

PENGUKURAN KADAR ALKOHOL DENGAN SENSOR MQ-3 BERBASIS ARDUINO UNO

TUGAS AKHIR

Diajukan untuk Memenuhi Salah Satu Syarat Kelulusan
Program Pendidikan Sarjana

Oleh :
Xena Mariana Sudibyo
2017130036



JURUSAN INFORMATIKA
SEKOLAH TINGGI MANAJEMEN INFORMATIKA & KOMPUTER – LIKMI
BANDUNG
2020

PENGUKURAN KADAR ALKOHOL DENGAN SENSOR MQ-3 BERBASIS ARDUINO UNO

Oleh:
Xena Mariana Sudibyo
2017130036

Bandung, Mei 2020
Menyetujui,

Dhanny Setiawan, S.T.,M.T.
Pembimbing

Dhanny Setiawan, S.T.,M.T.
Ketua Jurusan

JURUSAN INFORMATIKA
SEKOLAH TINGGI MANAJEMEN INFORMATIKA & KOMPUTER – LIKMI
BANDUNG
2020

ABSTRAK

Kesehatan merupakan faktor penting yang harus diperhatikan dalam kehidupan. Kesehatan dapat dipengaruhi oleh makanan yang sehat juga, namun pada kenyataannya banyak masyarakat yang kurang memperhatikan kesehatannya dengan mengkonsumsi alkohol. Maka dari itu dibutuhkan alat pendeteksi kadar alkohol sehingga akan lebih mudah untuk mengukur kadar alkohol yang ada. Salah satu caranya yaitu dengan menggunakan Sensor MQ-3. Sensor ini berguna untuk mengukur kadar Alkohol yang terdapat pada makanan, minuman, medis, bahan bakar, maupun kosmetik.

Pengukuran kadar alkohol yang direalisasikan ini menggunakan sensor analog yaitu Sensor MQ-3 yang dirancang menggunakan Arduino Uno, dan *Liquid Crystal Display* (LCD), selain itu juga menggunakan diagram UML berupa Diagram *Deployment*, *Class* Diagram dan *State* Diagram. Proses kerja dari alat ini alat menerima data berupa analog kemudian Arduino Uno mengelola input yang didapat dari Sensor MQ-3 dan ditampilkan pada LCD. Alat pembanding yang digunakan pada pengujian ini adalah alkoholmeter.

Pada pengujian yang telah dilakukan jarak antar Sensor MQ-3 dan sampel mempengaruhi hasil semakin dekat jarak dengan sampel maka hasil semakin mendekati pembanding. Sehingga, semakin dekat jarak sampel dengan Sensor MQ-3 maka semakin akurat hasil yang didapat sedangkan semakin jauh antara jarak dan sampel maka hasil semakin berbeda dengan pembanding. Alat ini diharapkan dapat bermanfaat untuk pengukuran kadar alkohol agar lebih mudah tanpa memerlukan jumlah ml (mililiter) alkohol yang banyak dan diharapkan dapat berguna bagi kehidupan sehari-hari.

KATA PENGANTAR

Puji Syukur penulis panjatkan kepada Tuhan Yang Maha Esa karena atas berkat dan rahmat-Nya yang diberikan sehingga penulis dapat menyelesaikan tugas akhir dengan judul “Pengukuran Kadar Alkohol Dengan Sensor MQ-3 Berbasis Arduino Uno” dengan baik dan tepat waktu.

Tugas akhir ini tentu tidak akan terselesaikan jika tanpa adanya bantuan serta dukungan secara langsung maupun tidak langsung dari berbagai pihak. Maka dari itu, penulis sampaikan ucapan terimakasih kepada :

1. Bapak Dhanny Setiawan, S.T.,M.T. selaku dosen pembimbing dan ketua program studi S1 Informatika STMIK LIKMI yang telah banyak membantu dan membimbing penulis dalam menyelesaikan tugas akhir baik dalam saran, kritik, arahan serta motivasi selama penulis menyelesaikan tugas akhir ini.
2. Bapak Sudimanto, S.T.M.Kom selaku dosen wali yang telah memberikan dukungan, saran, kritik, arahan dan juga motivasi kepada penulis untuk menyelesaikan tugas akhir ini.
3. Keluarga besar penulis yang telah memberikan dukungan doa dan menyediakan berbagai fasilitas yang menunjang penulis untuk menyelesaikan tugas akhir ini.
4. Sahabat-sahabat penulis yang selalu menemani dan mendukung penulis untuk menyelesaikan tugas akhir dengan saling memberikan motivasi dan menyemangati.
5. Seluruh teman Angkatan 2016,2017 dan seluruh teman-teman yang tidak disebutkan satu per satu yang membantu dalam menyusun tugas akhir ini.

Penulis menyadari bahwa dalam penulisan tugas akhir ini masih belum sempurna, oleh sebab itu penulis menerima segala kritik dan saran yang akan membangun dalam penyempurnaan tugas akhir ini. Penulis berharap semoga tugas akhir ini dapat bermanfaat dan menambah wawasan bagi setiap pembacanya.

Penulis

DAFTAR ISI

ABSTRAK.....	i
KATA PENGANTAR	ii
DAFTAR ISI.....	iii
DAFTAR GAMBAR	v
DAFTAR TABEL	i
DAFTAR SIMBOL	ii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penelitian	2
1.4 Batasan Masalah	3
1.5 Kegunaan Hasil.....	3
1.6 Metodologi Penelitian.....	3
1.7 Sistematika Penulisan.....	3
BAB II LANDASAN TEORI.....	5
2.1 Alkohol.....	5
2.1.1 Manfaat Alkohol.....	5
2.1.2 Bahaya Alkohol	6
2.2 Mikrokontroler.....	7
2.3 Arduino	9
2.4 Sensor	12
2.5 LCD (<i>Liquid Crystal Display</i>).....	15
2.6 Arduino IDE (<i>Integrated Development Environment</i>).....	16
2.7 Bahasa Pemograman Arduino	17
2.8 Rekayasa Perangkat Lunak	17
2.9 Pemograman Berorientasi Objek.....	18
2.10 <i>The Unified Modeling Language</i> (UML)	18

2.10.1	<i>Use Case Diagram</i>	20
2.10.2	<i>Deployment Diagram</i>	20
2.10.3	<i>Class Diagram</i>	21
2.10.4	<i>State Machine Diagram</i>	21
BAB III ANALISIS DAN PERANCANGAN		23
3.1	Gambaran Umum	23
3.2	Spesifikasi Kebutuhan	23
3.2.1	Kebutuhan Perangkat Keras	23
3.2.2	Kebutuhan Perangkat Lunak	23
3.3	Diagram <i>Deployment</i>	24
3.4	<i>Class Diagram</i>	25
3.5	<i>State Machine Diagram</i>	25
BAB IV IMPLEMENTASI DAN PENGUJIAN		26
4.1	Implementasi Perangkat Keras	26
4.2	Implementasi Kalibrasi Sensor MQ-3	28
4.3	Pengujian dan Analisa Perangkat Keras	28
BAB V KESIMPULAN DAN SARAN		31
5.1	Kesimpulan	31
5.2	Saran	31
DAFTAR PUSTAKA		32
LAMPIRAN		34

DAFTAR GAMBAR

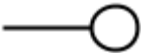
Gambar 1.1 Alkoholmeter	2
Gambar 2.1 Blok Diagram Mikrokontroler ATmega 328P	9
Gambar 2.2 Arduino Uno	12
Gambar 2.3 Sensor dalam suatu bagian sistem pengukuran.....	13
Gambar 2.4 Sensor MQ-3.....	14
Gambar 2.5 LCD (<i>Liquid Crystal Display</i>).....	16
Gambar 2.6 Jendela Arduino Uno	16
Gambar 2.7 Diagram UML 2.5.....	19
Gambar 2.8 Contoh Behaviolar <i>State Machines</i>	22
Gambar 2.9 Contoh Protokol <i>State Machines</i>	22
Gambar 3.1 Diagram <i>Deployment</i> Alat Ukur kadar Alkohol	24
Gambar 3.2 <i>Class</i> Diagram Alat Ukur kadar Alkohol.....	25
Gambar 3.3 Diagram <i>State Machine</i> Alat Ukur kadar Alkohol.....	25
Gambar 4.1 Papan Arduino UNO	26
Gambar 4.2 Sensor MQ-3.....	26
Gambar 4.3 LCD (<i>Liquid Crystal Display</i>).....	27
Gambar 4.4 Alat ukur kadar alkohol berbasis Arduino	27

DAFTAR TABEL

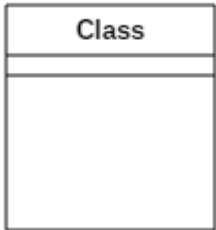
Tabel 2.1 Spesifikasi Arduino Uno	10
Tabel 2.2 Karakteristik Sensor	13
Tabel 2.3 Spesifikasi LCD	15
Tabel 4.1 Pengujian Sensor MQ-3.....	28
Tabel 4.2 Pengujian Sistem Alat ukur kadar alkohol	29

DAFTAR SIMBOL

Diagram *Deployment*

Nama Simbol	Simbol	Keterangan
<i>Node</i>		Mendeskripsikan sumberdaya komputasi, minimal memiliki <i>memory</i> dan kemampuan untuk memproses.
Komponen		Merupakan komponen dari sistem.
<i>Link</i>		Merupakan relasi antar <i>node</i> .
<i>Provided Interface</i>		Mendeskripsikan keterikatan kebutuhan dari sebuah komponen dan <i>node</i> .
<i>Required Interface</i>		Mendeskripsikan keterikatan kebutuhan dari sebuah komponen dan <i>node</i> .

Class Diagram

Nama Simbol	Simbol	Keterangan
<i>Class</i>		Merupakan himpunan dari objek-objek yang berbagi atribut operasi yang sama.
<i>Association</i>		Merupakan sebuah relasi antarkelas memiliki arti umum.

State Machine

Nama Simbol	Simbol	Keterangan
Kondisi Awal		Menggambarkan kondisi awal dari sebuah kejadian pada <i>State Machine</i> .
<i>State</i>		Kondisi selama objek menjalankan beberapa aktivitas.
Kondisi Akhir		Menggambarkan kondisi akhir dari sebuah kejadian pada <i>State Machine</i> .
<i>Transition</i>		Merupakan perpindahan dari <i>state</i> ke <i>state</i> berikutnya.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada saat ini salah satu hal penting yang harus diperhatikan oleh setiap umat manusia ialah kesehatan. Jika setiap manusia memiliki kesehatan yang baik dengan melakukan pola hidup sehat seperti aktivitas positif yaitu berolahraga maka akan meminimalisirkan penyakit yang ada. Bukan hanya olahraga saja tetapi tubuh sehat tercipta oleh makan dan minuman yang sehat juga. Dengan gaya hidup seperti mengurangi konsumsi makanan yang menggunakan bahan pengawet, meminum alkohol secara berlebihan, mengkonsumsi makanan atau minuman dengan pewarna dan banyak lagi dapat menyebabkan berbagai penyakit. Seperti mengkonsumsi alkohol akan menyebabkan beberapa gangguan kesehatan yaitu ketergantungan pada alkohol, penyakit sirosis hepar, kanker dan luka-luka akibat efek langsung maupun tidak langsung dari intoksikasi alkohol.

Penyalahgunaan alkohol ini telah menjadi masalah hampir di setiap negara di dunia. Menurut hasil penelitian yang dipublikasikan oleh alcohol.org sebanyak 6 orang meninggal setiap harinya akibat overdosis alkohol yang diawali dengan muntah, disorientasi, kesulitan bernapas, detak jantung yang cepat, kehilangan kesadaran, hingga kejang dan henti jantung.

Salah satu alat pengukur kadar alkohol yang beredar di masyarakat ialah alkoholmeter. Cara pengukuran alkoholmeter ini ialah masukan alkohol ke dalam gelas ukur yang tingginya lebih panjang dari alkoholometer. Kemudian masukan batang alkoholmeter ke dalam gelas ukur. Alkoholmeter akan tenggelam dan batas atasnya akan menunjukkan berapa kandungan alkohol di dalam larutan tersebut. Namun dalam pengukuran menggunakan alkoholmeter ini membutuhkan jumlah alkohol yang cukup banyak.



