

BAB II

TINJAUAN PUSTAKA

2.1 Sensor Ultrasonik

2.1.1 Pengertian Sensor Ultrasonik

alat elektronika yang kemampuannya bisa mengubah dari energi listrik menjadi energy mekanik dalam bentuk gelombang suara ultrasonic. Sensor ini terdiri dari rangkaian pemancar Ultrasonik yang dinamakan transmitter dan penerima ultrasonic yang disebut receiver. Alat ini digunakan untuk mengukur gelombang ultrasonik. Gelombang ultrasonic adalah gelombang mekanik yang memiliki cirri-ciri longitudinal dan biasanya memiliki frekuensi di atas 20 Khz. Gelombang Utrasonik dapat merambat melalui zat padat, cair maupun gas. Gelombang ultrasonik adalah gelombang rambatan energi dan momentum mekanik sehingga merambat melalui ketiga element tersebut sebagai interaksi dengan molekul dan sifat enersia medium yang dilaluinya.

Gelombang ultrasonik merambat melalui udara dengan kecepatan 344 meter per detik, mengenai obyek dan memantul kembali ke sensor ultrasonik. Seperti yang telah umum diketahui, gelombang ultrasonik hanya bisa didengar oleh makhluk tertentu seperti kelelawar dan ikan paus. Kelelawar menggunakan gelombang ultrasonic untuk berburu di malam hari sementara paus menggunakannya untuk berenang di kedalaman laut yang gelap.

Perhitungan waktu yang diperlukan modul sensor Ping untuk menerima pantulan pada jarak tertentu mempunyai rumus $S = (\Delta t \times V) : 2$. Rumus diatas mempunyai keterangan sebagai berikut. (S) adalah jarak antara sensor ultrasonik dengan obyek yang terdeteksi. (V) adalah cepat rambat gelombang ultrasonik di udara dengan kecepatan normal (344 meter per detik) (tIN) adalah selisih waktu pemancaran dan penerimaan pantulan gelombang. Ada 3 prnsip kerja dari sensor

ultrasonik yaitu, sinyal dipancarkan melalui pemancar gelombang ultrasonic. Sinyal yang dipancarkan akan merambat sebagai gelombang bunyi dengan kecepatan bunyi berkisar 344 m/s. Dan yang terakhir sinyal yang sudah diterima akan diproses untuk menghitung jaraknya.

2.1.2 Jenis –Jenis Sensor Ultrasonik

2.1.2.1 Sensor PING

Sensor PING merupakan sensor ultrasonik yang dapat mendeteksi jarak obyek dengan cara memancarkan gelombang ultrasonik dengan frekuensi 40 KHz dan kemudian mendeteksi pantulannya.

Sensor ini dapat mengukur jarak antara 3 cm sampai 300 cm. keluaran dari sensor ini berupa pulsa yang lebarnya merepresentasikan jarak. Lebar pulsanya bervariasi dari 115 uS sampai 18,5 mS. Pada dasarnya, Ping terdiri dari sebuah chip pembangkit sinyal 40KHz, sebuah speaker ultrasonik dan sebuah mikropon ultrasonik. Speaker ultrasonik mengubah sinyal 40 KHz menjadi suara sementara mikropon ultrasonik berfungsi untuk mendeteksi pantulan suaranya.

Pin signal dapat langsung dihubungkan dengan mikrokontroler tanpa tambahan komponen apapun. Ping hanya akan mengirimkan suara ultrasonik ketika ada pulsa trigger dari mikrokontroler (Pulsa high selama 5uS). Suara ultrasonik dengan frekuensi sebesar 40KHz akan dipancarkan selama 200uS. Suara ini akan merambat di udara dengan kecepatan 344.424m/detik (atau 1cm setiap 29.034uS), mengenai objek untuk kemudian terpantul kembali ke Ping. Selama menunggu pantulan, Ping akan menghasilkan sebuah pulsa. Pulsa ini akan berhenti (low) ketika suara pantulan terdeteksi oleh Ping. Oleh karena itulah lebar pulsa tersebut dapat merepresentasikan jarak antara Ping dengan objek.



Gambar 2.1 Sensor PING

Sumber: nurcholifah41,2014

2.1.2.2 Sensor Ultrasonik Devantech SRF04

Sensor Devantech SRF04 memiliki range finder sekitar 3 cm - 3m. Kit ini sangat mudah untuk dirangkai dan membutuhkan sumber daya yang kecil sekali, yang sangat ideal untuk aplikasi mobil robot pencari jarak ini bekerja dengan cara memancarkan pulsa suara dengan kecepatan suara (0,9 ft/milidetik).



Gambar 2.2 Sensor Ultrasonik Devantech SRF04

Sumber: nurcholifah41,2014

2.2 Mikrokontroler

2.2.1 Pengertian Mikrokontroller

Selama 40 tahun sejak pertama kali diperkenalkan, mikrokontroller telah mengalami banyak perkembangan. Berbagai teknologi, fungsi, serta periferal yang diterapkan pada komponen ini menjadikan mikrokontroller yang saat ini beredar memiliki banyak variasi. Sudah tak terhitung pula aplikasi yang dibuat menggunakan mikrokontroller, mulai dari untuk kehidupan sehari-hari hingga skala industri.

