendeteksi gas O₂. Sensor O₂ yang digunakan pada Proyek Akhir ini yaitu sensor MQ-135. Sensor MQ-135 memiliki kepekaan terhadap gas-gas yang menentukan kualitas udara, salah satunya adalah gas O₂. Agar sensor dapat mendeteksi data yang sesuai dengan yang kita inginkan, Arduino akan diberi sebuah program yang telah dibuat di software IDE seperti pada Gambar 4.2.

```

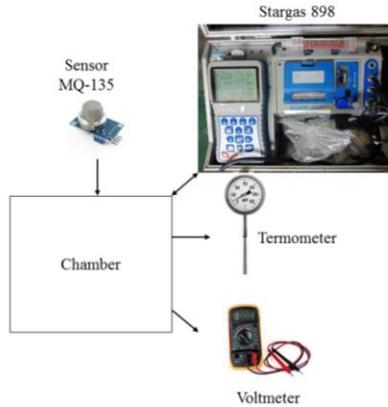
int sensorValue1 = analogRead(A1);
Serial.print("O2 = ");
Serial.print(sensorValue1);
Serial.println();
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Gas SO :");
lcd.print(sensorValue1);
float voltage1 = sensorValue1*5/1023.0;
lcd.setCursor(8,0);
lcd.print("\tv:");
lcd.print(voltage1, 3);
lcd.println();
Serial.print("\tv:");
Serial.print(voltage1, 3);
Serial.println();
delay(1000);

```

Gambar 4.4 Program Sensor O₂

Setelah pemrograman sensor O₂ selesai, maka sensor dapat digunakan. Untuk menguji apakah program yang ada pada sensor dapat bekerja, kami mencoba mengambil data dari asap kendaraan bermotor seperti yang dilakukan saat kalibrasi sensor, yaitu memasukkan asap kendaraan bermotor di tempat tertutup yang sudah terdapat sensor O₂ di dalamnya.

Sebelum pengambilan data dilakukan, maka diperlukan kalibrasi sensor terlebih dahulu. Kalibrasi merupakan suatu proses untuk mengetahui apakah sensor tersebut masih linear atau tidak. Sensor yang masih linear akan memberikan hasil yang linear pula. Ketika kalibrasi diperlukam rangkaian pengujian dan juga alat untuk pengujian. Rangkaian pengujian untuk kalibrasi sensor O₂ seperti pada Gambar 4.5



Gambar 4.5 Rangkaian Pengujian Sensor O₂

Dari rangkaian tersebut akan didapatkan data kalibrasi menggunakan alat ukur sensor. Alat ukur sensor ini berguna untuk mendeteksi kandungan gas O₂. Untuk percobaan pertama, kami menggunakan sepeda motor berbahan bakar pertamax. Pada hasil kalibrasi gas O₂ didapatkan hasil seperti Tabel 4.4.

Tabel 4.3 Hasil Kalibrasi Sensor O₂

No.	Temperatur (°C)	Tegangan (Volt)	LCD (PPM)	<i>Ecom</i> (PPM)
1	30,7	1,69	151012,8	148000
2	30,5	1,803	161980,2	157000
3	30,7	1,901	165636	162000
4	30,3	1,971	166854,6	170000
5	30,1	1,867	168073,2	173000
6	30,2	1,735	176197,2	179000
7	29,9	2,101	177415,8	184000
8	30,0	2,00	186352,2	186000
9	30,1	1,75	187977	189000
10	29,9	1,714	196507,2	190000

Pada hasil data kalibrasi tersebut didapatkan dengan sumber emisis yang berasal dari kendaraan sepeda motor yang berbahan bakar pertamax dan dari pengkalibrasian didapatkan hasil temperatur dan output tegangan. Pengujian yang berikutnya yaitu dengan menguji nilai tampilan LCD dan PPM dari alat ukur *Ecom*. Data yang didapatkan akan ditampilkan pada Tabel 4.4

Pada Tabel 4.4 dijelaskan bahwa didapatkan hasil dari konsentrasi yang tampil pada LCD dan konsentrasi yang tampil pada alat ukur ECOM, sensor MQ-135 masih membaca data dalam bentuk %. Oleh karena itu harus ada rumus konversi data dari bentuk % ke bentuk satuan satuan gas *standart* atau PPM (*Part Per Million*). Untuk mendapatkan rumus konversi tersebut dengan memasukkan data yang telah terkalibrasi ke *Microsoft Excel*. Setelah itu, masukkan rumus *Slope* dan *Intercept*. *Slope* secara sistematis merupakan ukuran kemiringan dari suatu garis atau suatu nilai yang menunjukkan seberapa besar kontribusi yang diberikan suatu variabel X terhadap variabel Y. Sementara *Intercept* merupakan suatu titik perpotongan antara suatu garis dengan sumbu Y. Pada Tabel 4.2 didapat hasil *intercept* sebesar 406,2 sedangkan hasil dari *slope* sebesar 21435. Dari Tabel 4.4 didapatkan data saat kenaikan tem-peratur, maka konsentrasi PPM juga semakin naik.

Kemudian dari hasil slope dan *intercept* yang terdapat pada Tabel 4.4 akan dimasukkan kedalam program Arduino sebagai rumus konversi seperti pada Gambar 4.6

```
void loop() {  
  // put your main code here, to run repeatedly:  
  int val2 = analogRead(A0);  
  O2 = ((val2 * 406,2) + 21435);  
  int sensorVal2 = val2;  
  //mb.Ireg(sencalc2, sensorVal2);  
  Serial.print("O2 :");  
  Serial.print(O2);  
  Serial.print(" % ");  
  Serial.println(" ");  
}
```

Gambar 4.6 Program Konversi pada Arduino

Setelah dikonversi dengan rumus yang ada pada program seperti Gambar 4.6. Hasil pembacaan dapat dilihat pada Tabel 4.3. dengan data tersebut dapat dibandingkan dengan hasil pengukuran yang terdapat di alat ukur O₂.

Tabel 4.4 Hasil Konversi Sensor O₂

No. Pengukuran	Pembacaan Pada <i>Ecom</i> (PPM)	Pembacaan Pada LCD (PPM)	% <i>Error</i>
1	148000	151012,8	0,020356757
2	157000	161980,2	0,031721019
3	162000	165636	0,022444444
4	170000	166854,6	0,018502353
5	173000	168073,2	0,028478613
6	179000	176197,2	0,015658101
7	184000	177415,8	0,035783696
8	186000	186352,2	0,001893548
9	189000	187977	0,005412698
10	190000	196507,2	0,034248421

Dari perhitungan konversi CO₂ serta *slope* dan *intercept* didapatkan kesalahan (*error*) yang sudah di paparkan pada Tabel 4.4. Untuk mendapatkan hasil % *error* pada Tabel 4.4 didapatkan dari rumus yang terdapat pada *Microsoft Excel* =ABS(nilai pada *Ecom* – nilai pada LCD) : nilai pada *Ecom*. Contoh : =ABS(148000 - 151012,8)/ 148000

4.3 Pengujian Sensor Karbon Monoksida (CO)

Pengujian sensor CO ini bertujuan untuk mengetahui apakah sensor sudah dapat bekerja dengan baik dan benar dalam mendeteksi gas CO. Sensor CO yang digunakan pada Proyek Akhir ini yaitu MQ-7. Sensor MQ-7 memiliki kepekaan yang baik terhadap gas Karbon Monoksida.