Gambar 1.1
Alkoholmeter

(Sumber: <https://shopee.co.id/Hydrometer-Alcoholmeter-Tester-Set-Alcohol-Concentration-Meter-Thermometer-i.80091031.1423510850>)

Berdasarkan permasalahan ini, perlu dibuat sebuah alat pengukur kadar alkohol yang nantinya diharapkan akan mempermudah masyarakat dalam pengecekan kadar alkohol yang ada. Maka dari itu penulis berharap alat ini dapat membantu masyarakat dalam mengukur kadar alkohol yang ada.

Penulis mencoba membuat suatu alat dan penulisan Tugas Akhir ini dengan judul **“Pengukuran Kadar Alkohol dengan Sensor MQ-3 berbasis Arduino Uno”**.

1.2 Rumusan Masalah

Hal-hal yang menjadi masalah untuk diselesaikan dalam tugas akhir ini yaitu :

1. Bagaimana cara membuat alat yang dapat mengukur kadar alkohol berbasis Arduino?
2. Bagaimana alat dapat memberikan informasi kadar alkohol?

1.3 Tujuan Penelitian

Berikut ini adalah Tujuan Penelitian :

1. Merancang alat pengukur kadar alkohol yang mampu mengukur bermacam-macam kadar alkohol.
2. Sebagai salah satu syarat kelulusan program pendidikan Sarjana di STMIK LIKMI.

1.4 Batasan Masalah

Berikut ini adalah batasan masalah yang ada dalam tugas akhir sebagai berikut :

1. Sensor yang digunakan untuk mengukur kadar alkohol adalah sensor MQ-3.
2. Hasil pengukuran kadar alkohol ditampilkan pada LCD 16 karakter x 2 baris.
3. Pengujian ini menggunakan kadar alkohol etanol dengan kandungan 1%-95%.
4. Pengaruh suhu ruang tidak digunakan pada penelitian ini.

1.5 Kegunaan Hasil

Kegunaan hasil penelitian ini adalah mewujudkan perancangan alat pengukur kadar alkohol yang nantinya akan berguna untuk masyarakat.

1.6 Metodologi Penelitian

Metodologi penelitian yang dilakukan yaitu :

1. Studi Pustaka

Pada tahap ini penulis mengumpulkan bahan-bahan sebagai referensi, baik dengan cara membaca buku, jurnal, ataupun mencari sumber-sumber lainnya di internet.

2. Perencanaan alat

Pada tahap ini penulis merancang perangkat keras dari komponen-komponen yang dibutuhkan.

3. Pengujian alat

Pada tahap ini penulis menguji alat ukur kadar alkohol dan mencatat setiap hasil yang ada, pengujian tersebut akan didiskusikan dengan dosen pembimbing untuk menarik kesimpulan dan memberikan saran pada tugas akhir ini

1.7 Sistematika Penulisan

Sistematika penulisan tugas akhir ini akan dibagi menjadi lima bab, dimana bab-bab yang akan dibuat sebagai berikut :

BAB I PENDAHULUAN

Bab ini berisikan tentang latar belakang masalah, rumusan masalah, tujuan penelitian, batasan masalah, kegunaan hasil, metodologi penelitian dan sistematika penulisan tugas akhir.

BAB II LANDASAN TEORI

Bab ini berisikan tentang teori alkohol, mikrokontroler, arduino, sensor, LCD, bahasa pemrograman arduino, rekayasa perangkat lunak, pemrograman berorientasi objek dan UML yang digunakan dalam pembahasan sistem perancangan alat kadar ukur alkohol.

BAB III ANALISI DAN PERANCANGAN

Bab ini berisikan tentang gambaran umum, spesifikasi kebutuhan perangkat keras dan perangkat lunak, diagram *deployment*, *class* diagram, dan *state machine* untuk pengukuran kadar alkohol.

BAB IV IMPLEMENTASI DAN PENGUJIAN

Bab ini berisikan tentang implementasi perangkat keras, pengujian dan analisa dari sensor MQ-3.

BAB V KESIMPULAN DAN SARAN

Bab ini berisikan tentang kesimpulan dan saran yang dilakukan selama penelitian berlangsung.

BAB II

LANDASAN TEORI

2.1 Alkohol

Yurkanis Paula Bruice menuliskan pada bukunya yang berjudul "*Organic Chemistry: Eighth Edition*" bahwa alkohol :

"An alcohol is a compound in which a hydrogen of an alkane has been replaced by an OH group." (Bruice, 2016:104)

Pada buku "*Kimia Organik*" tertulis pengertian alkohol sebagai berikut :

"Alkohol (atau alkanol) adalah istilah yang umum untuk senyawa organik apa pun yang dimiliki gugus hidroksil ($-OH$) yang terikat pada atom karbon, yang ia sendiri terikat pada atom hidrogen dan/atau atom karbon lain." (Subandi, 2010:52)

Dari pengertian diatas dapat diambil kesimpulan, alkohol merupakan nama yang digunakan untuk penggolongan senyawa organik, mempunyai gugus fungsi $-OH$ yang memiliki keterikatan dengan atom hidrogen maupun atom lainnya. Alkohol dapat memiliki sifat sebagai asam yang dapat menyumbangkan protonnya dan bersifat sebagai basa untuk menerima proton. Sedangkan air memiliki sifat asam, maka dari itu alkohol dapat menyumbang proton pada air.

Alkohol yang digunakan untuk bahan utama dari pembuatan minuman keras ialah etanol, memiliki sifat psikoaktif yang menyebabkan penurunan kesadaran bagi yang mengonsumsinya. Memiliki sifat berwarna bening yang mudah menguap, memiliki bau dengan ciri khasnya sendiri, dapat dengan mudah terbakar yang mengeluarkan nyala api berwarna biru dan tidak mengeluarkan asap.

2.1.1 Manfaat Alkohol

Menurut Veni Jumila Danin pada jurnalnya yang berjudul "*Manfaat Alkohol Dalam Kehidupan Sehari-Hari*" alkohol memiliki manfaat sebagai berikut :

1. Pada bidang kesehatan alkohol digunakan (kadar 60-90%) untuk pembasmi bakteri, penawar racun, antiseptik atau bahan pelarut.
2. Pada bidang industri kecantikan alkohol digunakan (kadar 11-15%) untuk pembersih wajah dan parfum.

3. Pada bidang industri bahan bakar, pada kebijakan pemerintah bahan bakar yaitu bensin harus mengandung 20-25% alkohol.
4. Pada bidang pangan alkohol dapat digunakan sebagai rum dengan kadar hingga 30%.
 Pada minuman alkohol, tertulis pada Peraturan Presiden Republik Indonesia Nomor No 74 Tahun 2013 tentang Pengawasan Dan Pengendalian Minuman Beralkohol tertulis pada pasal 3 bahwa:
 - a. Minuman Beralkohol yang berasal dari produksi dalam negeri atau asal impor dikelompokkan dalam golongan sebagai berikut:
 - (1) Minuman Beralkohol golongan A adalah minuman yang mengandung etil alkohol atau etanol (C_2H_5OH) dengan kadar sampai dengan 5% (lima persen);
 - (2) Minuman Beralkohol golongan B adalah minuman yang mengandung etil alkohol atau etanol (C_2H_5OH) dengan kadar lebih dari 5% (lima persen) sampai dengan 20% (dua puluh persen); dan
 - (3) Minuman Beralkohol golongan C adalah minuman yang mengandung etil alkohol atau etanol (C_2H_5OH) dengan kadar lebih dari 20% (dua puluh persen) sampai dengan 55% (lima puluh lima persen).
 - b. Minuman Beralkohol sebagaimana dimaksud pada ayat (1) ditetapkan sebagai barang dalam pengawasan.
 - c. Pengawasan sebagaimana dimaksud pada ayat (2) meliputi pengawasan terhadap pengadaan Minuman Beralkohol yang berasal dari produksi dalam negeri atau asal impor serta peredaran dan penjualannya.

2.1.2 Bahaya Alkohol

Pada jurnal yang berjudul “Konsumsi Alkohol dan Pengaruhnya terhadap Kesehatan” yang dituliskan oleh Topaz Kautsar Tritama menuliskan bahwa bahaya mengkonsumsi alkohol termasuk dalam lima besar faktor resiko untuk penyakit, kecacatan dan kematian di seluruh dunia. Resiko kesehatan tersebut dapat berupa :

1. Ketergantungan terhadap alkohol.
2. Sirosis hepar.
3. Luka-luka akibat efek langsung maupun tidak langsung dari intoksikasi alkohol.

4. Impotensi dan berkurangnya kesuburan, kesulitan tidur, kerusakan otak dengan perubahan kepribadian dan suasana perasaan,
5. Gangguan ingatan dan gangguan konsentrasi pada penggunaannya.
6. Disinhibisi, ataksia dan sedasi karena alkohol dapat mengganggu pengaturan eksitasi (penambahan) atau inhibisi (hambatan) di otak.
7. Dapat membuat perubahan suasana hati yang tidak terkontrol dan gejala emosi yang dapat disertai kekerasan pada orang yang mengkonsumsinya.
8. Konsumsi alkohol yang berlebihan dapat menyebabkan peningkatan tekanan darah yang kemudian dapat menjadi hipertensi, kerusakan jantung, stroke, kanker payudara, kerusakan hati, kanker saluran pencernaan dan gangguan pencernaan lainnya.
9. Mengonsumsi alkohol pada kehamilan dapat menyebabkan Sindrom Fetal Alkohol yang dapat mengganggu pertumbuhan dan perkembangan janin tersebut.

2.2 Mikrokontroler

Alan G Smith menuliskan pengertian mikrokontroler pada bukunya yang berjudul *"Introduction to Arduino: A piece of cake!"* yang dikutip dari Wikipedia sebagai berikut :

"A micro-controller is a small computer on a single integrated circuit containing a processor core, memory, and programmable input/ output peripherals" (Smith, 2011:1)

Pengertian mikrokontroler pada buku dengan judul *"Mikrokontroler Konsep Dasar dan Praktis"* menyebutkan bahwa :

"Mikrokontroler merupakan chip mikrokomputer yang secara fisik berupa sebuah IC (*Integrated Circuit*)" (Dharmawan, 2017:1)

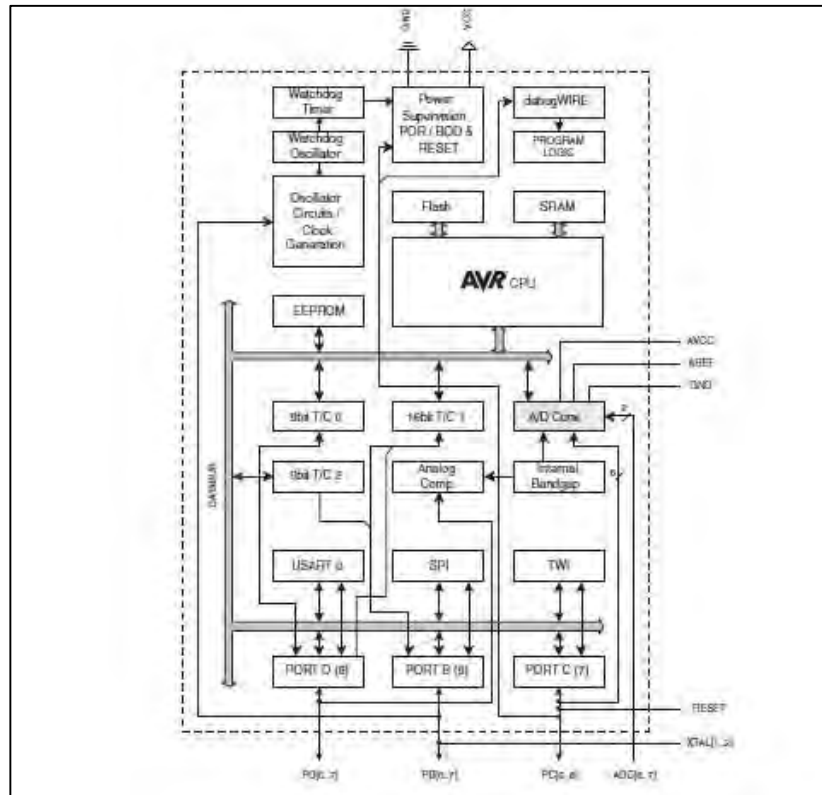
Sehingga dapat disimpulkan bahwa mikrokontroller merupakan prosesor dengan sejumlah memori dan sejumlah *peripheral interfaces* yang terintegrasi pada satu cip, cip atau IC (*integrated circuit*) tersebut dapat diprogram menggunakan komputer. Menanamkan program pada mikrokontroler memiliki tujuan yaitu agar setiap rangkaian suatu elektronik dapat membaca masukan, memproses masukan dan kemudian menghasilkan keluaran sesuai dengan yang diinginkan. Sehingga, dapat disimpulkan

bahwa mikrokontroler memiliki tugas sebagai otak untuk mengendalikan masukan, proses dan keluaran sebuah rangkaian elektronik. Mikrokontroler terdapat pada perangkat elektronik, seperti handphone, MP3 player, DVD, televisi, AC dan lain-lain. Selain itu Mikrokontroler dapat digunakan sebagai pengendali robot, baik robot mainan maupun robot industri. Program yang ditanamkan pada mikrokontroler merupakan sebuah instruksi-instruksi dalam bentuk kode yang dibuat dengan bahasa pemrograman tertentu.