Mikrokontroler adalah komputer mikro dalam satu chip tunggal. Mikrokontroler memadukan CPU, ROM, RWM, I/O paralel, I/O seri, *counter-timer*, dan rangkaian *clock* dalam satu chip. Dengan kata lain, mikrokontroler adalah suatu alat elektronika digital yang mempunyai masukan dan keluaran serta kendali dengan program yang bisa ditulis dan dihapus dengan cara khusus. Cara kerja mikrokontroler sebenarnya membaca dan menulis data. (Kaka, 2011)

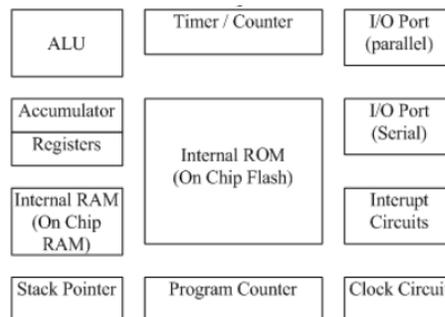
Sama halnya dengan *mikroprocessor*, mikrokontroler adalah piranti yang dirancang untuk kebutuhan umum. Fungsi utama dari mikrokontroler adalah mengontrol kerja mesin atau sistem menggunakan program yang disimpan pada sebuah ROM. Mikrokontroler merupakan komputer didalam chip yang digunakan untuk mengontrol peralatan elektronik, yang menekankan efisiensi dan efektifitas biaya. Secara harfiah dapat disebut sebagai “pengendali kecil” dimana sebuah sistem elektronik yang sebelumnya banyak memerlukan komponen-komponen pendukung seperti IC TTL dan CMOS dapat direduksi/diperkecil dan akhirnya terpusat serta dikendalikan oleh mikrokontroler ini.

Mikrokontroler digunakan dalam produk dan alat yang dikendalikan secara otomatis, seperti sistem kontrol mesin, *remote control*, mesin kantor, peralatan rumah tangga, alat berat, dan mainan. Dengan mengurangi ukuran, biaya, dan konsumsi tenaga dibandingkan desain menggunakan mikroprosesor memori dan alat input output yang terpisah, kehadiran *mikrokontroler* membuat kontrol elektrik untuk berbagai proses menjadi lebih ekonomis. Dengan penggunaan *mikrokontroler* ini maka:

1. Sistem elektronik akan menjadi lebih ringkas,
2. Rancang bangun sistem elektronik dapat dilakukan lebih cepat karena sebagian besar sistem merupakan perangkat lunak yang mudah dimodifikasi,
3. Gangguan yang terjadi lebih mudah ditelusuri karena sistemnya yang kompak.

Mikrokontroler tidak sepenuhnya dapat mereduksi komponen IC TTL dan CMOS yang seringkali masih diperlukan untuk aplikasi kecepatan tinggi atau sekedar menambah jumlah saluran masukan dan keluaran (I/O). Dengan kata lain, mikrokontroler adalah versi mini atau mikro dari sebuah komputer karena mikrokontroler telah mengandung beberapa periferal yang langsung bisa dimanfaatkan, misalnya port paralel, port serial, komparator, konversi digital ke analog (DAC), konversi analog ke digital dan sebagainya hanya menggunakan sistem minimum yang sederhana.

Agar sebuah mikrokontroler dapat berfungsi, maka mikrokontroler tersebut memerlukan komponen eksternal yang kemudian disebut dengan sistem minimum. Untuk membuat sistem minimum paling tidak dibutuhkan sistem *clock* dan *reset*, walaupun pada beberapa mikrokontroler sudah menyediakan sistem *clock* internal, sehingga tanpa rangkaian eksternal pun mikrokontroler dapat beroperasi. Untuk merancang sebuah sistem berbasis mikrokontroler, kita memerlukan perangkat keras dan perangkat lunak, yaitu sistem minimum mikrokontroler, *software* pemrograman dan *compiler*, serta *downloader*.



Gambar 2.3 Blok Diagram Mikrokontroler Secara Umum

Sumber : Kaka, 2011

Dalam gambar 2.3 terlihat bahwa sebuah mikrokontroler terdiri dari beberapa bagian. Bagian-bagian tersebut saling dihubungkan dengan internal dan pada

umumnya terdiri dari 3 macam bus yaitu *address bus*, *data bus* dan *control bus*. Masing-masing bagian memiliki fungsi-fungsi sebagai berikut:

1. Register

Register merupakan suatu tempat penyimpanan (variabel) bilangan bulat yang terdiri dari 8 atau 16 bit. Pada umumnya register memiliki jumlah yang banyak, masing-masing ada yang memiliki fungsi khusus dan ada pula yang memiliki fungsi atau kegunaan secara umum. Register yang memiliki fungsi secara khusus misalnya register *timer* yang berisi data penghitungan pulsa untuk *timer*, atau register pengatur mode operasi *counter* (penghitung pulsa). Sedangkan register yang memiliki fungsi umum digunakan untuk menyimpan data sementara yang diperlukan untuk proses penghitungan dan proses operasi mikrokontroler. Register dengan fungsi umum sangat dibutuhkan dalam sistem mikrokontroler karena mikrokontroler hanya mampu melakukan operasi aritmetik atau logika hanya pada satu atau dua operasi saja, sehingga untuk operasi-operasi yang melibatkan banyak variabel harus dimanipulasi dengan menggunakan variabel-variabel register umum.

2. Accumulator

Accumulator merupakan salah satu register khusus yang berfungsi sebagai operasi umum proses aritmetika dan logika.

3. Program Counter

Program *counter* merupakan salah satu register khusus yang berfungsi sebagai pencacah atau penghitung eksekusi program mikrokontroler.

4. ALU (*Arithmetic and Logic Unit*)

ALU memiliki kemampuan dalam mengerjakan proses-proses aritmatika (penjumlahan, pengurangan, perkalian, pembagian) dan operasi logika (AND, OR, XOR, NOT) terhadap bilangan bulat 8 atau 16 bit.

