

BAB II

TEORI DASAR DAN PENDUKUNG

2.1 Metode Pendaratan Helikopter Tanpa Awak

Kebanyakan penelitian sistem pendaratan darurat pada helikopter dilakukan oleh perusahaan pembuatnya dan terdapat beberapa metode yang digunakan perusahaan-perusahaan tersebut, diantaranya menggunakan Gelombang Radio, Pengolahan Citra (Image Processing) dan Estimasi Posisi (GPS).

2.1.1 Radar (*Radio Detection and Ranging*)

Radar merupakan sistem gelombang elektromagnetik yang digunakan untuk mendeteksi, mengukur jarak dan membuat peta benda-benda seperti posisi pesawat terbang, kendaraan bermotor dan informasi cuaca/hujan. Walaupun sedang cuaca buruk seperti hujan lebat dan berkabut, namun dengan menggunakan radar, informasi berupa jarak dan kecepatan suatu obyek dari posisi radar masih bisa didapatkan. Selain itu, radar dapat mendeteksi obyek pada jarak yang sangat jauh (ratusan kilometer). Radar pada metode pendaratan ini digunakan untuk mengontrol pesawat dari pusat pemancar (transmitter) untuk menentukan arah, kecepatan dan ketinggian helikopter. Sierra Nevada Corporation mengembangkan teknologi penggunaan radar untuk lepas landas dan pendaratan pesawat metode ini membutuhkan operator untuk mengedalikan pesawat pada pusat pemancar, walaupun demikian metode ini telah mencapai kesuksesan lebih dari 400 kali sejak tahun 2001⁽¹⁾.

2.1.2 Pengolahan Citra (*Visual Landmarks*)

Gambar yang didapatkan kamera berupa 2 dimensi diproses menjadi nilai atau informasi dengan mengubah gambar tersebut dengan mengubah nilai-nilai pada gambar tersebut baik dalam cahaya, warna dan resolusi gambar tersebut, lalu dilakukan pendekatan kemiripan, pada Helipad (Landasan Helikopter) diberi tanda huruf “H” sehingga driver helikopter tahu dimana posisi landasan, sehingga gambar yang dihasilkan menjadi titik acuan helikopter untuk mendarat⁽²⁾.

Berdasarkan tujuan transformasi operasi pengolahan citra dikategorikan sebagai berikut⁽³⁾ :

- **Peningkatan Kualitas Citra (*Image Enhancement*)**

Operasi peningkatan kualitas citra bertujuan untuk meningkatkan fitur tertentu pada citra.

- **Pemulihan Citra (*Image Restoration*)**

Operasi pemulihan citra bertujuan untuk mengembalikan kondisi citra pada kondisi yang diketahui sebelumnya akibat adanya pengganggu yang menyebabkan penurunan kualitas citra.

Berdasarkan cakupan operasi yang dilakukan terhadap citra, Operasi pengolahan citra dikategorikan sebagai berikut :

- **Operasi titik**, yaitu operasi yang dilakukan terhadap setiap piksel pada citra yang keluarannya hanya ditentukan oleh nilai piksel itu sendiri.
- **Operasi area**, yaitu operasi yang dilakukan terhadap setiap piksel pada citra yang keluarannya dipengaruhi oleh piksel tersebut dan piksel lainnya dalam suatu daerah tertentu. Salah satu contoh dari operasi berbasis area adalah operasi ketetanggaan yang nilai keluaran dari operasi tersebut ditentukan oleh nilai piksel-piksel yang memiliki hubungan ketetanggaan dengan piksel yang sedang diolah.
- **Operasi global**, yaitu operasi yang dilakukan terhadap setiap piksel pada citra yang keluarannya ditentukan oleh keseluruhan piksel yang membentuk citra.

2.1.3 Estimasi Posisi (*Position Estimation*)

Estimasi Posisi atau perkiraan posisi landasan ditentukan oleh pemetaan lokasi landasan yang ditangkap oleh penerima sinyal GPS (*Global Positioning System*) pada helikopter, sinyal yang dipancarkan oleh satelit diterima oleh penerima dan penerima mengirimkan posisi koordinat penerima. Tidak seperti radar, metode ini dapat mengarahkan helikopter kelokasi yang diinginkan tanpa bantuan operator.

2.1.3.1 Cara Kerja GPS

Sistem ini menggunakan sejumlah satelit yang berada di orbit bumi, yang memancarkan sinyalnya ke bumi dan ditangkap oleh sebuah alat penerima. Ada tiga bagian penting dari sistim ini, yaitu bagian kontrol, bagian angkasa, dan bagian pengguna⁽⁴⁾.