2.2.1 Mikrokontroler ATmega 328

Mikrokontroler ATmega 328 merupakan mikrokontroler dengan 8-bit CMOS (*Complementary Metal Oxide Semiconductor*) yang berdaya rendah dan salah satu mikrokontroler keluaran dari atmel dengan arsitektur RISC (*Reduce Instruction Set Computer*) yang disempurnakan. Agar mekanisme berjalan dengan maksimal dan paralelisme maka digunakannya arsitektur Harvard yang memori dan bus untuk program dan datanya. Instruksi dalam memori program dijalankan dengan satu tingkat *pipelining* (kerja secara bersamaan tetapi dalam tahap yang berbeda dialirkan secara kontinu pada unit pemrosesor). Ketika satu instruksi sedang dieksekusi maka instruksi berikutnya akan diambil dari memori program.

Mikrokontroler ATmega 328 memiliki kapasitas flash yaitu program memori sebesar 32 KB (*Kilobyte*), SRAM (*Static Random Access Memory*) sebesar 2 KB (*Kilobyte*), EEPROM (*Electrically Erasable Programmable Read Only Memory*) sebesar 1 KB (*Kilobyte*), USART (*Universal Synchronous-Asynchronous Receiver/Transmitter*) serial yang dapat diprogram dan SPI (*Serial Peripheral Interface*) serial port. Kecepatan maksimum yang dapat dicapai adalah 20 MHz. Dalam cip yang dipaketkan pada mikrokontroler ATmega 328 dalam bentuk *dual in-line* terdapat 20 pin masukan atau keluaran (21 pin jika pin reset tidak digunakan, 23 pin jika tidak menggunakan osilator eksternal), dengan 6 diantaranya dapat berfungsi sebagai pin ADC (*analog-to-digital converter*), dan 6 lainnya memiliki fungsi sebagai PWM (*pulse width modulation*).



dengan lampu dan tombol. Ketika tombol ditekan Arduino akan menyalakan lampu dan mulai menghitung. Setelah dihitung sebanyak 30 detik, lampu akan dimatikan dan kemudian lampu akan menyala kembali jika tombol ditekan.

Papan Arduino ini terdiri dari mikroprosesor Atmel AVR, kristal atau osilator dan regulator tegangan 5V. Beberapa jenis Arduino memiliki soket USB agar dapat terhubung ke komputer untuk mengunggah atau mengambil data. Papan akan mengekspos pin masukan atau keluaran mikrokontroler agar dapat menghubungkan pin tersebut ke sirkuit lain atau ke sensor. Agar Arduino dapat diprogram untuk melakukan apa yang diinginkan maka dari itu memerlukan perangkat lunak yaitu Arduino IDE, dengan bahasa C / C++ dan dapat diperpanjang melalui perpustakaan C++. IDE memungkinkan untuk menulis program komputer, yang merupakan serangkaian instruksi langkah demi langkah yang kemudian unggah ke Arduino. Arduino akan memproses instruksi tersebut kemudian jalankan instruksi tersebut dan berinteraksi dengan apa pun yang terhubung dengan Arduino tersebut.

2.3.1 Arduino Uno

Arduino Uno ialah salah satu papan mikrokontroler berbasis mikrokontroler ATmega328P dengan 8-bit. Arduino Uno tersebut memiliki 14 pin masukan atau keluaran digital dengan 6 pin diantaranya dapat digunakan sebagai keluaran PWM, 6 pin masukan analog, koneksi USB, *jack* listrik, *header* ICSP dan tombol res.

Tabel 2.1
Spesifikasi Arduino Uno

Mikrokontroler	ATmega328P – 8bit AVR mikrokontroler
Tegangan Operasi	5 Volt
Tegangan masukan yang disarankan	5-12 Volt
Batas tegangan masukan	5-20 Volt
Pin masukan analog	6 (A0 – A5)
Pin masukan / keluaran digital	14 (6 pin untuk keluaran PWM)
Arus DC pada pin masukan / keluaran	40 mA
Arus DC pada pin 3.3 Volt	50 mA
Memori flash	32 KB (0.5 KB digunakan sebagai <i>bootloader</i>)
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

(Sumber : <https://components101.com/microcontrollers/arduino-uno>)

Berikut ini merupakan penjelasan tentang spesifikasi dari Arduino Uno menurut www.arduino.cc, sebagai berikut :

1. Catu daya arduino uno

Papan arduino dapat diaktifkan menggunakan koneksi USB ataupun menggunakan catu daya eksternal. Sumber dayanya tidak dipilih secara manual melainkan secara otomatis. Jika menggunakan daya eksternal (bukan koneksi USB) maka daya dapat berasal dari adaptor AC ke DC atau baterai. Papan arduino ini dapat beroperasi dengan stok catu daya eksternal dari 6 hingga 20 Volt. Akan tetapi, jika disediakan kurang dari 7 Volt, pin 5 Voltnya yang mungkin akan ada kurang dari lima volt tersebut dan akan menyebabkan papan menjadi tidak stabil. Jika yang digunakan lebih dari 12Volt, akan menyebabkan pengatur tegangan menjadi panas dan papan akan menjadi rusak. Disarankan daya yang digunakan adalah 7 hingga 12 volt.

2. Memori

Pada mikrokontroler ATmega328 memiliki memori 32 KB (dengan 0,5 KB ditempati oleh *bootloader*), 2 KB SRAM dan 1 KB EEPROM yang dapat dibaca dan ditulis dengan EEPROM *library*.

3. Masukan dan Keluaran

Arduino Uno memiliki 14 pin yang masing-masingnya dapat digunakan sebagai masukan dan keluaran dengan menggunakan fungsi dari *pinMode()*, *digitalWrite()*, dan *digitalRead()*. Seluruhnya beroperasi dengan 5Volt. Pada setiap pin dapat menyediakan atau menerima 20 mA sebagai kondisi yang direkomendasikan dan memiliki resistor *pull-up* bagian dalam (terputus secara default) 20-50k ohm. Agar tidak terjadi kerusakan tetap pada mikrokontroler maka nilai maksimumnya adalah 40mA yang tidak boleh dilewati pada pin masukan atau keluaran.

Beberapa pin memiliki khusus yaitu :

- a. Serial: 0 (RX) dan 1 (TX). Kegunaannya dibagi menjadi dua yaitu menerima (RX) dan mengirimkan (TX) data serial TTL. Pin ini akan terhubung ke pin yang sesuai dari cip Serial ATmega8U2 USB ke TTL.

- b. Interupsi eksternal : 2 dan 3. Pin ini dapat dikonfigurasi untuk memicu interupsi pada nilai rendah, naik atau turunnya tepi, atau perubahan nilainya.
- c. PWM: 3, 5, 6, 9, 10, dan 11. Ini menyediakan keluaran PWM 8-bit dengan fungsi *analogWrite()*.
- d. SPI : 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Pin ini berfungsi untuk mendukung komunikasi SPI menggunakan *SPI library*.
- e. LED : 13. Terdapat LED internal yang digerakkan oleh pin digital 13. Ketika pin bernilai tinggi maka LED akan menyala dan kebalikannya ketika pin rendah maka LED akan mati.
- f. TWI : pin A4 atau SDA dan pin A5 atau SCL. Mendukung komunikasi TWI menggunakan perpustakaan *Wire*.
- g. Arduino Uno ini memiliki 6 pin masukan analog dengan label A0 hingga A5, yang masing-masing akan menyediakan 10bit resolusi dengan 1024 nilai yang berbeda-beda. Pin tersebut secara *default* dapat diukur dari bawah hingga 5 volt, meskipun untuk mengubah ujungnya atas rentang menggunakan pin AREF dan fungsi *analogReference ()*.



Gambar 2.2
Arduino Uno

(Sumber : <https://components101.com/microcontrollers/arduino-uno>)

2.4 Sensor

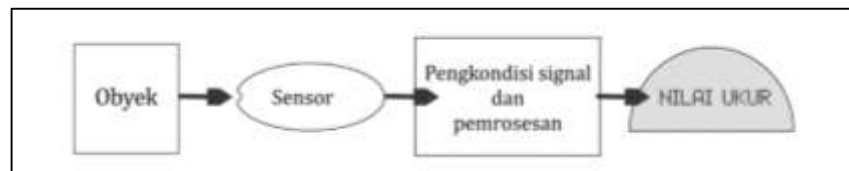
Menurut Setyawan P Sakti pada bukunya “Pengantar Teknologi Sensor: Prinsip Dasar Sensor Berdasarkan Mekanik” menuliskan pengertian sensor yang diambil dari *Comprehensive Dictionary of Instrumentation and Control* bahwa :

“Sensor didefinisikan sebagai: “sebuah nama generik untuk sebuah divaise yang mendeteksi harga absolut nilai kuantitas fisis atau perubahan harga nilai kuantitas fisis dan mengubah pengukuran menjadi sebuah signal yang berguna untuk indikator maupn instrument pencatat”” (Sakti, 2017:3)

Pada buku yang berjudul “*Beginning Sensor Networks with Arduino and Raspberry Pi*” menuliskan bahwa :

“A sensor is a device that measures phenomena of the physical world. These phenomena can be things you see, like light, gases, water vapor, and so on. They can also be things you feel, like temperature, electricity, water, wind, and so on.” (Bell,2013:10)

Sehingga dapat disimpulkan bahwa sensor merupakan perangkat yang dapat digunakan sebagai pendeteksi perubahan besaran fisik seperti tekanan, gaya, besaran listrik, cahaya, gerakan, kelembaban, suhu, kecepatan dan fenomena-fenomena lingkungan lainnya. Sensor



Gambar 2.3

Sensor dalam suatu bagian sistem pengukuran

(Sumber : Sakti, 2017:2)

Tabel 2.2
Karakteristik Sensor

Sifat	Kondisi Ideal Sensor
Hubungan input-output	Diharapkan merupakan persamaan linear sederhana.
Histerisis	Sifat yang menjelaskan perubahan dan nilai keluaran sensor pada saat terjadi perubahan nilai masukan dari kecil ke besar dan dari besar ke kecil. Taka da histerisis, respon keluaran akan kembali melalui linasan hubungan luaran dan masukan yang sama.
Selektivitas	Sensor hanya bereaksi terhadap nilai masukan yang diinginkan, dan tidak terpengaruh oleh yang lain.
Sensitivitas	Sensor mampu membedakann perubahan nilai besaran ukur yang kecil dnegan baik.
Waktu tanggapan	Sensor memiliki waktu tanggapan nol. Ketika ada perubahan masukan, maka tanpa ada penundaan waktu akan dihasilkan keluaran yang sesuai dengan masukan.

(Sumber : Sakti, 2017:15-16)

2.4.1 Sensor MQ-3

Sensor MQ-3 pada klasifikasinya merupakan sensor pasif, yang digunakan untuk mengukur kadar alkohol memiliki sensitifitas tinggi terhadap alkohol dan *breathalyzer* (pengukuran menggunakan napas) untuk seseorang yang meminum bir, anggur, atau minuman keras lainnya. Sensor MQ-3 juga memiliki kecepatan respon yang cepat dengan sensitifitas yang tinggi terhadap alkohol. Dalam penggunaannya MQ-3 memiliki hasil yang stabil dibanding dengan sensor alkohol lainnya dan dapat digunakan dalam jangka waktu yang cukup lama.