5. *Clock Circuits*

Mikrokontroller merupakan rangkaian logika sekuensial, dimana proses kerjanya berjalan melalui sinkronisasi *clock*. Oleh karena itu, diperlukan *clock circuits* yang menyediakan *clock* untuk seluruh bagian rangkaian.

6. *Internal ROM (Read Only Memory)*

Internal ROM (*Read Only Memory*) merupakan memori penyimpanan data dimana data tersebut tidak dapat diubah atau dihapus (hanya dapat dibaca). ROM biasanya diisi dengan program untuk dijalankan oleh mikrokontroller segera setelah power dihidupkan. Data dalam ROM tidak dapat hilang meskipun power dimatikan.

7. *Stack Pointer*

Stack pointer merupakan bagian dari RAM yang memiliki metode penyimpanan dan pengambilan data secara khusus. Data yang disimpan dan dibaca tidak dapat dilakukan dengan cara acak karena data yang dituliskan ke dalam *stack* yang berada pada urutan yang terakhir merupakan data yang pertama kali dibaca kembali. *Stack pointer* berisi *offset* dimana posisi data *stack* yang terakhir masuk atau yang pertama kali dapat diambil.

8. *I/O (Input/Output) Ports*

I/O (Input/Output) ports merupakan sarana yang digunakan oleh mikrokontroller untuk mengakses peralatan-peralatan lain di luar sistem. *I/O port* berupa pin-pin yang dapat berfungsi untuk mengeluarkan data digital ataupun sebagai masukan data eksternal.

9. *Interrupt Circuits*

Interrupt circuits merupakan rangkaian yang memiliki fungsi untuk mengendalikan sinyal-sinyal interupsi baik internal maupun eksternal. Adanya sinyal interupsi akan menghentikan eksekusi normal program mikrokontroler untuk selanjutnya menjalankan sub-program untuk melayani interupsi tersebut.

Diagram blok di atas tidak selalu sama untuk setiap jenis mikrokontroler. Beberapa mikrokontroler menyertakan rangkaian ADC (*Analog to Digital Converter*) di dalamnya, ada pula yang menyertakan *port I/O* serial disamping *port I/O* paralel yang sudah ada.

10. *Internal RAM (Random Acces Memory)*

Internal RAM (*Random Acces Memory*) merupakan memori penyimpanan data dimana data tersebut dapat diubah atau dihapus. RAM biasanya berisi data-data variabel dan register. Data yang tersimpan pada RAM bersifat *volatile* yaitu akan hilang bila catu daya yang terhubung padanya dimatikan.

2.2.2 Sistem Mikrokontroler

Mikroprosesor dan mikrokontroler berasal dari ide dasar yang sama. Mikroprosesor adalah istilah yang merujuk pada *central processing unit* (CPU) *computer digital* untuk tujuan umum. Untuk membuat sistem komputer, CPU harus ditambahkan memori, umumnya *read only memory* (ROM) dan *random access memory* (RAM), dekoder memori, osilator dan sejumlah *input/output device* seperti port data paralel dan serial. (Syeptianda, 2011).

Mikrokontroler umumnya dikelompokkan dalam satu keluarga besar, contoh-contoh keluarga mikrokontroler yaitu :

1. Keluarga MCS-51
2. Keluarga MC68HC05

3. Keluarga MC68HC11
4. Keluarga AVR
5. Keluarga PIC8

2.3 MikroKontroler Atmega 16

2.3.1 Pengertian Mikrokontroler Atmega16

Mirkokontroler AVR (Alf and Vegard's Risc processor) standar memiliki arsitektur 8 bit, dimana semua instruksi dikemas dalam kode 16-bit, dan sebagian besar intruksi dieksekusi dalam 1(satu) siklus clock. AVR berteknologi RISC (Reduced Instruction Set Computing), sedangkan MCS51 berteknologi CISC (Complex Intruction Set Computing).

AVR dapat dikelompokkan menjadi empat kelas, yaitu keluarga Attiny, keluarga AT902xx, keluarga Atmega, dan keluarga AT86RFxx. Pada dasarnya yang membedakan masing-masing kelas adalah memori, peripheral, dan fungsinya. Silahkan buka www.atmel.com untuk informasi lebih lanjut tentang berbagai variasi AVR. Untuk mikrokontroler AVR yang berukuran lebih kecil, silahkan mencoba Atmega8, Attiny2313 dengan ukuran Flash Memory 2KB dengan dua input analog.

Mikrokontroler pada dasarnya diprogram dengan bahasa assembler. Tetapi Saat ini mikrokontroler dapat deprogram dengan menggunakan bahasa tingkat tinggi seperti BASIC, PASCAL atau C.

Bahasa tingkat tinggi tersebut memiliki beberapa keuntungan dibandingkan dengan bahasa assembler:

1. Lebih mudah membangun program dengan menggunakan bahasa tingkat tinggi
2. Perbaikan program lebih mudah jika program dibangun menggunakan bahasa tingkat tinggi

3. Testing program didalam bahasa tingkat tinggi lebih mudah
4. Bahasa tingkat tinggi lebih banyak dikenal dan error program yang dibuat dapat dihindari
5. Mudah mendokumentasikan sebuah program tingkat tinggi

Meskipun demikian, bahasa tingkat tinggi juga memiliki beberapa kelemahan, contohnya ukuran kode memori biasanya besar, dan program yang dibangun menggunakan bahasa assembler biasanya bekerja cepat dibandingkan dengan program yang dibangun menggunakan bahasa tingkat tinggi.