- Bagian Kontrol

Seperti namanya, bagian ini untuk mengontrol. Setiap satelit dapat berada sedikit diluar orbit, sehingga bagian ini melacak orbit satelit, lokasi, ketinggian, dan kecepatan. Sinyal-sinyal dari satelit diterima oleh bagian kontrol, dikoreksi, dan dikirimkan kembali ke satelit. Koreksi data lokasi yang tepat dari satelit ini disebut dengan data ephemeris, yang nantinya akan di kirimkan kepada alat navigasi kita.

- Bagian Angkasa

Bagian ini terdiri dari kumpulan satelit-satelit yang berada di orbit bumi, sekitar 12.000 mil diatas permukaan bumi. Kumpulan satelit-satelit ini diatur sedemikian rupa sehingga alat navigasi setiap saat dapat menerima paling sedikit sinyal dari empat buah satelit. Sinyal satelit ini dapat melewati awan, kaca, atau plastik, tetapi tidak dapat melewati gedung atau gunung. Satelit mempunyai jam atom, dan juga akan memancarkan informasi 'waktu/jam' ini. Data ini dipancarkan dengan kode 'pseudo-random'. Masing-masing satelit memiliki kodenya sendiri-sendiri. Nomor kode ini biasanya akan ditampilkan di alat navigasi, maka kita bisa melakukan identifikasi sinyal satelit yang sedang diterima alat tersebut. Data ini berguna bagi alat navigasi untuk mengukur jarak antara alat navigasi dengan satelit, yang akan digunakan untuk mengukur koordinat lokasi. Kekuatan sinyal satelit juga akan membantu alat

dalam penghitungan. Kekuatan sinyal ini lebih dipengaruhi oleh lokasi satelit, sebuah alat akan menerima sinyal lebih kuat dari satelit yang berada tepat di atasnya (bayangkan lokasi satelit seperti posisi matahari ketika jam 12 siang) dibandingkan dengan satelit yang berada di garis cakrawala (bayangkan lokasi satelit seperti posisi matahari terbenam/terbit).

Ada dua jenis gelombang yang saat ini dipakai untuk alat navigasi berbasis satelit pada umumnya, yang pertama lebih dikenal dengan sebutan L1 pada 1575.42 MHz. Sinyal L1 ini yang akan diterima oleh alat navigasi. Satelit juga mengeluarkan gelombang L2 pada frekuensi 1227.6 Mhz. Gelombang L2 ini digunakan untuk tujuan militer dan bukan untuk umum.

- **Bagian Pengguna**

Bagian ini terdiri dari alat navigasi yang digunakan. Satelit akan memancarkan data almanak dan ephemeris yang akan diterima oleh alat navigasi secara teratur. Data almanak berisikan perkiraan lokasi (*approximate location*) satelit yang dipancarkan terus menerus oleh satelit. Data ephemeris dipancarkan oleh satelit, dan valid untuk sekitar 4-6 jam. Untuk menunjukkan koordinat sebuah titik (dua dimensi), alat navigasi memerlukan paling sedikit sinyal dari 3 buah satelit. Untuk menunjukkan data ketinggian sebuah titik (tiga dimensi), diperlukan tambahan sinyal dari 1 buah satelit lagi.

Dari sinyal-sinyal yang dipancarkan oleh kumpulan satelit tersebut, alat navigasi akan melakukan perhitungan-perhitungan, dan hasil akhirnya adalah koordinat posisi alat tersebut. Makin banyak jumlah sinyal satelit yang diterima oleh sebuah alat, akan membuat alat tersebut menghitung koordinat posisinya dengan lebih tepat.