Agar sensor mendeteksi sesuai yang kita inginkan, arduino akan diberi sebuah program yang telah dibuat di *software* IDE seperti Gambar 4.7

```

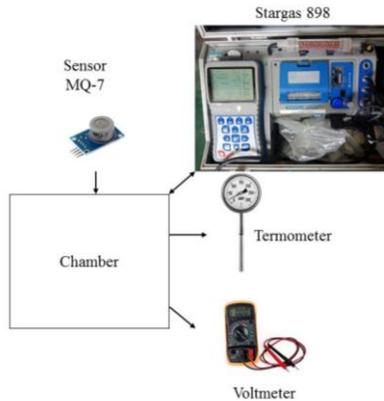
int sensorValue3 = analogRead(A3);
Serial.print("CO = ");
Serial.print(sensorValue3);
Serial.println();
lcd.clear();
lcd.setCursor(7, 0);
lcd.print("");
lcd.print(sensorValue3);
float voltage3 = sensorValue3 * 5/1023.0;
lcd.setCursor(7, 1);
lcd.print("\tv:");
lcd.print(voltage3, 2);
lcd.println();
Serial.print("\tv:");
Serial.print(voltage3, 3);
Serial.println();
delay(1000);

```

Gambar 4.7 Program Gas CO

Setelah pemrograman sensor CO selesai, maka sensor dapat digunakan. Untuk menguji apakah program yang ada pada sensor dapat bekerja, kelompok kami mencoba mengambil data dari asap kendaraan bermotor seperti yang dilakukan saat kalibrasi sensor, yaitu memasukkan asap kendaraan bermotor di tempat tertutup yang sudah terdapat sensor CO di dalamnya.

Sebelum pengambilan data dilakukan, maka diperlukan kalibrasi sensor terlebih dahulu. Kalibrasi merupakan suatu proses untuk mengetahui apakah sensor tersebut masih linear atau tidak. Sensor yang masih linear akan memberikan hasil yang linear pula. Ketika kalibrasi diperlukam rangkaian pengujian dan juga alat untuk pengujian. Rangkaian pengujian untuk kalibrasi sensor CO seperti pada Gambar 4.8



Gambar 4.8 Rangkaian Pengujian Sensor CO

Dari rangkaian tersebut akan didapatkan data kalibrasi menggunakan alat ukur sensor. Alat ukur sensor ini berguna untuk mendeteksi kandungan gas CO. Untuk percobaan kali ini, kami menggunakan sepeda motor berbahan bakar pertamax. Pada data hasil kalibrasi gas CO didapatkan hasil seperti Tabel 4.5

Tabel 4.5 Hasil Kalibrasi Sensor CO

No.	Temperatur (°C)	Tegangan (Volt)	<i>Ecom</i>	LCD
1	30,7	0,63	1756	2252,577
2	30,5	0,63	1906	2271,2
3	30,7	0,557	1954	2289,823
4	30,3	0,713	2269	2438,807
5	30,1	0,621	2945	2550,545
6	30,2	0,513	2947	2587,791
7	29,9	0,797	3135	2625,037
8	30,0	0,572	3303	3053,366
9	30,1	0,626	3511	3258,219
10	29,9	0,458	3940	4338,353

Pada hasil data kalibrasi tersebut didapatkan dengan sumber emisii yang berasal dari kendaraan sepeda motor yang berbahan bakar pertamax dan dari pengkalibrasian didapatkan hasil temperatur dan output tegangan. Pengujian yang berikutnya yaitu dengan menguji nilai tampilan LCD dan PPM dari alat ukur *Ecom*. Data yang didapatkan akan ditampilkan pada Tabel 4.5

Pada Tabel 4.5 dijelaskan bahwa didapatkan hasil dari konsentrasi yang tampil pada LCD dan konsentrasi yang tampil pada alat ukur *ECOM*, Dalam mendeteksi gas, diperlukan rumus konversi untuk merubah data yang masih dalam bentuk % ke bentuk PPM Untuk mendapatkan rumus konversi tersebut dengan memasukkan data yang telah terkalibrasi ke *Miscrosoft Excel. Slope* secara sistematis merupakan ukuran kemiringan dari suatu garis atau suatu nilai yang menunjukkan seberapa besar kontribusinya yang diberikan suatu variabel X terhadap variabel Y. Sementara *Intercept*

merupakan suatu titik perpotongan antara suatu garis dengan sumbu Y. Setelah itu, masukkan rumus *Slope* dan *Intercept*. Pada Tabel 4.6 didapat hasil *intercept* sebesar 18623 sedangkan hasil dari *slope* sebesar 222,67. Kemudian hasil dari *slope* dan *intercept* dimasukkan kedalam program Arduino sebagai rumus konversi seperti pada Gambar 4.9

```
int val3 = analogRead(A1);
CO = ((val3 * 18623) + 222,67);
int sensorVal3 = val3;
//mb.Ireg(sencalc3, sensorVal3);
Serial.print("CO :");
Serial.print(CO);
Serial.print(" % ");
Serial.println(" ");
```

Gambar 4.9 Program Konversi Sensor CO

Setelah dikonversi dengan rumus yang ada pada program seperti Gambar 4.9 maka pembacaan yang semula dalam bentuk % sudah berubah dalam bentuk PPM. Hasil pembacaan dapat dilihat pada Tabel 4.7 dengan data tersebut dapat dibandingkan dengan hasil pengukuran yang terdapat di alat ukur CO

Tabel 4.6 Hasil Konversi Sensor CO

No. Pengukuran	Pembacaan Pada <i>Ecom</i> (PPM)	Pembacaan Pada LCD (PPM)	% <i>Error</i>
1	1756	2252,577	0,282788724
2	1906	2271,2	0,191605456
3	1954	2289,823	0,171864381
4	2269	2438,807	0,074837814
5	2945	2550,545	0,133940577
6	2947	2587,791	0,121889718
7	3135	2625,037	0,162667624
8	3303	3053,366	0,075577959
9	3511	3258,219	0,071996867
10	3940	4338,353	0,101104822

Dari perhitungan konversi CO₂ serta *slope* dan *intercept* didapatkan kesalahan (*error*) yang sudah di paparkan pada Tabel 4.6. Untuk mendapatkan hasil % *error* pada Tabel 4.6 didapatkan dari rumus yang terdapat pada *Microsoft Excel* =ABS(nilai pada *Ecom* – nilai pada LCD) : nilai pada *Ecom*. Contoh : =ABS(1756-2252,577)/1756

4.4 Pengujian Sensor Suhu DS-18B20

Pengujian pada sensor suhu DS-18B20 bertujuan untuk mengetahui apakah sensor sudah dapat bekerja dengan baik dan benar dalam mendeteksi suhu yang terdapat pada alat tersebut. Pertama-tama terdapat program untuk mengetahui apakah sensor tersebut dapat bekerja atau tidak. Program tersebut dapat di lihat pada Gambar 4.10

```
Serial.print("Requesting temperatures...");  
sensors.requestTemperatures();  
Serial.println("DONE");  
Serial.print("Temperature for the device 1 (index 0) is: ");  
Serial.println(sensors.getTempCByIndex(0));
```

Gambar 4.10 Program Sensor Temperatur

Setelah memakai program pada sensor tersebut, maka hal pertama yang disiapkan adalah termometer serta kompor dan air mendidih.



Gambar 4.11 Kalibrasi Sensor Suhu

Pada Gambar 4.11 terdapat sensor suhu yang di letakkan pada wadah yang dipanaskan terlebih dahulu lalu diukur suhu dari sensor DS-18B20 dengan alat ukur termometer.

Tabel 4.7 Hasil Kalibrasi Sensor Suhu

No.	Termometer (°C)	DS-18B20 (°C)
1	50	54
2	60	63
3	70	69
4	80	81
5	90	94

Pada Tabel 4.7 merupakan percobaan menggunakan air yang dipanaskan terlebih dahulu menggunakan kompor, lalu ditunggu hingga nilai di termometer akan mencapai nilai yang diinginkan dan dilihat apakah nilai yang terdapat pada sensor DS-18B20 sesuai dengan nilai yang terdapat pada nilai di termometer. Setelah itu akan dihitung seberapa % *Error* yang terdapat pada sensor tersebut yang dapat dilihat pada Tabel 4.8

Tabel 4.8 Persentase *Error* Pada Sensor

No.	Termometer (°C)	DS-18B20 (°C)	<i>Error</i>
1	50	54	0,074074074
2	60	63	0,047619048
3	70	69	0,014492754
4	80	81	0,012345679
5	90	94	0,042553191

Pada Tabel 4.9 dapat dilihat bahwa persentase *Error* yang terdapat pada sensor suhu tidak lebih dari 0,07 %.