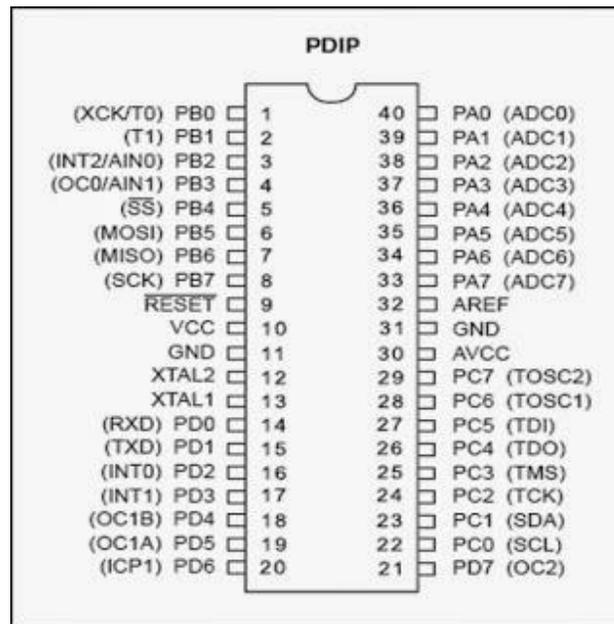
2.3.2 Didalam mikrokontroler Atmega16 terdiri dari:

1. Saluran I/O ada 32 buah, yaitu Port A, Port B, Port C, dan Port D.
2. ADC (Analog to Digital Converter) 10 bit sebanyak 8 channel.
3. Tiga buah Timer/Counter dengan kemampuan perbandingan.
4. CPU yang terdiri dari 32 register.
5. 131 intruksi andal yang umumnya hanya membutuhkan 1 siklus clock.
6. Watchdog Timer dengan oscilator internal.
7. Dua buah Timer/Counter 8 bit.
8. Satu buah Timer /Counter 16 bit.
9. Tagangan operasi 2.7 V - 5.5 V pada Atmega16.
10. Internal SRAM sebesar 1KB.
11. Memory Flash sebesar 16KB dengan kemampuan Read While Write.
12. Unit interupsi internal dan eksternal.
13. Port antarmuka SPI.
14. EEPROM sebesar 512 byte dapat diprogram saat operasi.
15. Antar muka komparator analog.
16. 4 channel PWM.
17. 32x8 general purpose register.

18. Hampir mencapai 16 MIPS pada Kristal 16 MHz.
19. Port USART programmable untuk komunikasi serial.

2.3.3 Konfigurasi Pin ATmega 16

Atmega 16 mempunyai kaki standart 40 pin PDIP yang mempunyai fungsi sendiri-sendiri. Untuk lebih jelas tentang konfigurasi Pin Atmega 16 bisa di lihat pada gambar berikut.



Gambar 2.4 Susunan Kaki Standar 40 Pin Mikrokontroler AVR Atmega16.
Sumber: kaka, 2013

Berikut penjelasan umum susunan kaki Atmega16 tersebut:

1. VCC merupakan pin masukan positif catudaya. Setiap peralatan elektronika digital tentunya butuh sumber catu daya yang umumnya sebesar 5 V, itulah sebabnya di PCB kit rangkaian mikrokontroler selalu dipasang IC regulator 7805.
2. GND sebagai PIN ground.
3. Port A (PA0 ... PA7) merupakan pin I/O dua arah dan dapat diprogram sebagai pin masukan ADC.
4. Port B (PB0 ... PB7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu Timer/Counter, Komparator Analog, dan SPI.

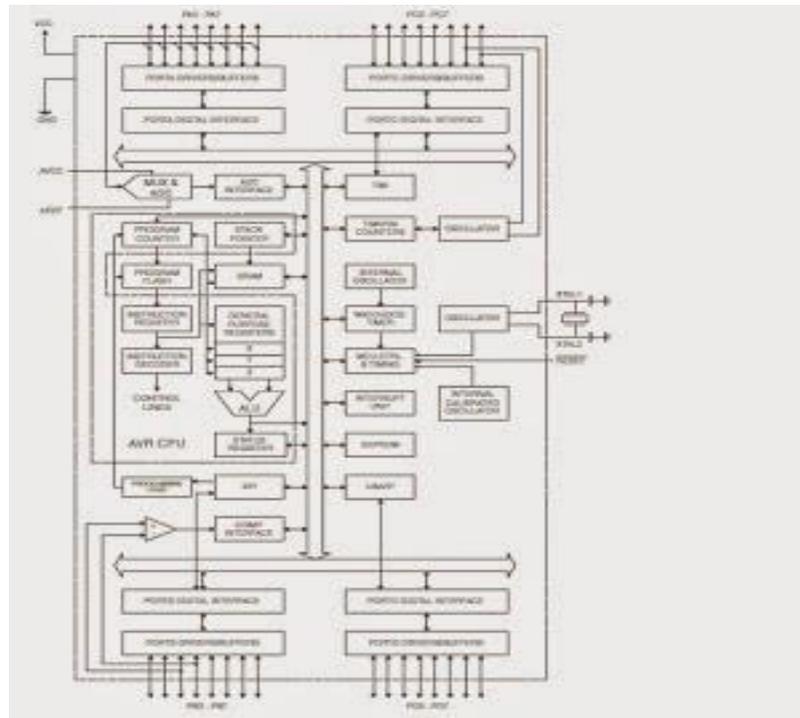
5. Port C (PC0 ... PC7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu TWI, komparator analog, dan Timer Oscilator.
6. Port D (PD0 ... PD7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu komparator analog, interupsi eksternal, dan komunikasi serial.
7. Reset merupakan pin yang digunakan untuk me-reset mikrokontroler ke kondisi semula.
8. XTAL 1 dan XTAL 2 sebagai pin masukan clock eksternal. Suatu mikrokontroler membutuhkan sumber detak (clock) agar dapat mengeksekusi intruksi yang ada di memori. Semakin tinggi nilai kristalnya, maka semakin cepat pula mikrokontroler tersebut dalam mengeksekusi program.
9. AVCC sebagai pin masukan tegangan untuk ADC.
10. AREF sebagai pin masukan tegangan referensi.