Sensor MQ-3 memiliki sumber tegangan AC atau DC sebesar 5 Volt dengan tipe interface jenis analog. Sensor MQ-3 ini memiliki 6 pin yang 4 diantaranya dapat digunakan untuk mengambil sinyal dan terdapat 2 lainnya digunakan untuk arus untuk pemanasan. Waktu yang diperlukan untuk merespon dari sensor ini relatif cukup cepat. Ukuran sensor ini sebesar 40x20mm. Sensor MQ-3 memiliki daya cukup besar yaitu 750mW. Agar sensor menghasilkan hasil akurat, titik sinyal secara tepat untuk detektor gas kemudian suhu dan kelembaban harus dipertimbangkan dengan baik.

Resistansi sensor MQ-3 ini dapat berubah-ubah seiringan dengan terbacanya keberadaan gas etanol oleh elemen sensor. Jika konsentrasi etanol tinggi, maka resistansi sensor akan berkurang dan menghasilkan tegangan keluaran yang akan meningkat. Nilai dari resistansi sensor MQ-3 memiliki perbedaan terhadap jenis dan konsentrasi gas yang terdapat didalam udara bersih, sehingga pada saat penggunaannya diperlukan penyesuaian



Gambar 2.4
Sensor MQ-3

(Sumber: <http://www.learningaboutelectronics.com/Articles/MQ-3-alcohol-sensor-circuit-with-arduino.php>)

2.5 LCD (*Liquid Crystal Display*)

Pada jurnal yang berjudul “Pendeteksi Kadar Gas Alkohol Pada Peuyeum Menggunakan Sensor Mq-3 Berbasis Arduino” yang dituliskan oleh Churrotul Aini dan Nazuwatussya’diyah menuliskan LCD (*Liquid crystal display*) merupakan jenis tampilan elektronik yang dibuat dengan teknologi CMOS *logic* yang cara kerjanya dengan memantulkan cahaya yang ada di sekelilingnya terhadap *front-lit* atau mentransmisikan cahaya dari *back-lit*.

LCD memiliki pin, kaki atau jalur input dan kontrol sebagai berikut:

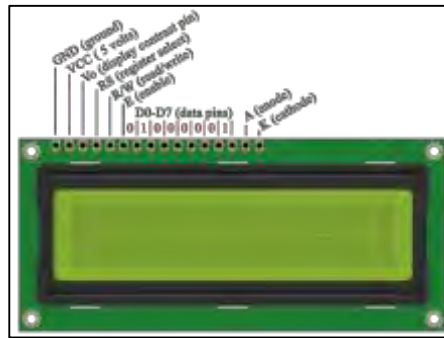
1. Pin data merupakan jalur untuk memberikan data karakter yang ingin ditampilkan menggunakan LCD (*Liquid Cristal Display*) dapat dihubungkan dengan bus data dari rangkaian lain seperti mikrokontroler dengan lebar data 8 bit.
2. Pin RS (*Register Select*) memiliki fungsi sebagai indikator atau menentukan jenis data yang masuk, yang dapat merupakan data atau perintah. Logika low akan menunjukan perintah sedangkan logika high menunjukan data.
3. Pin R/W (*Read Write*) berfungsi sebagai instruksi pada modul, *low* merupakan menutulis data, sedangkan *high* merupakan membaca data.
4. Pin E (*Enable*) digunakan untuk memegang data baik masuk atau keluar.
5. Pin VLCD berfungsi mengatur kecerahan tampilan (kontras), pin ini dihubungkan dengan trimpot 5 Kohm, jika tidak digunakan akan dihubungkan ke *ground*, sedangkan tegangan catu daya ke LCD sebesar 5 Volt.

I2C LCD merupakan modul LCD yang dikendalikan secara serial pada waktu yang sama dengan protokol I2C/IIC (*Inter Integrated Circuit*) atau TWI (*Two Wire Interface*). Dengan spesifikasi sebagai berikut:

Tabel 2.3
Spesifikasi I2C LCD

I2C Address Range	2 baris dengan 16 karakter 0x20 hingga 0x27
Tegangan operasi	5 V
Latar lampu	Putih
Kontras	Dapat disesuaikan dengan potensiometer pada I2C
Ukuran	80mm x 36mm x 20 mm 66mm x 16mm

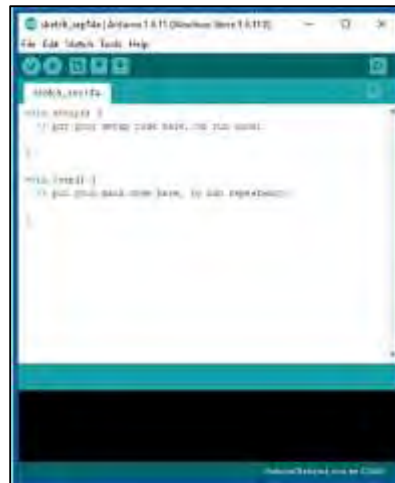
(Sumber : Datasheet I2C LCD)



Gambar 2.5
LCD (*Liquid Crystal Display*)

(Sumber : <https://howtomechatronics.com/tutorials/arduino/lcd-tutorial/>)

2.6 Arduino IDE (*Integrated Development Environment*)



Gambar 2.6
Jendela Arduino Uno

(Sumber : <https://www.microsoft.com/enus/p/arduinoide/9nblggh4rsd8?activetab=pivot:overviewtab#>)

Arduino IDE merupakan teks editor untuk menulis kode, pesan, konsol teks, *toolbar* dengan tombol-tombolnya yang memiliki fungsi masing-masing dan serangkaian menu. Terhubung ke perangkat keras Arduino dan Genuino untuk memprogram dan berkomunikasi. (www.arduino.cc) Arduino IDE membantu untuk menerjemahkan bahasa pemrograman Arduino yang sederhana menjadi hex file biner yang lebih rumit agar dapat diunggah langsung ke mikrokontroler.

Seperti pada gambar 2.6 yang merupakan Jendela Arduino Uno memiliki toolbar dengan berbagai fungsi yaitu:

1. *Verify* untuk mengecek apakah kode yang diketik sudah benar atau belum, jika terjadi kesalahan maka program tidak akan berjalan.

2. *Upload* untuk mengunggah kode ke papan yang dikonfigurasi.
3. *New* untuk membuat *sketch* yang baru.
4. *Open* untuk membuka *sketch* yang pernah dikerjakan sebelumnya.
5. *Save* untuk menyimpan *sketch* yang sedang dikerjakan.
6. *Serial Monitor* untuk mencari tahu jika terjadi kesalahan.

2.7 Bahasa Pemrograman Arduino

Bahasa pemrograman yang digunakan untuk Arduino ialah bahasa pemrograman yang mirip dengan bahasa C digunakan pada AVR. Bahasa C digunakan karena lebih sederhana dan mudah untuk dipelajari bagi pemula. Karena merupakan *multi-platform* maka bahasa C juga dapat digunakan diberbagai sistem operasi seperti Windows, Macintosh, Linux, maupun sistem operasi lainnya tanpa harus mengalami perubahan *source code*.

Kode `//` digunakan untuk pemberian komentar/catatan yang hanya bisa berisikan satu baris saja, sedangkan jika ingin memberikan komentar/catatan lebih dari satu baris dapat menggunakan kode `/* */`. Sedangkan kode `{ }` memiliki fungsi untuk menentukan awal dan akhir dari program tersebut. Masing-masing dari baris kode pada Arduino harus menggunakan `;`. Kode dalam Arduino merupakan hal yang sensitif yang maksudnya ialah dalam penggunaan huruf kecil atau huruf besar sangat berpengaruh pada hasilnya.

2.8 Rekayasa Perangkat Lunak

Pada buku *“Software Engineering: Ninth Edition”* yang dituliskan oleh Ian Sommerville bahwa :

“Software engineering is an engineering discipline that is concerned with all aspects of software production from the early stages of system specification through to maintaining the system after it has gone into use.” (Sommerville,2010:7)

Selain itu juga pada buku yang berjudul *“Rekayasa Perangkat Lunak”* yang menuliskan pengertian Rekayasa Perangkat Lunak sebagai berikut:

Rekayasa Perangkat Lunak (RPL atau SE (*Software Engineering*)) adalah satu bidang profesi yang mendalami cara-cara pengembangan perangkat lunak termasuk pembuatan, pemeliharaan, manajemen organisasi pengembangan perangkat lunak, dan sebagainya. (Simarmata,2010:10)

Berdasarkan pengertian-pengertian dari para ahli dapat disimpulkan bahwa rekayasa perangkat lunak merupakan ilmu yang berkaitan dengan perangkat lunak dari pembuatan hingga menjadi menghasilkan produk yang profesional dan pemeliharaan produk tersebut.

2.9 Pemrograman Berorientasi Objek

Pada jurnal yang berjudul “Pembelajaran Pemrograman Berorientasi Objek (*Object Oriented Programming*) Berbasis *Project Based Learning*” yang dituliskan oleh Endang Retnoningsih, Jafar Shadiq, Dony Oscar yang dikutip dari Ramadhani menuliskan pemrograman berorientasi objek ialah

“Pemrograman berorientasi objek atau *object oriented programming* (OOP) merupakan suatu pendekatan pemrograman yang menggunakan *object* dan *class*.” (Retnoningsih, 2017:96)

Terdapat pengertian lainnya pada buku yang berjudul “*UML For The It Business Analyst, Second Edition: A Practical Guide to Requirements Gathering Using The Unified Modeling Language*” ditulis oleh Howard Podeswa bahwa :

“OO is an acronym for “object-oriented”. The OO analyst sees a system as a set of objects that collaborate by sending requests to each other.” (Podeswa,2010:18)

Sehingga dapat disimpulkan bahwa Objek Orientasi merupakan pendekatan pemrograman dengan menggunakan objek yang berkolaborasi dengan mengirimkan permintaan satu sama lain dan *class*-nya.

2.10 The Unified Modeling Language (UML)

Buku “Referensi dan Panduan UML 2.4 Singkat Tepat Jelas” yang ditulis oleh Akil menuliskan pengertian UML sebagai berikut :

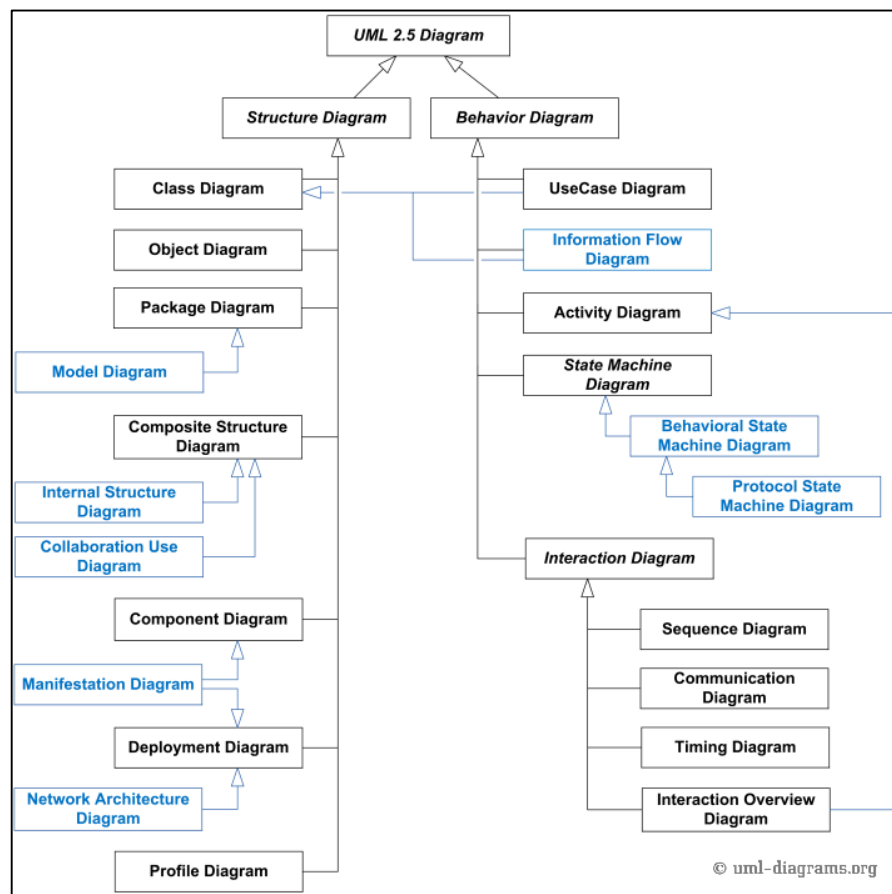
“*Unified Modeling Language* (UML) adalah bahasa pemodelan visual yang digunakan untuk menspesifikasikan, memvisualkan, membangun dan mendokumentasikan rancangan dari suatu sistem perangkat lunak” (Akil,2018:3)

Selain pada buku dengan judul “*System Analysis and Design Fifth Edition*” menuliskan bahwa UML adalah :

“UML is a standard set of diagramming techniques that provide a graphical representation rich enough to model any systems development project, from analysis through implementation.” (Alan Dennis,2012:539)

Sehingga dapat disimpulkan bahwa *The Unified Modeling Language* (UML) merupakan sebuah standard dari pembuatan pemodelan berorientasi objek untuk setiap perangkat lunak yang ada, pengembangannya dimulai dari analisis hingga pengimplementasinya.