4.5 Pengujian Keseluruhan

Setelah sensor sudah dikalibrasi dan mendapatkan perhitungan konversi dari kalibrasi sebagai keterangan dalam pembacaan sensor, maka perlu dilakukan pengujian dan pengambilan data pada alat tersebut. Pada alat ini dilakukan 1 pengambilan data, yaitu di daerah parkir sepeda motor vokasi ITS.

Percobaan pertama dilakukan pada parkir sepeda motor vokasi ITS pada tanggal 22 Januari 2019 yang dilakukan selama 30

menit pada pukul 17.00 sampai dengan 17.30 dengan objek asap kendaraan bermotor. Hasil pengujian dapat dilihat pada Tabel 4.9

Tabel 4.9 Hasil Pengujian di Area Parkiran Sepeda Motor

No.	PPM O ₂	PPM CO ₂	PPM CO	Temperatur (°C)
1	43500	85100	440	33
2	45700	86000	470	33
3	44400	86100	455	34
5	44700	86700	441	33
6	44000	84300	439	33
7	44000	84500	443	34
8	43900	87200	434	35
9	44300	84000	441	33
10	44100	84400	471	34
11	45900	88000	465	33
12	45000	87200	450	33
13	43900	86000	439	33
14	46500	87900	455	34
15	47400	86900	450	34

Dari hasil yang di dapat pada Tabel 4.8 gas O₂, CO₂ dan gas CO yang terdapat di area parkir dengan kendaraan sepeda motor dengan bahan bakar pertalite. Gas yang terbaca dapat dikategorikan gas yang aman, namun apabila dalam jangka yang lama dalam hitungan berjam-jam maka dapat mengakibatkan iritasi tenggorokan, batuk hingga pingsan karena kandungan CO₂ yang terlalu banyak dan kandungan O₂ yang terlalu sedikit. Pengambilan data diambil pada pagi hari menggunakan kendaraan bermotor yang dapat dilihat pada Gambar 4.12



Gambar 4.12 Pengambilan Data Pada Alat

Pada Gambar 4.12 terlihat alat yang diletakkan di depan kendaraan agar asap pada kendaraan tersebut diserap pada alat dan akan dibaca kandungannya pada alat tersebut.

-----Halaman ini sengaja dikosongkan-----

BAB V

PENUTUP

Dari hasil yang telah didapatkan selama proses pembuatan serta proses analisa data untuk Proyek Akhir ini, maka dapat diambil beberapa kesimpulan dan saran yang berguna untuk perbaikan dan pengembangan agar nantinya bisa bermanfaat.

5.1 Kesimpulan

Dari hasil penelitian dan pembahasan mengenai rancang alat kualitas udara dapat disimpulkan bahwa:

1. Dari hasil uji coba, pembacaan pada setiap kalibrasi akan berbeda-beda (data naik-turun) sehingga perlu dilakukan kalibrasi hingga mendapatkan data yang bagus.
2. Dari hasil pengujian, pada sensor O₂ persentase *Error* terbesar yaitu 0,035, pada sensor CO₂ persentase *Error* terbesar yaitu 0,27, pada sensor CO persentase *Error* terbesar yaitu 0,28.
3. Data dari kalibrasi didapatkan apabila *Slope* dan *Intercept* sudah dimasukkan ke program IDE di Arduino.
4. Pada saat sensor CO belum diberi gas, maka yang akan keluar di LCD adalah hasil (-) dikarenakan penhitungan saat kalibrasi tidak sesuai dengan hasil yang diharapkan (kurang tepat)

5.2 Saran

Untuk lebih memperbaiki dan menyempurnakan kinerja dari alat ini, maka perlu disarankan :

Di ITS perlu adanya tempat kalibrasi untuk beberapa gas agar dapat memudahkan dalam mengkalibrasi sensor.

DAFTAR PUSTAKA

- [1] Abdul, Kadir, “*Panduan Praktis Mempelajari Aplikasi Mikrokontroler Dan Pemrogramannya Menggunakan Arduino*”, Andi Offset, Jogjakarta , 2013
- [2] Agung Bairuni Wardi, “Rancang Bangun Sistem Alat Pendeteksi Gas CO, dan CO₂ Sebagai Informasi Pencemaran Udara Menggunakan Arduino dan Ethernet”, *Tugas Akhir Program Studi D3 Teknik Elektro FTI-ITS*, 2012
- [3] Dickry Junior Triandy, “Rancang Bangun Alat Pendeteksi Gas Beracun Pada Area Aktivitas Gunung Berapi”, *Tugas Akhir Program Studi D3 Teknik Elektro FTI-ITS*, 2015
- [4] Didin Jalanudin “*Sensor Gas CO, CO₂ dan O₂*” Detik.com, <http://detik.com/macam-macam/sensor/udara/> 15 April 2018
- [5] Mochamad, Fajar, “*Mudah Belajar Mikrokontroler Arduino*”, Penerbit Informatika, Bandung, 2017
- [6] Muhammad Amnuddin “*LCD*” detik.com, <http://detik.com/apa/itu/lcd/> 15 April 2018
- [7] Nila Kusuma “*Belasan Warga Karawang Keracunan Akibat Keracunan Gas*” Kompas, <http://kompas.com/belasan/warga/keracunan/gas/beracun/> 1 Januari 2019
- [8] Sri Suryaningsih, dkk, “Rancang Bangun Alat Ukur Kadar Gas Co Berbasis Nirkabel Rf Untuk Pemantauan Kondisi Pencemaran Udara”, *Essay*, Departemen Fisika FMIPA Universitas Padjadjaran, 2015
- [9] Wicaksono “*Penjelasan Mengenai Kualitas Udara*” Kompas, <http://kompas.com/penjelasan/tentang/kualitas/udara/> 15 April 2018
- [10] Wisnu Prasetyo Wicaksana, dkk, “Simulasi Detektor Co Pada Kabin Untuk Mencegah Keracunan Penumpang Akibat Peningkatan Kadar Co Yang Tinggi”, *Essay*, STMIK AMIKOM Yogyakarta, 2015

LAMPIRAN A

A. Program pada Arduino

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <LCD.h>
#include <SPI.h>
#include <OneWire.h>
#include <DallasTemperature.h>

#define ONE_WIRE_BUS 2

OneWire oneWire(ONE_WIRE_BUS);

DallasTemperature sensorSuhu(&oneWire);

LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7);
//const int sencalc = 100;
//const int sencalc1 = 110;
//const int sencalc2 = 120;
//const int sencalc3 = 130;
//const int sencalc4 = 140;

long ts;

int Calc;

double CO2;
double CO;
double O2;
float suhuSekarang;
void setup() {
  lcd.begin(20, 4);
  lcd.setBacklightPin(3, POSITIVE);
  lcd.setBacklight(HIGH);
  // put your setup code here, to run once:
  Serial.begin(9600);
  digitalWrite(31, OUTPUT);
```

```

digitalWrite(33, OUTPUT);
digitalWrite(35, OUTPUT);
digitalWrite(37, OUTPUT);
digitalWrite(39, OUTPUT);
digitalWrite(41, OUTPUT);
digitalWrite(43, OUTPUT);
digitalWrite(45, OUTPUT);
digitalWrite(47, OUTPUT);

//Serial.begin(9600);
//Serial.println("Dallas Temperature IC Control Library
Demo");

    sensorSuhu.begin();
}

void loop() {
    suhuSekarang = ambilSuhu();
    Serial.println(suhuSekarang);
    // put your main code here, to run repeatedly:
    int val2 = analogRead(A0);
    O2 = ((val2 * 406.2) + 21435);
    int sensorVal2 = val2;
    //mb.Ireg(sencalc2, sensorVal2);
    Serial.print("O2 :");
    Serial.print(O2);
    Serial.print(" % ");
    Serial.println(" ");

    //Serial.print("Requesting temperatures...");
    //sensors.requestTemperatures();
    //Serial.println("DONE");
    //Serial.print("Temperature for the device 1 (index 0) is: ");
    //Serial.println(sensors.getTempCByIndex(0));

    int val3 = analogRead(A1);
    CO = ((val3 * 18623) + 222.67);
    int sensorVal3 = val3;
    //mb.Ireg(sencalc3, sensorVal3);

```