2.3.4 Port sebagai input/output digital

ATMega16 mempunyai empat buah port yang bernama PortA, PortB, PortC, dan PortD. Keempat port tersebut merupakan jalur bidirectional dengan pilihan internal pull-up. Tiap port mempunyai tiga buah register bit, yaitu DDxn, PORTxn, dan PINxn. Huruf 'x' mewakili nama huruf dari port sedangkan huruf 'n' mewakili nomor bit. Bit DDxn terdapat pada I/O address DDRx, bit PORTxn terdapat pada I/O address PORTx, dan bit PINxn terdapat pada I/O address PINx. Bit DDxn dalam register DDRx (Data Direction Register) menentukan arah pin. Bila DDxn diset 1 maka Px berfungsi sebagai pin output. Bila DDxn diset 0 maka Px berfungsi sebagai pin input. Bila PORTxn diset 1 pada saat pin terkonfigurasi sebagai pin input, maka resistor pull-up akan diaktifkan. Untuk mematikan resistor pull-up, PORTxn harus diset 0 atau pin dikonfigurasi sebagai pin output. Pin port adalah tri-state setelah kondisi reset.

Bila PORTxn diset 1 pada saat pin terkonfigurasi sebagai pin output maka pin

port akan berlogika 1. Dan bila PORT_{xn} diset 0 pada saat pin terkonfigurasi sebagai pin output maka pin port akan berlogika 0. Saat mengubah kondisi port dari kondisi tri-state (DD_{xn}=0, PORT_{xn}=0) ke kondisi output high (DD_{xn}=1, PORT_{xn}=1) maka harus ada kondisi peralihan apakah itu kondisi pull-up enabled (DD_{xn}=0, PORT_{xn}=1) atau kondisi output low (DD_{xn}=1, PORT_{xn}=0). Biasanya, kondisi pull-up enabled dapat diterima sepenuhnya, selama lingkungan impedansi tinggi tidak memperhatikan perbedaan antara sebuah strong high driver dengan sebuah pull-up. Untuk lebih jelasnya, dapat dilihat pada gambar berikut.

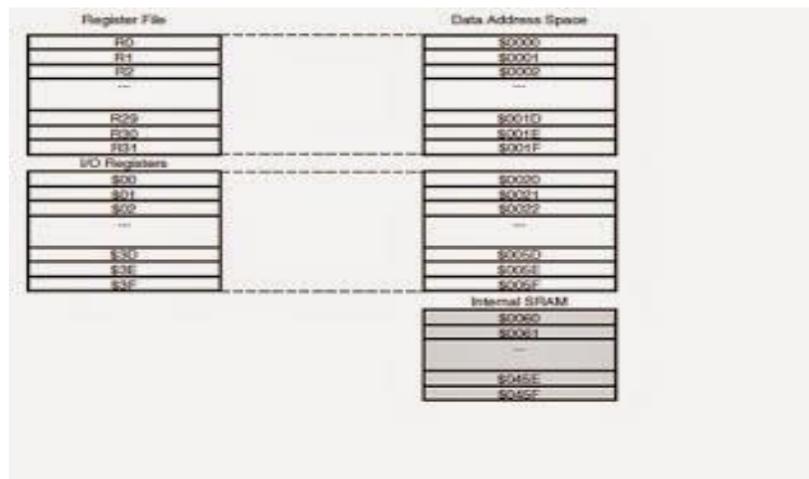


Gambar 2.5 Susunan Port Mikrokontroler Atmega 16
 Sumber: kaka, 2013

2.3.5 PetaMemori

AVR ATmega16 memiliki ruang pengalamatan memori data dan memori program yang terpisah. Memori data terbagi menjadi 3 bagian, yaitu 32 buah register umum, 64 buah register I/O, dan 1kb SRAM internal.

Register keperluan umum menempati space data pada alamat terbawah, yaitu \$00 sampai \$1F. Sementara itu, register khusus untuk menangani I/O dan kontrol terhadap mikrokontroler menempati 64 alamat berikutnya, yaitu mulai dari \$20 hingga \$5F. Register tersebut merupakan register yang khusus digunakan untuk mengatur fungsi terhadap berbagai peripheral mikrokontroler, seperti kontrol register, timer/counter, fungsi – fungsi I/O, dan sebagainya. Alamat memori berikutnya yang digunakan untuk SRAM 1kb, yaitu pada lokasi \$60 sampai dengan \$45F. Konfigurasi memori data ditunjukkan pada gambar berikut.



Gambar 2.6 Konfigurasi Memori Data

Sumber : kaka , 2013

Mikrokontroler disini digunakan sebagai komunikasi antara computer dengan Plant, dimana digunakan komunikasi serial RS232 sebagai komunikasi antara Mikrokontroler dengan Komputer. Sumber (Wardhana.L ,2006).

1. Pulse Width Modulation (PWM)

PWM atau modulasi lebar pulsa adalah salah satu keunggulan dari Timer/Counter yang terdapat pada ATmega16. Ketiga jenis Timer/Counter pada ATmega32 dapat menghasilkan pulsa PWM. Pulsa PWM adalah sederetan pulsa yang lebar pulsanya dapat diatur. Pulsa PWM berfungsi mengatur kecepatan motor DC, mengatur gelap terang LED dan lain sebagainya.