UML atau *Unified Modeling Language* dapat diterapkan ke dalam berbagai domain aplikasi misalnya perbankan, keuangan, internet, layanan kesehatan, dan lain-lain. Selain itu juga dapat digunakan dengan metode utama pengembangan perangkat lunak objek dan komponen untuk berbagai platform implementasi.



Gambar 2.7
Diagram UML 2.5

(Sumber: <https://www.uml-diagrams.org/uml-25-diagrams.html>)

2.10.1 Use Case Diagram

Menurut uml-diagrams.org *use case diagram* merupakan diagram perilaku yang menggambarkan sebuah tindakan sebuah sistem atau sistem sebagai subjek atau dapat melakukan kolaborasi dengan satu atau lebih pengguna eksternal sistem sebagai aktor. Setiap hasil dari *use case* harus memberikan hasil yang dapat diamati dan dapat digunakan bagi para aktor atau yang memiliki kepentingan dari sistem tersebut.

Sistem *use case* pada uml-diagrams.org menuliskan bahwa *use case* digunakan untuk menentukan hal-hal sebagai berikut :

1. Persyaratan dari luar, penggunaan yang membutuhkan suatu sistem dalam desain atau analisis (subjek) untuk menangkap apa yang harus dilakukan oleh sistem.
2. Sebuah fungsi yang ditawarkan oleh subjek, yaitu apa yang dapat dilakukan oleh sebuah sistem.
3. Persyaratan yang ditentukan oleh subjek pada lingkungannya adalah dengan mendefinisikan bagaimana lingkungan tersebut harus berinteraksi dengan subjek sehingga dapat melakukan tugas-tugasnya.

2.10.2 Deployment Diagram

Pada uml-diagrams.org menuliskan pengertian *deployment diagram* ialah sebuah struktur diagram yang menampilkan arsitektur dari sistem yang didistribusi oleh pengembangan perangkat lunak untuk target *deployment* tersebut. Target *deployment* mewakili sebuah *node* (simpul) yang merupakan perangkat keras. Pengembangan perangkat lunak tersebut mewakili elemen fisik yang merupakan hasil dari proses pengembangan. Contohnya adalah file yang dapat dieksekusi, perpustakaan, arsip, skema basis data, dan file konfigurasi.

Arsitektur *deployment diagram* digambarkan menjadi dua yaitu :

1. Diagram penyebaran level spesifikasi menunjukkan beberapa gambaran umum penyebaran artefak ke target penyebaran, tanpa merujuk contoh artefak atau node tertentu.
2. Diagram penyebaran tingkat instance menunjukkan penyebaran instance artefak ke instance spesifik target penyebaran. Ini dapat digunakan misalnya untuk menunjukkan

perbedaan dalam penyebaran ke lingkungan pengembangan, pementasan atau produksi dengan nama / id dari server atau perangkat pembangunan atau penerapan tertentu.

2.10.3 *Class Diagram*

Pengertian class diagram pada uml-diagrams.org dituliskan sebagai berikut struktur diagram UML untuk menunjukkan struktur sistem yang dirancang pada tingkat setiap kelas, menunjukkan fitur, kendala, dan hubungan setiap kelasnya. Terdapat beberapa jenis *class* diagram yaitu :

1. Domain model diagram
2. Implementasi *class* diagram

2.10.4 *State Machine Diagram*

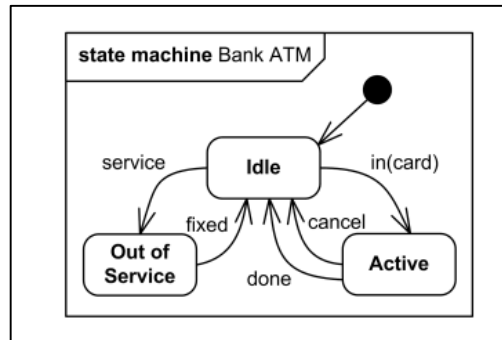
Pengertian State Machine Diagram yang dituliskan pada uml-diagram.org merupakan diagram perilaku yang menunjukkan sebuah perilaku diskrit dari bagian sistem yang dirancang melalui terbatasnya transisi keadaan.

Pada uml-diagram.org menuliskan terdapat dua jenis *State Machine* Diagram yaitu:

1. *Behavioral State Machines*

Behavioral State Machines adalah spesialisasi *behavioral* dan digunakan untuk menentukan *behavioral* diskrit dari berbagai sistem yang dirancang melalui transisi terbatasnya situasi. Didefinisikan menjadi tiga jenis yaitu :

- a. Simple state.
- b. Composite state.
- c. Submachine state.



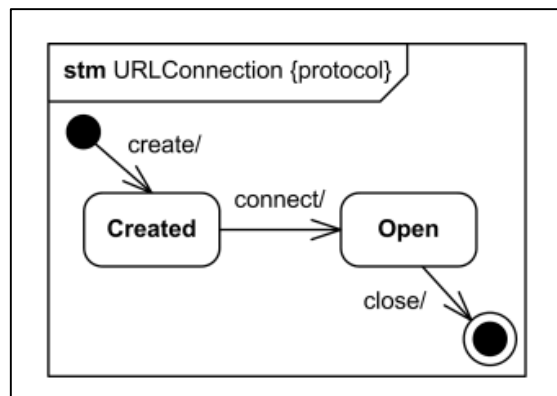
Gambar 2.8

Contoh *Behavioral State Machines*

(Sumber : <https://www.uml-diagrams.org/state-machine-diagrams.html#behavioral-state-machine>)

2. *Protocol State Machines*

Protocol State Machines merupakan spesialisasi *behavioral state machines* dan digunakan untuk mengekspresikan penggunaan suatu protokol atau penggolongan suatu siklus. Notasi *Protocol State Machines* mirip dengan yang dimiliki oleh *Behavioral State Machines*.



Gambar 2.9

Contoh *Protocol State Machines*

Sumber : (<https://www.uml-diagrams.org/protocol-state-machine-diagrams.html>)

BAB III

ANALISIS DAN PERANCANGAN

3.1 Gambaran Umum

Alat ukur kadar alkohol merupakan alat yang digunakan untuk mengukur kadar alkohol yang beredar dimasyarakat baik minuman alkohol ataupun alkohol lainnya dengan kadar yang berbeda-beda. Alat ini dibuat menggunakan bantuan dari sensor MQ-3 sebagai sensor pendeteksi alkohol yang akan ditetaskan ke kapas kemudian diletakan diatas sensor, kemudian sensor akan membaca berapa persen alkohol tersebut. LCD akan menampilkan hasil dari kadar alkohol tersebut, selain itu juga jika alkohol tidak terdeteksi LCD akan menampilkan bahwa alkohol tidak terdeteksi. Alat ini menggunakan Arduino Uno sebagai pusat kendali dari pengukuran kadar alkohol yang akan dibuat. Arduino sebagai alat bantu kendali pengukuran ini karena Arduino akan membantu menghubungkan perangkat keras dengan perangkat lunak agar hasil dapat diterima secara maksimal.

Cara kerja dari alat kadar ukur alkohol ini adalah sensor menerima masukan dari kapas dengan tetesan alkohol yang ditelakkan diatas sensor. Kemudian Arduino akan memproses sehingga LCD menampilkan data.

3.2 Spesifikasi Kebutuhan

3.2.1 Kebutuhan Perangkat Keras

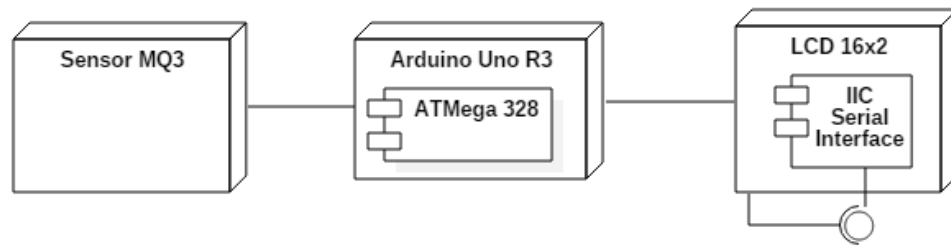
Pada perancangan alat ukur kadar alkohol ini membutuhkan

1. Papan Arduino Uno
2. Sensor MQ-3
3. LCD (*Liquid Crystal Display*) 16 x 2 karakter.

3.2.2 Kebutuhan Perangkat Lunak

Perangkat lunak yang dibutuhkan dalam pembuatan alat ukur kadar alkohol ialah Arduino IDE.

3.3 Diagram Deployment



Gambar 3.1
Diagram *Deployment* Alat Ukur kadar Alkohol

Penjelasan pada Gambar 3.1 sebagai pengukuran kadar alkohol diagram *deployment* yang digunakan sebagai berikut :

1. Sensor MQ-3

Sensor MQ-3 yang akan digunakan sebagai sensor pendeteksi alkohol.

Sensor ini akan mendeteksi alkohol yang diletakkan dengan kapas diatas sensor.

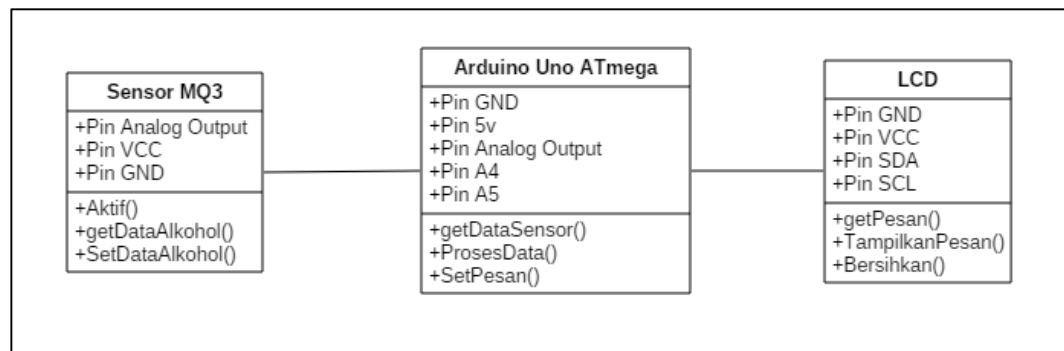
2. Papan Arduino Uno

Papan Arduino Uno yang digunakan menggunakan mikrokontroler ATmega328 yang berfungsi sebagai pusat kendali yang mengolah informasi dari sensor.