```

Serial.print("CO :");
Serial.print(CO);
Serial.print(" % ");
Serial.println(" ");

int val4 = analogRead(A2);
CO2 = ((val4 * 1688.5) + 70868);
int sensorVal4 = val4;
//mb.Ireg(sencalc4, sensorVal4);
Serial.print("CO2 :");
Serial.print(CO2);
Serial.print(" % ");
Serial.println(" ");

if (O2 <= 200000) {
    digitalWrite(31, HIGH);
    digitalWrite(33, LOW);
    digitalWrite(35, LOW);
}
if (O2 >= 200000 <= 500000) {
    digitalWrite(31, LOW);
    digitalWrite(33, HIGH);
    digitalWrite(35, LOW);
}
if (O2 >= 500000) {
    digitalWrite(31, LOW);
    digitalWrite(33, LOW);
    digitalWrite(35, HIGH);
}
if (CO <= 450) {
    digitalWrite(37, HIGH);
    digitalWrite(39, LOW);
    digitalWrite(41, LOW);
}
if (CO >= 450 <= 1000) {
    digitalWrite(37, LOW);
    digitalWrite(39, HIGH);
    digitalWrite(41, LOW);
}
}

```

```

if (CO >= 1000) {
    digitalWrite(37, LOW);
    digitalWrite(39, LOW);
    digitalWrite(41, HIGH);
}
if (CO2 <= 500000) {
    digitalWrite(43, HIGH);
    digitalWrite(45, LOW);
    digitalWrite(47, LOW);
}
if (CO2 >= 500000 <= 1000000) {
    digitalWrite(43, LOW);
    digitalWrite(45, HIGH);
    digitalWrite(47, LOW);
}
if (CO2 >= 1000000) {
    digitalWrite(43, LOW);
    digitalWrite(45, LOW);
    digitalWrite(47, HIGH);
}
//sei();
lcd.setCursor(0, 0);
lcd.print("O2 :");
lcd.setCursor(5, 0);
lcd.print(O2);
lcd.print("PPM");

lcd.setCursor(0, 1);
lcd.print("CO :");
lcd.setCursor(5, 1);
lcd.print(CO);
lcd.print("PPM");

lcd.setCursor(0, 2);
lcd.print("CO2:");
lcd.setCursor(5, 2);
lcd.print(CO2);
lcd.print("PPM");

```