Untuk memahami penggunaan PWM, disini digunakan Timer/Counter 1 sebagai PWM. PWM adalah Timer Mode Output Compare yang canggih. Mode PWM timer juga dapat mencacah turun yang berlawanan dengan mode Timer lainnya yang hanya mencacah naik. Pada mode PWM tersebut, Timer mencacah naik hingga mencapai nilai TOP, yaitu 0xFF untuk PWM 8 bit. Timer/Counter 1 memiliki PWM 9 bit dan PWM 10 bit, selain PWM 8 bit. Pemilihan Timer Mode PWM diseting melalui bit WGM01 dan bit WGM00 pada register TCCR0. Tabel Konfigurasi Bit WGM01 dan WGM00 dapat dilihat pada tabel berikut.

Tabel 1.1 Konfigurasi Bit WGM01 dan WGM00

Mode	WGM01 (CTC 0)	WGM00 (PWM0)	Timer Counter Mode of Operation	TOP	Update of OCR0	TOV0 Flag Set-on
0	0	0	Normal	0xFF	Immadiete	MAX
1	0	1	PWM, Phase Correct	0xFF	TOP	BOTTOM
2	1	0	CTC	OCR0	Immadiete	MAX
3	1	1	Fast PWM	0xFF	TOP	MAX

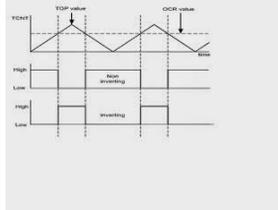
Gambar 2.7 Konfigurasi Bit WGM01 dan WGM00
Sumber: kaka,2013

Sebagai penggunaan mode PWM Timer / Counter 0, keluaran sinyal PWM terletak pada pin OC0 sehingga pada contoh ini LED diletakkan pada pin OC0. Ketika nilai TCNT0 sama dengan nilai pada OCR0, maka output pada OC0 akan berlogika nol atau berlogika satu, tergantung pada pemilihan mode PWM. Anda dapat memilih mode normal atau mode inverted PWM. Pemilihan mode PWM diseting melalui bit COM01 dan bit COM00 pada register TCCR0 yang konfigurasinya seperti tabel berikut.

COM01	COM00	Dscription
0	0	Normal port operation, OC0 disconnected
0	1	Reserved
1	0	Clear OC0 on compare match when up-connection Set OC0 on compare match downcounting
1	1	Set OC0 on compare match when up-counting. Clear OC0 on compare match when downcounting

Gambar 2.8 Konfigurasi Bit COM01 dan COM00 Compare Output Mode Phase Correct PWM.
Sumber:kaka,2013

Dari tabel diatas dapat diketahui saat COM00 clear dan COM01 set, pin OC0 clear saat timer mencacah diatas Compare Match dan pin OC0 set saat timer mencacah dibawah Compare Match atau non-inverting PWM. Kebalikannya, saat COM00 set dan COM01 juga set, maka pin OC0 set saat timer mencacah diatas Compare Match dan pin OC0 clear saat mencacah dibawah Compare Match atau disebut juga inverting PWM. Untuk lebih jelasnya, perhatikan gambar berikut.



Gambar 2.9 Grafik COM00 dan COM01
Sumber:kaka,2013

2.4 Code Vision AVR

2.4. Pengertian Code Vision AVR

CodeVisionAVR pada dasarnya merupakan perangkat lunak pemrograman microcontroller keluarga AVR berbasis bahasa C. Ada tiga komponen penting yang telah diintegrasikan dalam perangkat lunak ini, yaitu *Compiler C*, IDE dan Program generator. Berdasarkan spesifikasi yang dikeluarkan oleh perusahaan pengembangnya, *Compiler C* yang digunakan hampir mengimplementasikan semua komponen standar yang ada pada bahasa C standar ANSI (seperti struktur program, jenis tipe data, jenis operator, dan *library* fungsi standar-berikut penamaannya). Tetapi walaupun demikian, dibandingkan bahasa C untuk aplikasi komputer, *Compiler C* untuk microcontroller ini memiliki sedikit perbedaan yang disesuaikan dengan arsitektur AVR tempat program C tersebut ditanamkan (*embedded*). Khusus untuk *library* fungsi, disamping *library* standar (seperti fungsi-fungsi matematik, manipulasi String, pengaksesan memori dan sebagainya), CodeVisionAVR juga menyediakan fungsi-fungsi tambahan yang sangat bermanfaat dalam pemrograman antarmuka AVR dengan perangkat luar yang umum digunakan dalam aplikasi kontrol. Beberapa fungsi *library* yang penting diantaranya adalah fungsi-fungsi untuk pengaksesan LCD, komunikasi I2C, IC RTC (*Real time Clock*), sensor suhu LM75,

SPI (*Serial Peripheral Interface*) dan lain sebagainya. Untuk memudahkan pengembangan program aplikasi, CodeVisionAVR juga dilengkapi IDE yang sangat bersahabat dengan pengguna. Selain menu-menu pilihan yang umum dijumpai pada setiap perangkat lunak berbasis Windows, CodeVisionAVR ini telah mengintegrasikan perangkat lunak *downloader (in system programmer)* yang dapat digunakan untuk mentransfer kode mesin hasil kompilasi ke dalam sistem memori microcontroller AVR yang sedang di program.