3. LCD (*Liquid Crystal Display*)

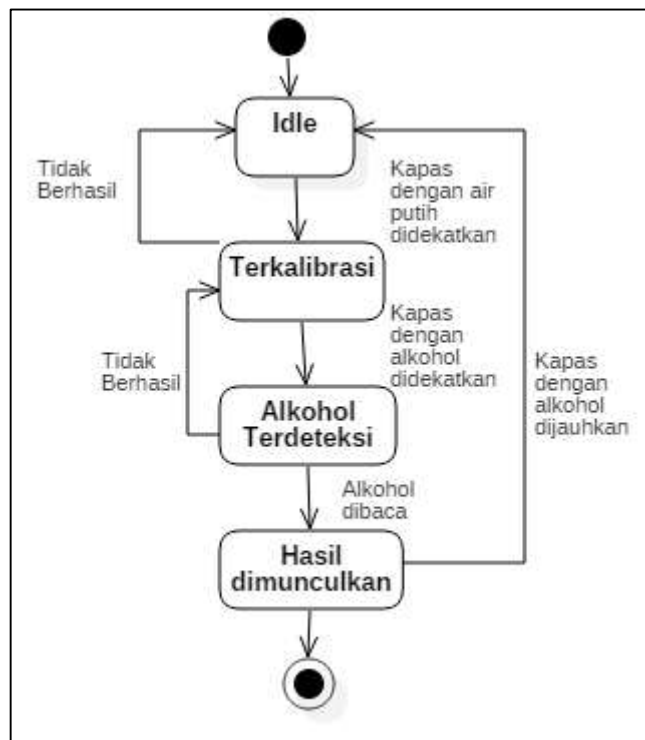
LCD digunakan untuk menampilkan hasil yang didapat dari sensor, yang dimaksud ialah berapa kadar sensor yang terdapat pada alkohol yang ditetes pada kapas.

3.4 Class Diagram



Gambar 3.2
Class Diagram Alat Ukur kadar Alkohol

3.5 State Machine Diagram



Gambar 3.3
Diagram State Machine Alat Ukur kadar Alkohol

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi Perangkat Keras

Pada gambar 4.1 merupakan hasil foto papan Arduino Uno yang sudah dipasangkan di dalam kotak.



Gambar 4.1
Papan Arduino UNO

Pada gambar 4.2 merupakan hasil foto Sensor MQ-3 yang sudah dipasangkan diatas kotak dan telah dipasangkan pada papan Arduino.



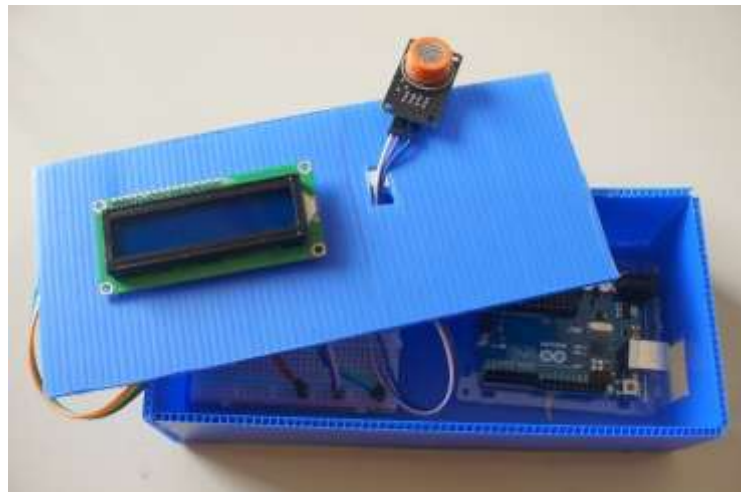
Gambar 4.2
Sensor MQ-3

Pada gambar 4.3 adalah hasil foto LCD (*Liquid Crystal Display*) yang sudah dipasangkan diatas kotak, bersebelahan dengan Sensor MQ-3.



Gambar 4.3
LCD (*Liquid Crystal Display*)

Pada gambar 4.4 merupakan hasil akhir dari alat ukur kadar alkohol berbasis Arduino. Perancangan ini menggunakan papan Arduino UNO, sebuah sensor MQ-3, sebuah LCD dan bantuan *Breadboard* Arduino. Mikrokontroler yang digunakan pada pengujian ini adalah Mikrokontroler ATmega 328P.



Gambar 4.4
Alat ukur kadar alkohol berbasis Arduino

4.2 Implementasi Kalibrasi Sensor MQ-3

Pada sub bab ini yaitu pengujian dan analisa memiliki tujuan yaitu menguji setiap modul yang digunakan untuk perancangan alat ukur kadar alkohol ini. Pengujian ini bertujuan untuk menganalisa kinerja, kesalahan, keakuratan dan keterbatasan dari setiap modul dan sistem yang digunakan. Dari hasil yang didapatkan akan digunakan untuk menarik kesimpulan dan menjadi sebuah acuan untuk menjadi jawaban rumusan masalah yang pada tugas akhir ini.

4.2.1 Pengujian Sensor MQ-3

Pada sub bab ini membahas tentang pengujian modul dari MQ-3 yang digunakan untuk mendeteksi kadar alkohol. Hasil dari kadar alkohol ditampilkan pada serial print Arduino IDE dengan waktu pengukuran kurang lebih 30 detik per pengukuran. Antara sensor dengan kapas yang sudah ditetesi alkohol diberikan jarak 0.5 cm per pengukurannya.

Tabel 4.1
Pengujian Sensor MQ-3

Sample Uji	Hasil Angka Analog	Kadar Alkohol (%)
Air	46	0
Alkohol 95%	929	93
Alkohol 75%	724	74
Alkohol 70%	690	70
Alkohol 60%	576	60
Alkohol 40%	357	40
Alkohol 20%	179	20

4.3 Pengujian dan Analisa Perangkat Keras

4.3.1 Pengujian Sistem dan Analisa Alat Ukur Kadar Alkohol

Pengujian sistem ini bertujuan untuk mengetahui kadar alkohol yang ada dengan alat pemanding adalah alkoholmeter. Pengujian menggunakan kapas yang ditetesi oleh alkohol kemudian diletakkan diatas sensor dengan jarak yang berbeda-beda.

Tabel 4.2
Pengujian Sistem Alat ukur kadar alkohol

SAMPEL UJI	ALKOHOL 95% (ml)	AIR (ml)	JARAK SENSOR DENGAN SAMPEL (cm)	HASIL ALKOHOL METER (%)	HASIL MQ-3(%)	SELISIH	STATUS PADA LCD	KETERANGAN LCD
Alkohol	50	0	0.5	95	93	2	Medis	Sesuai
			1		80	15	Medis	Sesuai
			1.5		38	57	Konsumsi	Sesuai
			2		23	72	Konsumsi dan Bahan bakar	Sesuai
Alkohol + Air	40	10	0.5	78	78	0	Medis	Sesuai
			1		65	13	Medis	Sesuai
			1.5		38	40	Konsumsi	Sesuai
			2		22	56	Konsumsi dan Bahan bakar	Sesuai
Alkohol + Air	30	20	0.5	60	60	0	Medis	Sesuai
			1		55	5	Konsumsi	Sesuai
			1.5		36	24	Konsumsi	Sesuai
			2		26	34	Konsumsi dan Bahan bakar	Sesuai
Alkohol + Air	20	30	0.5	40	40	0	Konsumsi	Sesuai
			1		38	2	Konsumsi	Sesuai
			1.5		8	32	Konsumsi	Sesuai
			2		2	38	Konsumsi	Sesuai
Alkohol + Air	10	40	0.5	20	20	0	Konsumsi dan Bahan bakar	Sesuai
			1		18	2	Konsumsi dan Bahan bakar	Sesuai
			1.5		9	11	Konsumsi	Sesuai
			2		5	15	Konsumsi	Sesuai
Air	0	50	0.5	0	0	0	Tidak ada alkohol	Sesuai
			1		0	0	Tidak ada alkohol	Sesuai
			1.5		0	0	Tidak ada alkohol	Sesuai
			2		0	0	Tidak ada alkohol	Sesuai

Sensor MQ-3 merupakan sensor gas dan pada pengujian ini dilakukan berbagai pengujian dengan jarak antara sensor dengan kapas yang telah ditetesi alkohol. Jaraknya dibagi menjadi 4 yaitu 0.5cm, 1cm, 1.5cm dan 2cm. Lalu, pada LCD menampilkan status dengan kegunaan alkohol yang berbeda-beda sesuai dengan kadar seperti kadar alkohol 56-95% akan menampilkan kegunaan sebagai “Medis” yang artinya pada kadar 56-95% alkohol digunakan pada bidang Kesehatan untuk keperluan medis, kadar alkohol 31-55% akan menampilkan kegunaan sebagai “Konsumsi” yang artinya pada kadar 31-55% alkohol dapat digunakan pada bidang pangan, kadar alkohol 16-30% akan menampilkan kegunaan sebagai “Konsumsi dan Bahan Bakar” yang artinya pada kadar 16-30% alkohol dibagi menjadi 2 kegunaanya pada bidang industri alkohol dapat digunakan sebagai bahan bakar dan pada bidang pangan alkohol dapat berupa minuman beralkohol atau sebagai rum, kadar alkohol 11-15% akan menampilkan kegunaan sebagai “Konsumsi dan Kosmetik” yang artinya pada kadar 11-15% alkohol digunakan pada bidang industri sebagai alat kecantikan yang terdapat kandungan alkoholnya dan pada bidang pangan yang dapat digunakan sebagai minuman beralkohol, kadar alkohol 1-10% akan menampilkan kegunaan sebagai “Konsumsi” yang artinya digunakan pada bidang pangan sebagai minuman beralkohol dan untuk kadar 0% maka akan menampilkan “Tidak ada alkohol” yang artinya tidak mengandung alkohol.

Masing-masing alkohol yang sudah ditetesi pada kapas dan diletakkan diatas sensor dengan jarak seperti tabel 4.2 diukur dengan waktu kurang lebih 30 detik per pengukuran. Selisih pada tabel 4.2 didapatkan dari pengurangan hasil Alkoholmeter dengan hasil yang ditampilkan pada LCD. Sehingga dapat disimpulkan bahwa jarak antara sensor MQ-3 dan alkohol berpengaruh terhadap hasil yang didapatkan karena sensor merupakan sensor gas.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan penyusunan perancangan yang sudah disusun dan dilakukan pengujiannya, maka mendapatkan beberapa kesimpulan, sebagai berikut :

1. Alat Ukur Kadar Alkohol dibuat dengan menggunakan sensor MQ-3 yang menerima data analog yang dikalibrasi dan hasilnya akan ditampilkan pada LCD.
2. Alat Ukur Kadar Alkohol ditampilkan hasilnya pada LCD, dengan baris pertama kadar alkohol dan baris ke dua merupakan kegunaan alkohol yang dapat digunakan dikehidupan sehari-hari.
3. Ketika melakukan pengujian kadar alkohol jarak berpengaruh terhadap hasil, semakin jauh jarak terhadap sensor maka hasil semakin berbeda dengan hasil kadar alkohol dari alkoholmeter. Dengan jarak 0.5 cm maka hasil yang didapat semakin mendekati dengan hasil kadar dari pembanding (alkoholmeter).

5.2 Saran

Pada perancangan Alat Ukur Kadar Alkohol ini masih memiliki banyak kekurangan dan masih jauh dari kata sempurna. Alat Ukur Kadar Alkohol yang dirancang masih membutuhkan bantuan untuk menyempurnakannya dan dikarenakan beberapa hal sehingga penulis memiliki keterbatasan untuk penambahannya, sehingga beberapa saran yang mungkin akan lebih membantu untuk penelitian selanjutnya, sebagai berikut :

1. Pada percobaan sebaiknya dilakukan pada suhu ruangan yang tepat hasil yang didapatkan sesuai.
2. Alkohol yang akan di ukur sebaiknya disimpan dekat sensor agar hasilnya lebih maksimal dan akurat.
3. Sebaiknya menggunakan LED atau *buzzer* sebagai penanda batasan dari kadar alkohol.