```
lcd.setCursor(0, 3);  
lcd.print("temp:");  
lcd.setCursor(5, 3);  
lcd.print(suhuSekarang);  
lcd.print(" ");  
delay(500);  
}  
float ambilSuhu()  
{  
  sensorSuhu.requestTemperatures();  
  float suhu = sensorSuhu.getTempCByIndex(0);  
  return suhu;  
}
```


LAMPIRAN B

Applications include thermostatic controls, industrial systems, consumer products, thermometers, or any thermally sensitive system

DESCRIPTION

PRELIMINARY



DS18B20 Programmable Resolution 1-Wire® Digital Thermometer

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

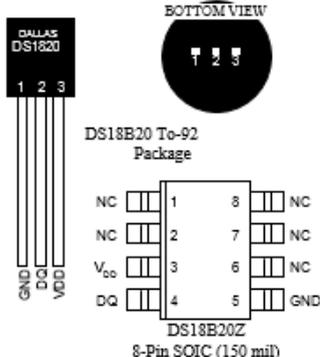
Converts 12-bit temperature to digital word in 750 ms (max.)

User-definable, nonvolatile temperature alarm settings

Alarm search command identifies and addresses devices whose temperature is outside of programmed limits (temperature alarm condition)

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

PIN ASSIGNMENT



PIN DESCRIPTION

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

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.

050400

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.

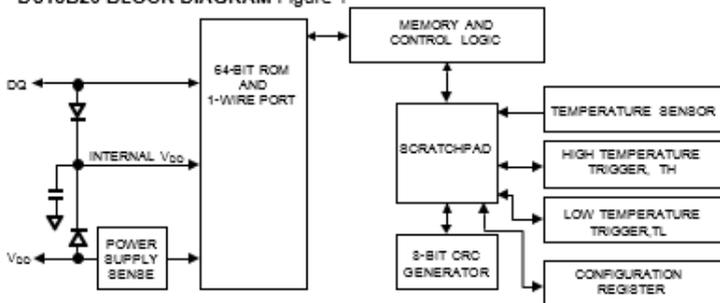
OVERVIEW

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.

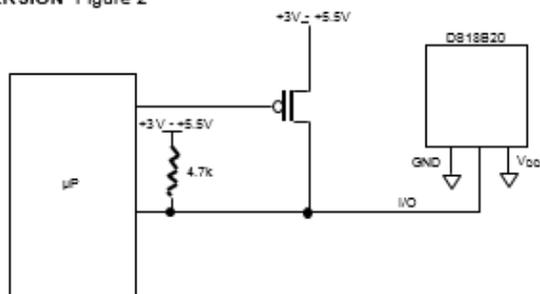
There are two ways to assure that the DS18B20 has sufficient supply current during its active conversion cycle. The first is to provide a strong pullup on the DQ line whenever temperature conversions or copies to the E² memory are taking place. This may be accomplished by using a MOSFET to pull the DQ line directly to the power supply as shown in Figure 2. The DQ line must be switched over to the strong pullup within 10 μ s maximum after issuing any protocol that involves copying to the E² 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.

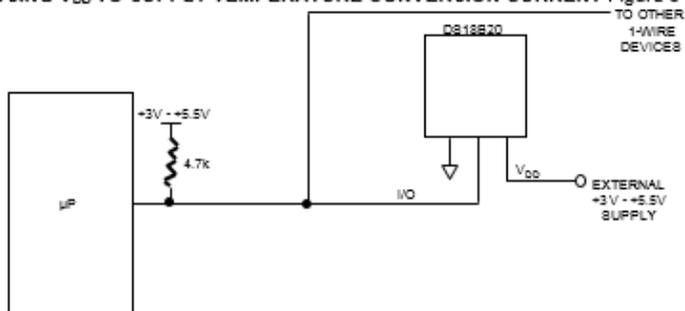
The use of parasite power is not recommended above 100°C, since it may not be able to sustain communications given the higher leakage currents the DS18B20 exhibits at these temperatures. For applications in which such temperatures are likely, it is strongly recommended that V_{DD} be applied to the DS18B20.

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 ₇	2 ₆	2 ₅	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:

$$\text{CRC} = X^8 + X^2 + X^1 + 1$$

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

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

0	R1	R0	1	1	1	1	1	
MSb								LSb

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

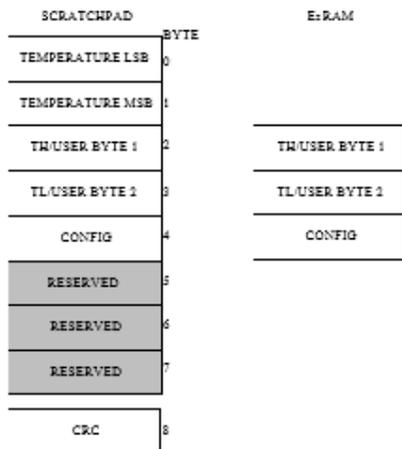
Bit 7 is a don't care on a write but will always read out "0".

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



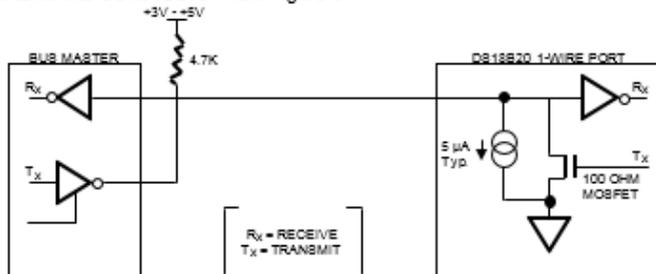
1-WIRE BUS SYSTEM

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



Figure 10 (cont'd)

MEMORY FUNCTIONS FLOW CHART

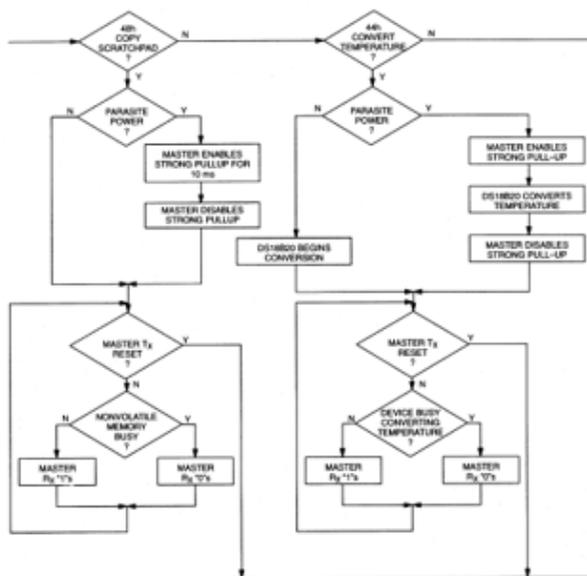
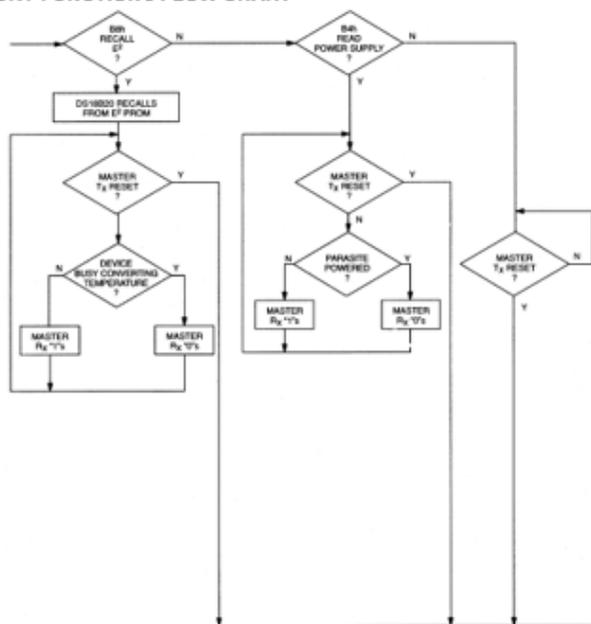
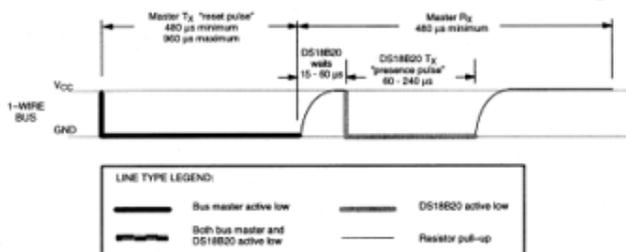


Figure 10 (cont'd)

MEMORY FUNCTIONS FLOW CHART



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:

- 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_{CON} 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.
- 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.
- 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 μ s in duration with a minimum of a 1- μ s recovery time between individual write cycles.

The DS18B20 samples the DQ line in a window of 15 μ s to 60 μ s 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 μ s 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 μ s.

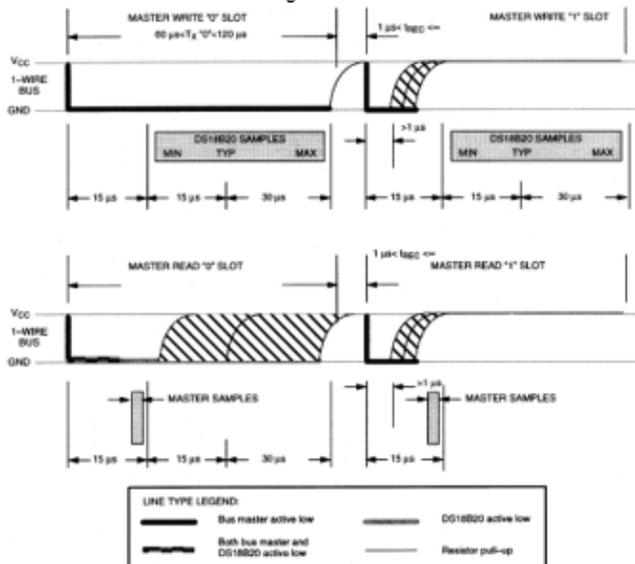
Read Time Slots

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 μ s; output data from the DS18B20 is valid for 15 μ s 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 μ s from the start of the read slot (see Figure 12). By the end of the read time slot, the DQ pin will pull back

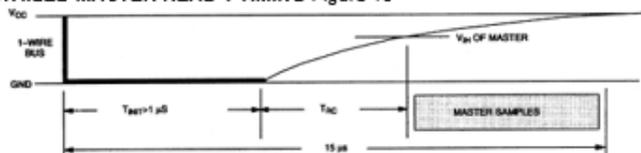
high via the external pullup resistor. All read time slots must be a minimum of $60 \mu\text{s}$ in duration with a minimum of a $1\text{-}\mu\text{s}$ recovery time between individual read slots.

Figure 12 shows that the sum of T_{DNTT} , T_{RC} , and T_{SAMPLS} must be less than $15 \mu\text{s}$. Figure 14 shows that system timing margin is maximized by keeping T_{DNTT} and T_{RC} as small as possible and by locating the master sample time towards the end of the $15\text{-}\mu\text{s}$ period.

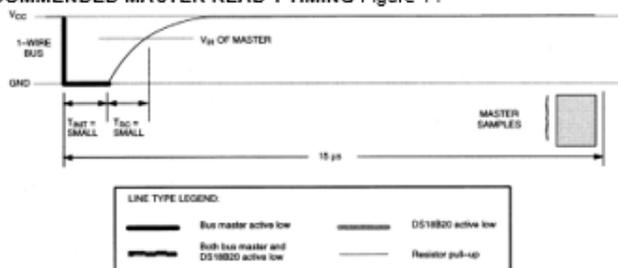
READ/WRITE TIMING DIAGRAM Figure 12



DETAILED MASTER READ 1 TIMING Figure 13



RECOMMENDED MASTER READ 1 TIMING Figure 14

**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_{CC}	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; $V_{CC}=3.0V$ to 5.5V)

PARAMETER	SYMBOL	CONDITION	MIN	TYP	MAX	UNITS	NOTES
Thermometer Error	t_{s2a}	-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	$V_{DD}=0.4V$	-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)

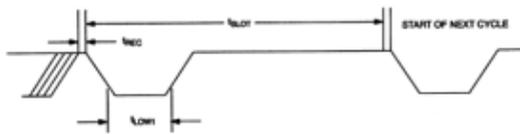
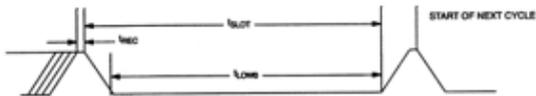
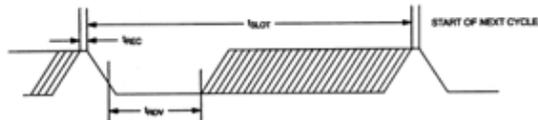
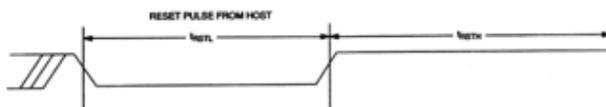
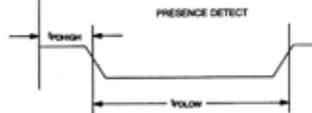
PARAMETER	SYMBOL	CONDITION	MIN	TYP	MAX	UNITS	NOTES
NV Write Cycle Time	t_{w}			2	10	ms	
EEPROM Writes	N_{max}	-55°C to +55°C	50k			writes	
EEPROM Data Retention	t_{data}	-55°C to +55°C	10			years	

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_{drv}				15	μs	
Reset Time High	t_{rstH}		480			μs	
Reset Time Low	t_{rstL}		480			μs	9
Presence Detect High	t_{prsdH}		15		60	μs	
Presence Detect Low	t_{prsdL}		60		240	μs	
Capacitance	C_{input}				25	pF	

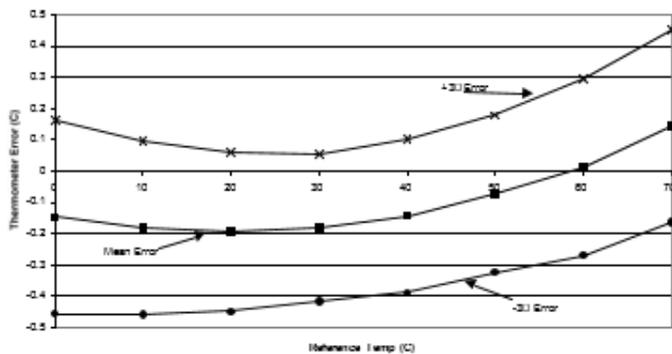
NOTES:

1. All voltages are referenced to ground.
2. Logic one voltages are specified at a source current of 1 mA.
3. Logic zero voltages are specified at a sink current of 4 mA.
4. Active current refers to either temperature conversion or writing to the E² memory. Writing to E² memory consumes approximately 200 μ A for up to 10 ms.
5. Input load is to ground.
6. Standby current specified up to 70°C. Standby current typically is 3 μ A at 125°C.
7. 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.
8. To minimize I_{DD5} , DQ should be: $GND \leq DQ \leq GND + 0.3V$ or $V_{DD} - 0.3V \leq DQ \leq V_{DD}$.
9. Under parasite power, the max t_{RSTL} before a power on reset occurs is 960 μ S.

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



TECHNICAL DATA**MQ-135 GAS SENSOR****FEATURES**

Wide detecting scope Fast response and High sensitivity Stable
and long life Simple drive circuit

APPLICATION

They are used in air quality control equipments for buildings/offices, are suitable for detecting of NH_3 , NO_x , alcohol, Benzene, smoke, CO_2 , etc.

SPECIFICATIONS**A. Standard work condition**

Symbol	Parameter name	Technical condition	Remarks
V_C	Circuit voltage	5V±0.1	AC OR DC
V_H	Heating voltage	5V±0.1	AC OR DC
R_L	Load resistance	can adjust	
R_H	Heater resistance	33Ω±5%	Room Tem
P_H	Heating consumption	less than 300mw	

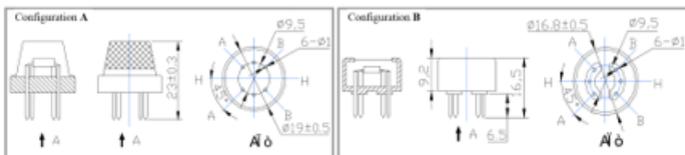
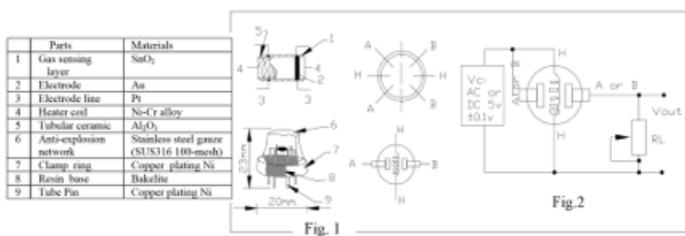
B. Environment condition

Symbol	Parameter name	Technical condition	Remarks
T_{so}	Using Tem	-10 ~45	
T_{st}	Storage Tem	-20 ~70	
R_H	Related humidity	less than 95%Rh	
O_2	Oxygen concentration	21%(standard condition)Oxygen concentration can affect sensitivity	

C. Sensitivity characteristic

Symbol	Parameter name	Technical parameter	Remark 2
R_s	Sensing Resistance	30KΩ-200KΩ (100ppm NH_3)	Detecting concentration scope 10ppm-300ppm NH_3 10ppm-1000ppm Benzene 10ppm-300ppm Alcohol
α (200/50) NH_3	Concentration Slope rate	≈0.65	
Standard Detecting Condition	Temp: 20±2 Humidity: 65%±5%	V_C : 5V±0.1 V_H : 5V±0.1	
Preheat time	Over 24 hour		

D. Structure and configuration, basic measuring circuit



Structure and configuration of MQ-135 gas sensor is shown as Fig. 1 (Configuration A or B), sensor composed by micro Al_2O_3 ceramic tube, Tin Dioxide (SnO_2) sensitive layer, measuring electrode and heater are fixed into a crust made by plastic and stainless steel net. The heater provides necessary work conditions for work of sensitive components. The enveloped MQ-135 have 6 pin, 4 of them are used to fetch signals, and other 2 are used for providing heating current.

Electric parameter measurement circuit is shown as Fig.2

E. Sensitivity characteristic curve

Fig.2 sensitivity characteristics of the MQ-135

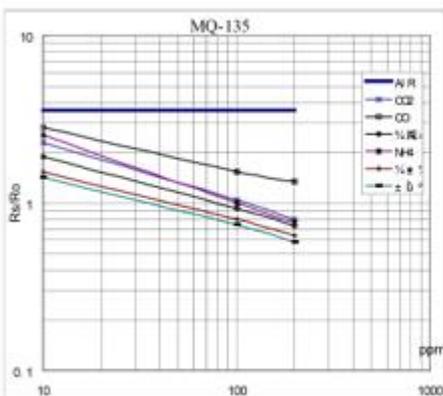


Fig.3 is shows the typical sensitivity characteristics of the MQ-135 for several gases. In their: Temp: 20^o C Humidity: 65% O₂ concentration 21% R_L = 20K Ω R_s: sensor resistance at 100ppm of NH₃ in the clean air. R₀: sensor resistance at various concentrations of gases.

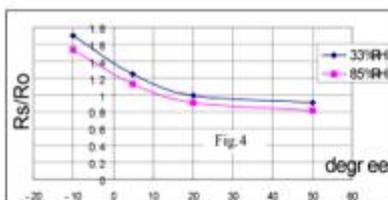


Fig.4 is shows the typical dependence of the MQ-135 on temperature and humidity. R_s: sensor resistance at 100ppm of NH₃ in air at 33%RH and 20 degree. R₀: sensor resistance at 100ppm of NH₃ at different temperatures and humidities.

SENSITIVITY ADJUSTMENT

Resistance value of MQ-135 is difference to various kinds and various concentration gases. So, When using its components, sensitivity adjustment is very necessary. we recommend that you calibrate the detector for 00ppm NH₃ or 50ppm Alcohol concentration in air and use value of Load resistance that (R_L) about 20 K Ω (10K Ω + 47 K Ω).

When accurately measuring, the proper alarm point for the gas detector should be determined after considering the temperature and humidity influence.



MG811 CO2 Sensor

Features

Good sensitivity and selectivity to CO₂
 Low humidity and temperature
 dependency Long stability and
 reproducibility

Application

Air Quality Control
 Ferment Process Control
 Room Temperature CO₂ concentration
 Detection



Structure and Testing Circuit

Sensor Structure and Testing Circuit as
 Figure, it composed by solid electrolyte layer
 (1), Gold electrodes (2), Platinum Lead
 (3), Heater (4), Porcelain Tube (5),
 100mm
 double-layer stainless steel net (6), Nickel
 and copper plated ring (7), Bakelite (8),
 Nickel and copper plated pin (9).

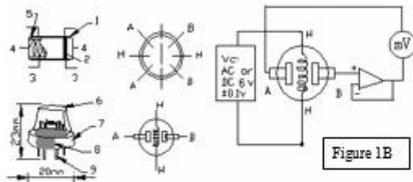


Figure 1B

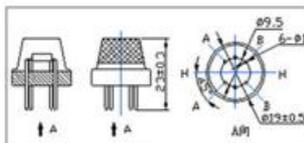
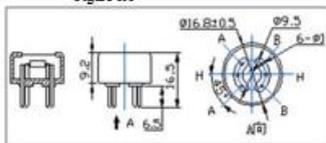


Figure 1A



Working Principle

Sensor adopt solid electrolyte cell Principle, It is composed by the following solid cells : Air,
 $\text{Au}|\text{NASICON}|\text{carbonate}|\text{Au}$, air, CO₂

caused by misuse, please read the manual carefully and operate it correctly in accordance with the instructions. If users disobey the terms or remove, disassemble, change the components inside of the sensor, we shall not be responsible for the loss.

The specific such as color, appearance, sizes &etc, please in kind prevail.

We are devoting ourselves to products development and technical innovation, so we reserve the right to improve the products without notice. Please confirm it is the valid version before using this manual. At the same time, users' comments on optimized using way are welcome.

Please keep the manual properly, in order to get help if you have questions during the usage in the future.

Zhengzhou Winsen Electronics Technology CO., LTD

MQ-7 Semiconductor Sensor for Carbon Monoxide

Profile