Tabel 2.1 Keterangan Lengkap Ikon-Ikon Dari CodeVision AVR

Sub-menu	Fungsi
New	membuat file baru
Open	membuka file yang sudah ada
Reopen	membuka kembali file yang sudah ditutup
Save	menyimpan yang yang sudah dibuat
Save-as	menyimpan kembali file yang aktif dan sudah disimpan dengan nama yang berbeda
Save All	menyimpan beberapa file yang sudah dibuat
Close	menutup tab file yang sedang dibuka (sama seperti excel, di CodeVisionAVR juga kita dapat membuat beberapa sheet dalam jendela yang sama)
Close	Multiple menutup sebagian tab yang sedang dibuka
Close All	menutup semua tab
Convert to Library	menyimpan file yang sudah dibuat ke dalam library
Page Setup	mengatur tata letak kertas
Print Preview	menampilkan file yang akan dicetak
Print	mencetak file
Exit	keluar dari CodeVisionAVR

Tabel 2.2 Menu Edit

Sub-menu	Fungsi
Undo	mengulang aksi yang terakhir kali dilakukan
Redo	mengembalikan proses undo yang terakhir kali dilakukan

Cut	memindahkan teks yang sudah di-blok ke tempat lain
Copy	menyalin teks yang sudah di-blok
Copy to Code Templates	menyalin teks yang sudah di-blok untuk dijadikan template
Paste	menduplikasi teks yang sudah di-cut atau copy
Delete	menghapus teks
Select All	mem-blok semua teks yang ada dalam satu file
Print Selection	mencetak sebagian teks yang sudah diseleksi
Indent Selection	Memberi spasi pada bagian yang dipilih
Unindent Selection	Menghilangkan spasi pada bagian yang dipilih
Comment Selection	mengubah teks yang dipilih menjadi komentar
Uncomment Selection	mengembalikan teks komentar menjadi teks semula (kebalikan dari comment selection)
Insert Special Character	memasukkan karakter spesial seperti tanda panah, simbol mata uang, huruf latin, dll
Toggle Bookmark	menandai baris tertentu
Jump Bookmark	pergi ke baris yang sudah ditandai
Jump to Next Bookmark	pergi ke baris selanjutnya yang sudah ditandai
Jump to Previous Bookmark	pergi ke baris sebelumnya yang sudah ditandai
Go Back	kembali ke baris yang sudah dijelajahi sebelumnya (setelah melakukan proses go to line)
Go Forward	kembali ke baris yang sudah dijelajahi setelahnya
Go to Definition/Declaration	pergi ke bagian definisi atau deklarasi
Go to Line	pergi ke baris yang diinginkan
Go to Matching Brace	pergi ke baris yang memiliki pasangan tanda kurung ()

Tabel 2.3 Menu Search

Sub-menu	Fungsi
Find	mencari teks tertentu
Find Next	mencari teks selanjutnya
Find Previous	mencari teks sebelumnya
Find in Files	mencari teks string yang ada pada file
Replace	mengganti suatu teks dengan teks lainnya
Replace in Files	mengganti teks string dengan teks string lainnya

Tabel 2.4 Menu View

Sub-menu	Fungsi
Toggle Fold	menyusutkan atau memperluas satu blok kode tempat kursor berada yang dihimpit oleh { }
Expand All Folds	memperluas satu blok kode tempat kursor berada yang dihimpit oleh { }
Collapse All Folds	menyusutkan satu blok kode tempat kursor berada yang dihimpit oleh { }
Visible Non-Printable Characters	menampilkan karakter yang tidak akan ikut dicetak, karena karakter ini hanya digunakan untuk formatting, misalnya karakter ¶
Code Navigator	menampilkan atau membuka file yang sudah dibuat dalam bentuk diagram pohon (seperti di windows explorer) beserta keterangan apakah terdapat error atau warning pada file tersebut
Code Information	menampilkan deklarasi atau definisi yang terdapat pada file yang sedang di-edit atau dibuka
Function Call Tree	menampilkan fungsi panggil yang menggunakan data stack terbanyak selama eksekusi program
Code Templates	menampilkan kode yang sering digunakan, untuk menggunakannya cukup klik kode yang akan digunakan dan drag ke window editing
Clipboard History	menampilkan sebagian teks yang baru saja digandakan (copy) ke clipboard

Messages	menampilkan pesan error atau warning
Find in Files	menampilkan jendela yang berisi tentang pencarian teks string
Information Window after Compile/Build	menampilkan jendela informasi ketika program selesai di-compile atau di-build
Toolbars	menampilkan toolbar
File Panes	menampilkan window editing dalam bentuk single pane, dual pane vertical, dan dual pane horizontal.

Tabel 2.5 Menu *Project*

Sub-menu	Fungsi
Check Syntax	mengecek sintaks yang digunakan
Compile	meng-compile program yang sudah dibuat
Build	mem-build program yang sudah dibuat
Build All	mem-build program yang sudah dibuat
Stop compilation	menghentikan kompilasi
Clean Up	menghapus file yang dihasilkan dari proses build
Information	menampilkan jendela informasi yang berisi compiler dan assembler
Go to Next Error	pergi menuju error selanjutnya
Go to Previous Error	pergi menuju error sebelumnya
Go to Next Warning	pergi menuju warning selanjutnya
Go to Previous Warning	pergi menuju warning sebelumnya
Notes	menampilkan jendela catatan
Configure	melakukan konfigurasi program, biasanya dilakukan sebelum program diinjeksikan ke mikrokontroler

Tabel 2.6 Menu *Tools*

Sub-menu	Fungsi
CodeWizardAVR	membuat program secara otomatis sehingga mempermudah penulisan kode inisialisasi
Debugger	memilih file debugger
Chip Programmer	pengaturan untuk memasukkan program yang sudah di-compile ke mikrokontroler
Terminal	pengaturan terminal komunikasi serial

Configure	menambah atau menghilangkan program ke dalam tools menu
-----------	---