DAFTAR PUSTAKA

- Aini, Churrotul, "Pendeteksi Kadar Gas Alkohol Pada Peuyeum Menggunakan Sensor Mq-3 Berbasis Arduino Uno", Program Studi Teknik Fisika, Fakultas Teknologi Industri, Institut Teknologi Bandung
- Akil, Ibnu, 2018, "Referensi dan Panduan UML 2.4 Singkat Tepat Jelas", CV. Garuda Mas Sejahtera.
- Bell, Charles, 2013, "*Beginning Sensor Networks with Arduino and Raspberry Pi.*", Apress.
- Bruice, Paula Yurkanis, 2016, "*Organic Chemistry: Eighth Edition*", Pearson.
- Danin, Veni Jumila dkk, "Manfaat Alkohol Dalam Kehidupan Sehari-Hari", Program Studi Pendidikan Kimia fakultas Sains Dan Teknologi universitas Islam Negeri Sunan Kalijaga Yogyakarta.
- Datasheet I2C LCD.
- Datasheet Mikrokontroler ATmega 328.
- Datasheet Sensor MQ-3.
- Dennis, Alan et.al, 2012, "*System Analysis and Design: Fifth Edition*", John Wiley & Sons, Inc.
- Dharmawan, Hari Arief, 2017, "Mikrokontroler Konsep Dasar dan Praktis", UB Press.
- Evans, Brian. 2011, "*Beginning Arduino Programming*", Apress.
- McRoberts, Michael, 2010, "*Beginning Arduino*", Apress.
- Pawanikar, Nikhil, 2014, "*Introduction to C++ programming*", Institute Of Distance & Open Learning.
- Pemerintah Indonesia. 2013. Peraturan Presiden Republik Indonesia Nomor No 74 Tahun 2013 tentang Pengawasan Dan Pengendalian Minuman Beralkohol No 3. Sekretariat Kabinet RI.
- Podeswa, Howard, 2010, "*UML For The It Business Analyst, Second Edition: A Practical Guide To Requirements Gathering Using The Unified Modeling Language*", Cengage Learning PTR.
- Retnoningsih, Endang, Jafar Shadiq, and Dony Oscar. "Pembelajaran Pemrograman Berorientasi Objek (Object Oriented Programming) Berbasis Project Based Learning." *Informatics for Educators and Professionals*.
- Sakti, Setyawan P, 2017, "Pengantar Teknologi Sensor: Prinsip Dasar Sensor Besaran Mekanik", UB Press.
- Simarmata, Janner, 2010, "Rekaya Perangkat Lunak", C.V ANDI OFFSET.
- Smith, Alan G, 2011, "*Introduction to Arduino : A piece of cake!*", Independent Publishing.
- Sommerville, Ian, 2010, "*Software Engineering: 9th Edition*", Pearson.

Subandi, Ir, 2010, "Kimia Organik", Dee Publish.

Tritama, Topaz Kautsar, "Konsumsi Alkohol dan Pengaruhnya terhadap Kesehatan",
Fakultas Kedokteran, Universitas Lampung

<https://www.teachmicro.com/mq-3-alcohol-sensor/> diakses 25 Oktober 2019, jam 13.38
WIB

<https://teknikelektronika.com/pengertian-sensor-jenis-jenis-sensor/> diakses tanggal 17
Oktober 2019, jam 16.44 WIB

<https://components101.com/microcontrollers/arduino-uno> diakses 14 Oktober 2019, jam
15.18 WIB

<https://www.arduino.cc/> diakses 15 Oktober 2019, jam 23.24 WIB

<https://kelasrobot.com/belajar-pemrograman-dasar-arduino/> diakses 28 Oktober, jam
20.32 WIB

<https://elektronika-dasar.web.id/lcd-liquid-cristal-display/> diakses 14 November jam 11.19
WIB

<https://www.uml-diagrams.org/> diakses 16 Maret 2020 jam 18.30 WIB

<http://saptaji.com/2016/06/27/bekerja-dengan-i2c-lcd-dan-arduino/> diakses 31 Agustus
2020 jam 19.48 WIB

LAMPIRAN

Listing Program Pengukuran Kadar Alkohol

```
#include <LiquidCrystal_I2C.h>
#include <Wire.h>

const int gasSensor    = A0;
int nilaiGasSensor     = 0;
int kalibrasiGasSensor;
int angkaSensor;
int batas ;
int screenWidth = 16;
int screenHeight = 2;
String line1 = "Konsumsi dan Bahan bakar";
String line2 = "Konsumsi dan kosmetik";
String line3 = "Tidak Ada Alkohol";
int stringStart, stringStop = 0;
int scrollCursor = screenWidth;

LiquidCrystal_I2C lcd(0x27,16,2);

void setup()
{
  Serial.begin(9600);
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0,0);
  lcd.print("XENA MARIANA");
  lcd.setCursor(0,1);
  lcd.print("2017130036");
  delay(2000);
  lcd.clear();

  pinMode(nilaiGasSensor,INPUT);
  nilaiGasSensor = analogRead(gasSensor);
  batas = nilaiGasSensor;
}

void loop()
{
  nilaiGasSensor = analogRead(gasSensor);
  kalibrasiGasSensor = map(nilaiGasSensor, batas, 1023, 0, 100);
  angkaSensor = constrain (kalibrasiGasSensor, 0, 100);
  Serial.println(angkaSensor);

  if(angkaSensor>=56)
  {
    lcd.setCursor(0,0);
    lcd.print("Kadar :  %");
    lcd.setCursor(8,0);
    lcd.print(angkaSensor,1);
    lcd.setCursor(0, 1);
    lcd.print("Medis");
    delay(500);
    lcd.clear();
  }
  else if(angkaSensor>=31)
  {
    lcd.setCursor(0,0);
    lcd.print("Kadar :  %");
    lcd.setCursor(8,0);
    lcd.print(angkaSensor,1);
    lcd.setCursor(0, 1);
    lcd.print("Konsumsi");
    delay(500);
    lcd.clear();
  }
}
```

```

else if(angkaSensor>=16)
{
    lcd.setCursor(0,0);
    lcd.print("Kadar :  %");
    lcd.setCursor(8,0);
    lcd.print(angkaSensor,1);
    lcd.setCursor(scrollCursor, 1);
    lcd.print(line1.substring(stringStart,stringStop));
    delay(500);
//agar text berjalan
if(stringStart == 0 && scrollCursor > 0)
{
    scrollCursor--;
    stringStop++;
}
else if (stringStart == stringStop)
{
    stringStart = stringStop = 0;
    scrollCursor = screenWidth;
}
else if (stringStop == line1.length() && scrollCursor == 0)
{
    stringStart++;
}
else
{
    stringStart++;
    stringStop++;
}
}
lcd.clear();
}
else if(angkaSensor>=11)
{
    lcd.setCursor(0,0);
    lcd.print("Kadar :  %");
    lcd.setCursor(8,0);
    lcd.print(angkaSensor,1);
    lcd.setCursor(scrollCursor, 1);
    lcd.print(line2.substring(stringStart,stringStop));
    delay(500);
//agar text berjalan
if(stringStart == 0 && scrollCursor > 0)
{
    scrollCursor--;
    stringStop++;
}
else if (stringStart == stringStop){
    stringStart = stringStop = 0;
    scrollCursor = screenWidth;
}
else if (stringStop == line1.length() && scrollCursor == 0) {
    stringStart++;
}
else
{
    stringStart++;
    stringStop++;
}
}
lcd.clear();
}
else if(angkaSensor>=1)
{
    lcd.setCursor(0,0);
    lcd.print("Kadar :  %");
    lcd.setCursor(8,0);
    lcd.print(angkaSensor,1);
    lcd.setCursor(0, 1);
    lcd.print("Konsumsi");
}

```

```

delay(500);
  lcd.clear();
}
else
{
  lcd.setCursor(0,0);
  lcd.print("Kadar : %");
  lcd.setCursor(8,0);
  lcd.print(angkaSensor,1);
  lcd.setCursor(scrollCursor, 1);
  lcd.print(line3.substring(stringStart,stringStop));
  delay(500);
//agar text berjalan
  if(stringStart == 0 && scrollCursor > 0)
  {
    scrollCursor--;
    stringStop++;
  }
  else if (stringStart == stringStop){
    stringStart = stringStop = 0;
    scrollCursor = screenWidth;
  }
  else if (stringStop == line1.length() && scrollCursor == 0)
  {
    stringStart++;
  }
  else
  {
    stringStart++;
    stringStop++;
  }
  lcd.clear();
}
}

```

Arduino UNO



Product Overview

The Arduino Uno is a microcontroller board based on the ATmega328 ([datasheet](#)). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the [index of Arduino boards](#).

Index

Technical
Specifications

Page 2

How to use Arduino
Programming Environment, Basic Tutorials

Page 6

Terms &
Conditions

Page 7

Enviromental Policies
half sqm of green via Impatto Zero®

Page 7

Technical Specification

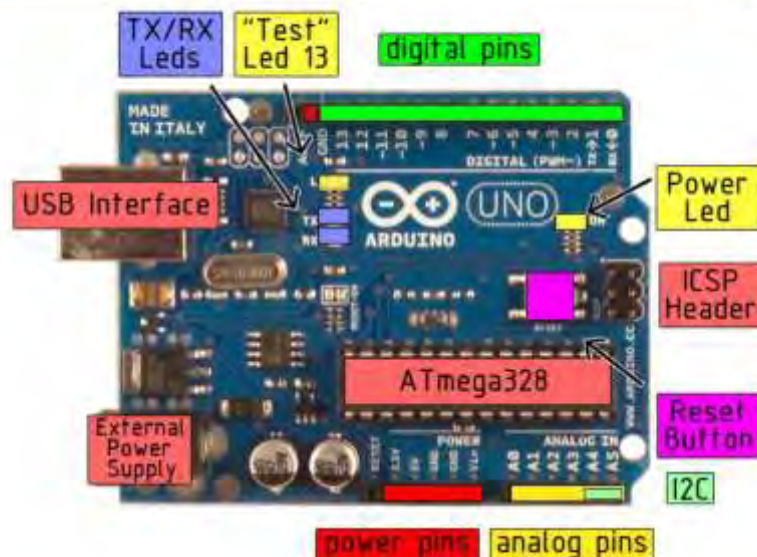


EAGLE files: [arduino-duemilanove-uno-design.zip](#) Schematic: [arduino-uno-schematic.pdf](#)

Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

the board



Power

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

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

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

The power pins are as follows:

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

Memory

The Atmega328 has 32 KB of flash memory for storing code (of which 0,5 KB is used for the bootloader); It has also 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip .
- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.



radiospares

RADIONICS



The Uno has 6 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the [analogReference\(\)](#) function. Additionally, some pins have specialized functionality:

- **I²C: 4 (SDA) and 5 (SCL).** Support I²C (TWI) communication using the

[Wire library](#). There are a couple of other pins on the board:

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

See also the [mapping between Arduino pins and Atmega328 ports](#).

Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega8U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '8U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, an *.inf file is required..

The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Uno's digital pins.

The ATmega328 also support I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the [documentation](#) for details. To use the SPI communication, please see the ATmega328 datasheet.

Programming

The Arduino Uno can be programmed with the Arduino software ([download](#)). Select "Arduino Uno w/ ATmega328" from the **Tools > Board** menu (according to the microcontroller on your board). For details, see the [reference](#) and [tutorials](#).

The ATmega328 on the Arduino Uno comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

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

The ATmega8U2 firmware source code is available . The ATmega8U2 is loaded with a DFU bootloader, which can be activated by connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2. You can then use [Atmel's FLIP software](#) (Windows) or the [DFU programmer](#) (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader).



radiospares

RADIONICS



Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

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

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

USB Overcurrent Protection

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

Physical Characteristics

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the

board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.



radiospares

RADIONICS



How to use Arduino



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

Arduino is a cross-platform program. You'll have to follow different instructions for your personal OS. Check on the [Arduino site](http://arduino.cc/en/Guide/HomePage) for the latest instructions. <http://arduino.cc/en/Guide/HomePage>

Linux Install

Windows Install

Mac Install

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

Blink led

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

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

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

In **Tools>Board** select

Now you have to go to



Tools>SerialPort

and select the right serial port,
the one arduino is attached to.