Sensitive material of MQ-7 gas sensor is SnO_2 , which with lower conductivity in clean air. It make detection by method of cycle high and low temperature, and detect CO at low temperature(heated by 1.5V).The sensor's conductivity gets higher along with the CO gas concentration rising. At high temperature(heated by 5.0V),it cleans the other gases adsorbed at low temperature. Users can convert the change of conductivity to correspond output signal of gas concentration through a simple circuit.



Features

It has good sensitivity to carbon monoxide in wide range, and has advantages such as long lifespan, low cost and simple drive circuit &etc.

Basic Circuit

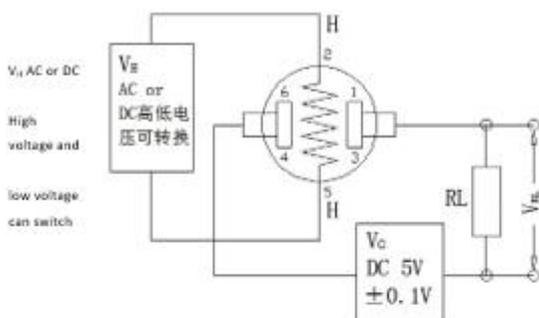


Fig2. MQ-7 Test Circuit

Instructions: The above fig is the basic test circuit of MQ-7. The sensor requires two voltage inputs: heater voltage (V_H) and circuit voltage (V_C). V_H is used to supply standard working temperature to the sensor and it can adopt DC or AC power. For this model sensor, V_H should be at $1.5V \pm 0.1V$ low voltage when detect CO while should be at $5V \pm 0.1V$ at non detection status (resuming period). V_C is the voltage of load resistance R_L which is in series with sensor. V_C supplies the detect voltage to load resistance R_L and it should adopts DC power.

Description of Sensor Characters

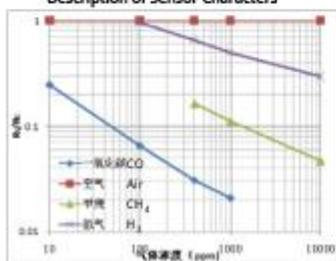
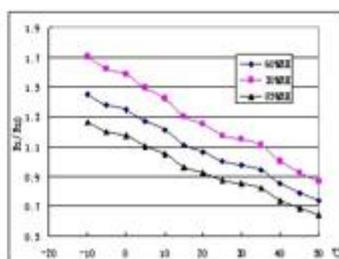


Fig3. Typical Sensitivity Curve



The ordinate is resistance ratio of the sensor (R_A/R_0), the

Fig4. Typical temperature/humidity characteristics

The ordinate is resistance ratio of the sensor (R_A/R_0), its abscissa is concentration of gases. R_A means resistance in means resistance of sensor in 150ppm CO gas under different target gas with different concentration, R_0 means zero, and h means temp. and h means humidity. R_{00} means resistance of the sensor in resistance of sensor in clean air. All tests are finished

150ppm CO gas under 20%±5%RH, under

standard test conditions.

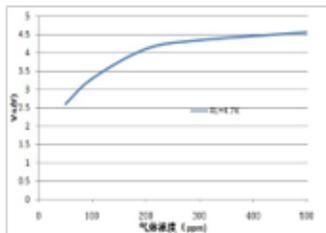


Fig5. Sensitivity Curve

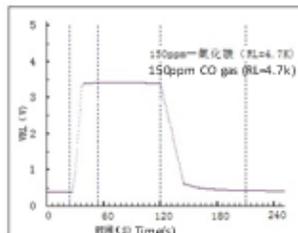


Fig6. Response and Resume

Fig5 shows the V_A in

CO gas with different concentration. Fig5 shows

the changing of V_A in

the process of putting

the sensor into target gas and removing it out.

The resistance load R_L is 4.7 kΩ and the test is finished in standard test conditions.

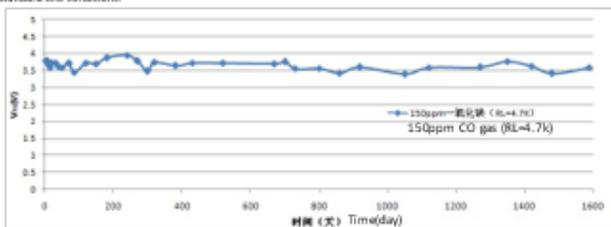


Fig7. Long-term Stability Test. It finished in standard test conditions, the

abscissa is observing time and the ordinate is V_A .

Cautions

1. Following conditions must be prohibited

1.1 Exposed to organic silicon steam

Sensing material will lose sensitivity and never recover if the sensor absorbs organic silicon steam. Sensors must avoid exposing to silicon bond, fixtural, silicon latex, putty or plastic contain silicon environment.

1.2 High Corrosive gas

If the sensors are exposed to high concentration corrosive gas (such as H_2S , SO_2 , Cl_2 , HCl etc.), it will not only result in corrosion of sensors structure, also it cause sincere sensitivity attenuation.

1.3 Alkali, Alkali metals salt, halogen pollution

The sensors performance will be changed badly if sensors be sprayed polluted by alkali metals salt especially brine, or be exposed to halogen such as fluorine.

1.4 Touch water

Sensitivity of the sensors will be reduced when splattered or dipped in water.

1.5 Freezing

Do avoid icing on sensor's surface, otherwise sensing material will be broken and lost sensitivity.

1.6 Applied higher voltage

Applied voltage on sensor should not be higher than stipulated value, even if the sensor is not physically damaged or broken, it causes down-line or heater damaged, and bring on sensors' sensitivity characteristic changed badly.

1.7 Voltage on wrong pins

For 6 pins sensor, Pin 2&5 is heating electrodes, Pin (1,3)/(4,6) are testing electrodes [Pin 1 connects with Pin 3, while Pin 4 connects with Pin 6]. If apply voltage on Pin 1&3 or 4&6, it will make lead broken; and no signal putout if apply on pins 2&4.

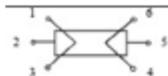


Fig8. Lead sketch

2. Following conditions must be avoided

2.1 Water Condensation

Indoor conditions, slight water condensation will influence sensors' performance lightly. However, if water condensation on sensors surface and keep a certain period, sensors' sensitive will be decreased.

2.2 Used in high gas concentration

No matter the sensor is electrified or not, if it is placed in high gas concentration for long time, sensors characteristic will be affected. If lighter gas sprays the sensor, it will cause extremely damage.

2.3 Long time storage

The sensors resistance will drift reversibly if it's stored for long time without electrify, this drift is related with storage conditions. Sensors should be stored in airproof bag without volatile silicon compound. For the sensors with long time storage but no electrify, they need long galvanical aging time for stability before using. The suggested aging time as follow: **Stable2.**

Storage Time	Suggested aging time
Less than one month	No less than 48 hours
1 ~ 6 months	No less than 72 hours
More than six months	No less than 168 hours

2.4 Long time exposed to adverse environment

No matter the sensors electrified or not, if exposed to adverse environment for long time, such as high humidity, high temperature, or high pollution etc., it will influence the sensors' performance badly.

2.5 Vibration

Continual vibration will result in sensors down-lead response then break. In transportation or assembling line, pneumatic screwdriver/ultrasonic welding machine can lead this vibration.

2.6 Concussion

If sensors meet strong concussion, it may lead its lead wire disconnected.

2.7 Usage Conditions