Tabel 2.7 Menu *Settings*

Sub-menu	Fungsi
IDE	pengaturan tampilan IDE, meliputi load last used project at startup, show hint for the code navigator window, show hint for the code information window, dan show hint for the function call tree window.
Editor	pengaturan ukuran huruf, warna huruf, menampilkan jumlah baris, dan lain-lain
Debugger	mengatur debugger sebelum dihubungkan ke AVR Studio
Programmer	pengaturan jenis mikrokontroler yang digunakan dan terminal komunikasi
Terminal	pengaturan terminal yang digunakan

Tabel 2.8 Menu *Help*

Sub-menu	Fungsi
Help Topics	berisi panduan mengenai penggunaan CodeVisionAVR
Getting Started with CodeVisionAVR	berisi panduan memulai CodeVisionAVR (butuh koneksi internet)
AVR Data Sheets	berisi data sheet mikrokontroler (butuh koneksi internet)
HP Info Tech on the Web	link ke HP Info Tech untuk meminta bantuan dan lainnya
About CodeVisionAVR	berisi penjelasan software CodeVisionAVR, seperti produsen dan versinya

(Sumber : Wida Lidiawati, 2015)

2.4 USB TTL (Serial)

2.5.1 Pengertian USB TTL (Serial)

Serial Port atau biasa disebut dalam bahasa Indonesia adalah port seri merupakan sebuah port pada personal computer yang berfungsi untuk mentransmisikan satu bit informasi pada satu satuan waktu. Dalam serial port, pengiriman informasi tidak memungkinkan untuk melakukan secara banyak sekaligus.



Gambar 2.10

Sumber: (<http://elearningelektronika.blogspot.co.id/2008/05/interfecing-menggunakan-usb.html>)

Hal ini disebabkan karena dalam melakukan pemindahan data, biasanya serial port bekerja seri, misalnya COM 1 dan COM 2. Untuk penggunaan port serial sekarang ini sudah berkurang. Penggunaan port serial telah tergantikan dengan port USB dan Firewire. Sedangkan untuk jaringan (networking) fungsinya sudah tergantikan dengan port Ethernet.

2.5 Virtual C#

2.5.1 Pengertian Virtual C#

Virtual C# adalah bahasa pemrograman computer, sama seperti C,C++, Java , maupun yang lainnya.perbedaannya, C# menggunakan library kelas yang terdapat pada NET Framework . hal ini tentu berbeda dengan C, C++, maupun java, yang masing-masing memiliki library kelas sendiri – sendiri. Kelebihan library kelas yang

terdapat di dalam .NET Framework adalah dapat digunakan oleh bahasa-bahasa lain yang mendukung .NET, seperti visual basic dan visual C++.(sumber : Budi Raharjo)

Perbedaan antara C, C++ , dan C# , sebenarnya tidak ada hubungan khusus diantara ketiganya . masing – masing nya ini merupakan bahasa pemrograman computer yang berjalan secara terpisah .

Program di dalam C# dapat berupa kelas maupun struktur . dalam kelas tersebut harus terdapat metode *Main* () , yaitu metode utama yang akan dipanggil pertama kali pada saat program dijalankan. Metode itu sendiri adalah fungsi yang didefinisikan didalam suatu kelas maupun struktur secara umum. Sebernarnya tidak ada hubungan khusus diantara ketiganya . masing – masing (C,C++, DAN C#)

Pada aplikasi C#, anda dapat menggunakan alias namespace untuk memudallkan pengaruh class-class .NET Framework. Alias namespace digunakan untuk merekomendasi atau memberi keterangan suatu class di dalam .NET Framework. Berikut merupakan suatu contoh source code sederhana untuk Pemrograman C# :

```
using system;
class hello
{
static void main() (
console. writclinc('Hello Word");
}
}
```

Ekstensi default dari file C# adalah .cs, kemudian akan dikompilasi menjadi file eksekusi menjadi file berekstensi .exe.

1"1

Ilustrasi dari progam ini dapat dilihat sebagai berikut

a. *Using system* : merupakan referensi langsung dari name space yang disebut system yang telah di\$edikan oleh .NET. name space ini mengandung console class yang terdapat pOOametode main. Name space menyediakan arti

secara hirarkial dari pengorganisasian elemen kelas library. "Hello, World" program menggunakan `console.WriteLine` sebagai sebuah kependekan dari `System.Console.WriteLine`. `System` adalah sebuah name space, `console` sebagai kelas yang didefinisikan pada name space dan `WriteLine` adalah metode statis yang didefinisikan pada kelas ini.

b. Mainfunction merupakan member statis dari kelas `Hello`.

Functions dan Variabel tidak terdapat pada level global~ seperti elemen yang harus didefinisikan pada tipe.

c. Output Hello World dihasilkan melalui penggunaan kelas library. C# sendiri tidak mempunyai kelas library sendiri. Sehingga penggunaan kelas library pada C# menggunakan kelas library yang dipakai pada Visual Basic dan Visual C++.

Untuk C dan C++ developer, adalah menarik untuk dilihat bahwa ada perbedaan mendasar pada bahasa C & C++ terhadap C#. Program ini tidak menggunakan baik operator ":" ataupun "->". Operator "<code>"</code>" bukanlah operator C# dan operator "->" hanya digunakan pada sebagian

kecil C# program. C# menggunakan "." sebagai pemisah penggunaan fungsi seperti `Console.WriteLine`. Program ini tidak mengandung deklarasi awal. Deklarasi awal tidak pernah digunakan pada program C#, seperti juga perincian deklarasi tidaklah sama.

Program ini tidak menggunakan `#include` untuk mengimpor file program. Ketergantungan antar program telah dilayani secara simbolik dibandingkan secara text program. Sistem ini menghilangkan batasan penulisan antar program yang ditulis menggunakan bahasa berbeda. Sebagai contoh, kelas `Console` dapat digunakan pada C# atau pada bahasa pemrograman lain.