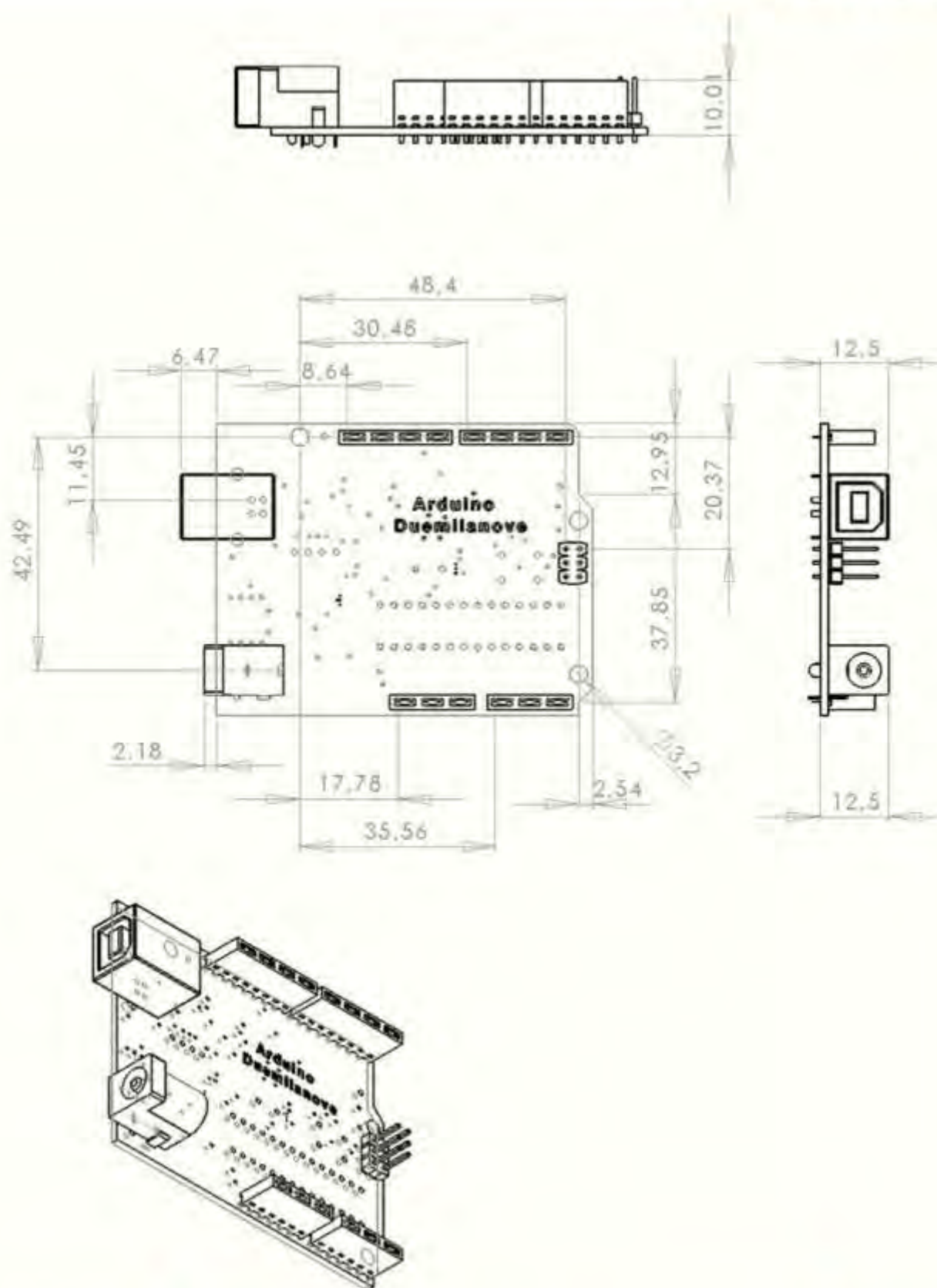


radiospares

RADIONICS



Dimensioned Drawing



Terms & Conditions



1. Warranties

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

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

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

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

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

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

2. Indemnification

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

3. Consequential Damages Waiver

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

4. Changes to specifications

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



Enviromental Policies



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



radiospares

RADIONICS



TECHNICAL DATA MQ-3 GAS SENSOR

FEATURES

- * High sensitivity to alcohol and small sensitivity to Benzine .
- * Fast response and High sensitivity
- * Stable and long life
- * Simple drive circuit

APPLICATION SPECIFICATIONS

They are suitable for alcohol checker, Breathalyser.

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	200KΩ	
R _H	Heater resistance	33Ω±5%	Room Tem
P _H	Heating consumption	less than 750mw	

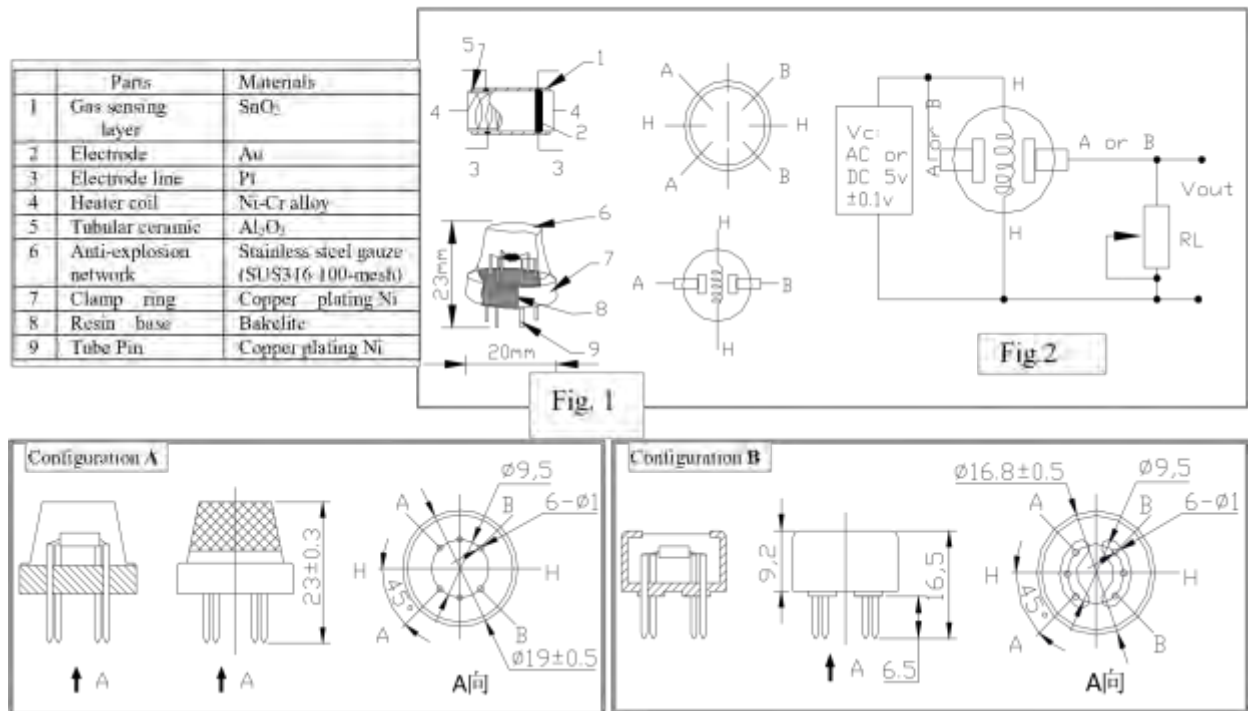
B. Environment condition

Symbol	Parameter name	Technical condition	Remarks
T _{ao}	Using Tem	-10°C-50°C	
T _{as}	Storage Tem	-20°C-70°C	
R _H	Related humidity	less than 95%Rh	
O ₂	Oxygen concentration	21%(standard condition)Oxygen concentration can affect sensitivity	minimum value is over 2%

C. Sensitivity characteristic

Symbol	Parameter name	Technical parameter	Remarks
Rs	Sensing Resistance	1MΩ- 8 MΩ (0.4mg/L alcohol)	Detecting concentration scope : 0.05mg/L—10mg/L Alcohol
α (0.4/1 mg/L)	Concentration slope rate	≤0.6	
Standard detecting condition	Temp: 20°C±2°C Vc:5V±0.1 Humidity: 65%±5% Vh: 5V±0.1		
Preheat time	Over 24 hour		

D. Structure and configuration, basic measuring circuit



TEL: 86-371- 67169070 67169080 FAX: 86-371-67169090
sales@hwsensor.com

E-mail:

HANWEI ELETRONICS CO.,LTD MQ-3

<http://www.hwsensor.com>

Structure and configuration of MQ-3 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-3 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

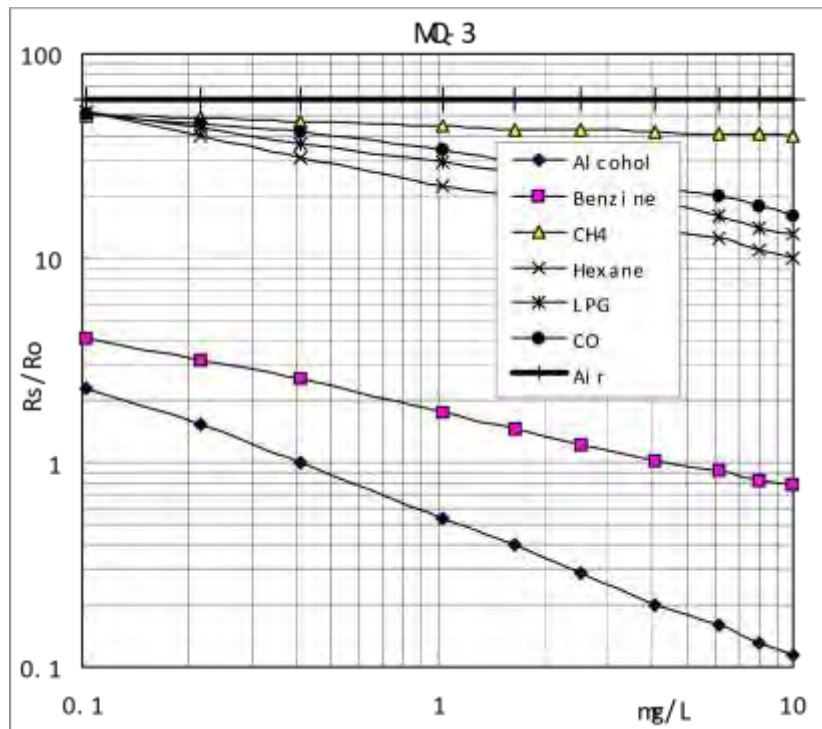


Fig.2 sensitivity characteristics of the MQ-3

Fig.3 is shows the typical sensitivity characteristics of the MQ-3 for several gases.

in their: Temp: 20°C、

Humidity: 65%、

O₂ concentration 21%

RL=200k Ω

Ro: sensor resistance at 0.4mg/L of Alcohol in the clean air.

Rs:sensor resistance at various concentrations of gases.

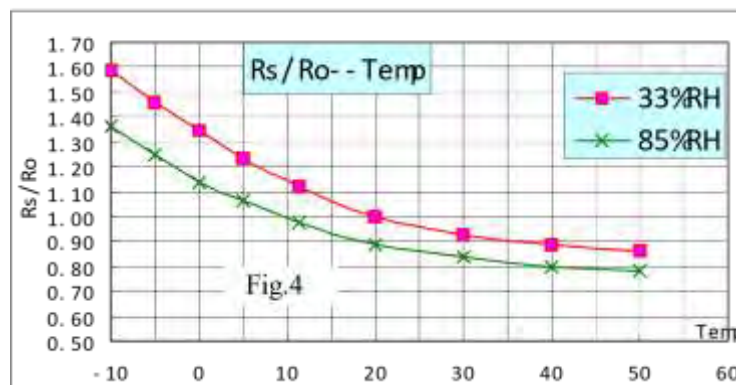


Fig.4 is shows the typical dependence of the MQ-3 on temperature and humidity.

Ro: sensor resistance at 0.4mg/L of Alcohol in air at 33%RH and 20 °C

Rs: sensor resistance at 0.4mg/L of

Alcohol at different temperatures and humidities.

SENSITIVITY ADJUSTMENT

Resistance value of MQ-3 is difference to various kinds and various concentration gases. So, When using this components, sensitivity adjustment is very necessary. we recommend that you calibrate the detector for 0.4mg/L (approximately 200ppm) of Alcohol concentration in air and use value of Load resistance that (R_L) about 200 KΩ (100KΩ to 470 KΩ).

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

TEL: 86-371- 67169070 67169080 FAX: 86-371-67169090
sales@hwsensor.com

E-mail:

Datasheet

I2C 1602 Serial LCD Module



Product features:

The I2C 1602 LCD module is a 2 line by 16 character display interfaced to an I2C daughter board. The I2C interface only requires 2 data connections, +5 VDC and GND to operate

For in depth information on I2C interface and history, visit: <http://www.wikipedia/wiki/I2c>

Specifications:

I2C Address Range	2 lines by 16 character 0x20 to 0x27 (Default=0x27, addressable)
Operating Voltage	5 Vdc
Backlight	White
Contrast	Adjustable by potentiometer on I2c
Size	interface 80mm x 36mm x 20 mm 66mm x
Viewable area	16mm

Power:

The device is powered by a single 5Vdc connection.