2.7.1 For sensor, handmade welding is optimal way. The welding conditions as follow:

- Soldering flux: Rosin soldering flux contains least chlorine
- homothermal soldering iron
- Temperature : 250°C
- Time : less than 3 seconds

2.7.2 If users choose wave-soldering, the following conditions should be obey:

- Soldering flux: Rosin soldering flux contains least chlorine
- Speed: 1-2 Meter/ Minute
- Warm-up temperature : 100±20°C
- Welding temperature : 250±10°C
- One time pass wave crest welding machine

If disobey the above using terms, sensors sensitivity will reduce.

Kitronik Ltd – [5mm RGB LED Common Cathode](#) TECHNOLOGY DATA SHEET & SPECIFICATIONS



Device Selection Guide

Chip		Lens Colour
Material	Emitted Color	
AlGaInP	Red	Water clear
InGaN	Green	
InGa1N	Blue	

Features

- Uniform light output.
- Low power consumption.
- I.C. compatible.
- Long life solderability.
- Common Cathode.

Descriptions

- The Red source colour devices are made with AlGaInP on GaAs substrate.
- The Green source colour devices are made with InGaN on sIC.
- The Blue source colour devices are made with InGaAlN on sIC.

Usage Notes:

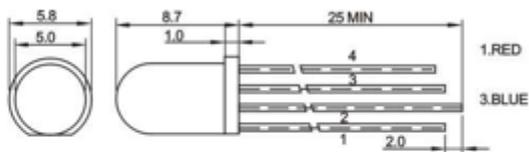
- The ultra bright LED is an electrostatic sensitive device, so static electricity and surge will damage the LED.
- It is required to wear a wrist-band when handling the LED. All device, equipment, machinery, desk and ground must be properly grounded.
- When using LED, it must use a protective resistor in series with DC current about 20mA.

Applications

- Status indicators.

- Commercial use.
- Advertising signs. □ Back lighting.

Package Dimensions



UNIT:mm

1. RED
2. COMMON CATHODE
3. BLUE
4. GREEN

Notes:

- Other dimensions are in millimetres, tolerance is 0.25mm except being specified.
- Protruded resin under flange is 1.5mm Max LED.
- Bare copper alloy is exposed at tie-bar portion after cutting.

Absolute Maximum Rating (Ta=25°C)

Parameter	Symbol	Absolute Maximum Rating	Unit
Forward Pulse Current	IFPM	70	mA
Forward Current	IFM	30	mA
Reverse Voltage	VR	5	V
Power Dissipation	PD	140	mW
Operating Temperature	Topr	-40--+80	°C
Storage Temperature	Tstg	-40--+100	°C
Soldering Temperature	Tsol	Reflow Soldering : 260 °C for 10 sec. Hand Soldering : 350 °C for 3 sec.	°C

Electro-Optical Characteristics (Ta=25°C)

Parameter	Symbol	Device	Min.	Typ.	Max.	Unit	Test Condition
Luminous Intensity	Iv	Red Green Blue	1000 1200 1000	1500 2000 1500	2300 2700 2200	md	IF=20mA
Viewing Angle	2θ1/2	Red Green Blue	40	---	50	Deg	
Peak Emission Wavelength	λp	Red Green Blue	635 520 460	640 525 465	650 530 470	nm	IF=20mA
Spectral Line Half-Width	Δλ	Red Green Blue	15 15 25	20 20 30	25 25 35	nm	IF=20mA
Forward Voltage	VF	Red Green Blue	1.9 2.9 2.9	---	2.5 3.5 3.5	V	IF=20mA
Reverse Current	IR	Red Green Blue	---	---	10	μA	VR=5V

Electrical Life Data

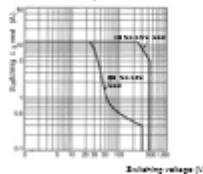
G2RL-1-C	10 Δ at 250 V _{A/C} (cosφ=1) 50,000 operations min. 10 Δ at 24 V _{D/C} 50,000 operations min. 5 Δ at 250 V _{A/C} (cosφ=0.6) 100,000 operation min. (normally open side operation) 5 Δ at 30 V _{D/C} (L/R=7 ms) 10,000 operation min. (normally open side operation)
G2RL-1	10 Δ at 250 V _{A/C} (cosφ=1) 50,000 operations min. 10 Δ at 24 V _{D/C} 50,000 operations min. 5 Δ at 250 V _{A/C} (cosφ=0.6) 100,000 operation min. (normally open side operation) 5 Δ at 30 V _{D/C} (L/R=7 ms) 10,000 operation min. (normally open side operation)
G2RL-1-H	10 Δ at 250 V _{A/C} (cosφ=1) 100,000 operations min. 10 Δ at 24 V _{D/C} 50,000 operations min.
G2RL-2	5 Δ at 250 V _{A/C} (cosφ=1) 50,000 operations min. 5 Δ at 30 V _{D/C} 50,000 operations min.

Note: 1. The results shown reflect values measured using very severe test conditions (i.e., Duty: 1 s ON/1 s OFF).
2. In order to obtain the full rated life cycles on the fully sealed models, the relay should be properly vented by removing the "knock off vent mat" on top of the relay case after the soldering/lead-in process.
3. Electrical endurance will vary depending on the test conditions. Contact your OMRON representative if you require more detailed information for the electrical endurance under your test conditions.

Engineering Data

■ Maximum Switching Capacity

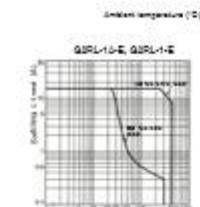
G2RL-1H, G2RL-1



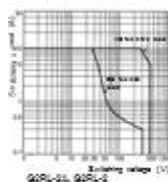
G2RL-1H, G2RL-1H

■ Ambient Temperature vs Rated Carry Current

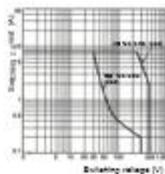




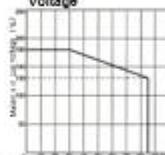
Ambient temperature (°C)



Ambient temperature (°C)



Ambient Temperature vs. Maximum Coil Voltage



Note: The maximum coil voltage refers to the maximum value in the operating range of operating power voltage, not a continuous voltage.

Note: Contact your OMRON representative for the data on fully sealed models.

DAFTAR RIWAYAT HIDUP



Nama : Heni Risa Aisyiah
TTL : Lumajang, 7
Desember 1996
Jenis Kelamin : Perempuan
Agama : Islam
Alamat Rumah : Jl. Dukuh Kupang
Barat XVII no.
10b Surabaya
Telpn/HP : 085796629623
E-mail :
heniriisa@gmail.
com

RIWAYAT PENDIDIKAN

1. 2002 – 2003 : TK Karunia
2. 2003 – 2009 : SDN Penjaringan Sari II Surabaya
3. 2009-2012 : SMP Negeri 7 Probolinggo
4. 2012 – 2015 : SMA Katolik Madei Probolinggo
5. 2015 – 2018 : Departemen Teknik Elektro
Otomasi Program Studi Komputer
Kontrol - Fakultas Vokasi (FV)
Institut Teknologi Sepuluh
Nopember (ITS)

PENGALAMAN KERJA

1. Kerja Praktek di PT. Telkom Surabaya
2. Kerja Praktek di PT. Kutai Timber Indonesia, Probolinggo

PENGALAMAN ORGANISASI

1. Staff Dalam Negeri Himpunan Mahasiswa D3 Teknik Elektro
ITS 2016-2017
2. Kabiro Hukum dan Politik Dalam Negeri Himpunan
Mahasiswa D3 Teknik Elektro ITS 2017-2018
3. Staff Hubungan Luar BEM Fakultas Teknologi Industri ITS
2016-2017