2.2 Mikrokontroler ATmega8

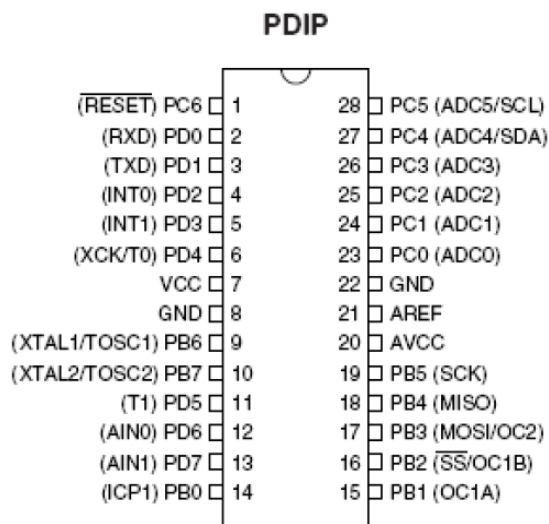
AVR merupakan salah satu jenis *microcontroller* yang di dalamnya terdapat berbagai macam fungsi. Perbedaannya dengan mikro yang pada umumnya digunakan seperti MCS 51 adalah pada AVR tidak perlu menggunakan *oscillator* eksternal karena di dalamnya sudah terdapat internal *oscillator*. Selain itu kelebihan dari AVR adalah memiliki *Power-On Reset*, yaitu tidak perlu adanya tombol *reset* dari luar karena cukup hanya dengan mematikan *supply*, maka secara otomatis AVR akan melakukan *reset*. Untuk beberapa jenis AVR terdapat beberapa fungsi khusus seperti ADC, EEPROM sekitar 128 *bytes* sampai dengan 512 *bytes*. Dalam hal ini yang

digunakan adalah AVR ATmega8, perbedaannya dengan AVR ATmega8L hanyalah terletak pada besarnya tegangan yang diperlukan untuk bekerja. Untuk ATmega8 tipe L dapat bekerja pada tegangan antara 2,7V – 5,5V sedangkan untuk ATmega8 hanya dapat bekerja pada tegangan 4,5V – 5,5V.

Kelebihan dari ATmega8 sehingga digunakan sebagai kontrol utama adalah sebagai berikut :

- Mempunyai performa yang tinggi (berkecepatan akses maksimum 16MHz) dan hemat daya
- Memori untuk program flash cukup besar yaitu 8K Byte
- Memori internal SRAM sebesar 1K Byte
- EEPROM sebesar 512 byte yang dapat diprogram saat operasi
- Port komunikasi SPI
- Komunikasi serial standar USART
- Tersedia 3 chanel PWM
- Tersedia 3 chanel timer/counter (2 untuk 8 bits dan 1 untuk 16 bits)

Berikut adalah gambar dari blok diagram untuk ATmega8.



Gambar 2.1. *Pin Configuration*

ATMega8 memiliki 28 *pin* yang masing-masing *pin*-nya memiliki fungsi yang berbeda-beda baik sebagai *port* ataupun sebagai fungsi yang lain. Berikut akan dijelaskan tentang kegunaan dari masing-masing kaki pada ATMega8.

□ VCC

Merupakan *supply* tegangan untuk digital.

□ GND

Merupakan *ground* untuk semua komponen yang membutuhkan *grounding*.

□ Port B

Di dalam Port B terdapat XTAL1, XTAL2, TOSC1, TOSC2. Jumlah Port B adalah 8 buah *pin* mulai dari *pin* B.0 sampai dengan *pin* B.7. Tiap *pin* dapat digunakan sebagai *input* dan juga *output*. Port B merupakan sebuah 8-bit *bi-directional* I/O port dengan internal *pull-up* resistor. Sebagai *input*, *pin-pin* yang terdapat pada port B yang secara eksternal diturunkan, maka akan mengeluarkan arus jika *pull-up* resistor diaktifkan. Jika ingin menggunakan tambahan kristal, maka cukup menghubungkan kaki dari kristal ke kaki pada *pin port B*. Namun jika tidak digunakan, maka cukup dibiarkan saja. Penggunaan kegunaan dari masing-masing kaki ditentukan dari *clock fuse setting*-nya.

□ Port C

Port C merupakan sebuah 7-bit *bi-directional* I/O port yang di dalam masing-masing *pin* terdapat *pull-up* resistor. Jumlah *pin*-nya hanya 7 buah mulai dari *pin* C.0 sampai dengan *pin* C.6. Sebagai keluaran / *output*, port C memiliki karakteristik yang sama dalam hal kemampuan menyerap arus (*sink*) ataupun mengeluarkan arus (*source*) dan juga berfungsi sebagai input ADC (Analog to Digital Converter).

□ Reset / PC6

Jika RSTDISBL Fuse diprogram, maka PC6 akan berfungsi sebagai *pin* I/O. Untuk diperhatikan juga bahwa *pin* ini memiliki karakteristik yang berbeda dengan *pin-pin* yang terdapat pada port C. Namun jika RSTDISBL Fuse tidak diprogram, maka *pin* ini akan berfungsi sebagai *input reset*. Dan jika level tegangan yang masuk ke *pin* ini

rendah dan pulsa yang ada lebih pendek dari pulsa minimum, maka akan menghasilkan suatu kondisi *reset* meskipun *clock*-nya tidak bekerja.

□ *Port D*

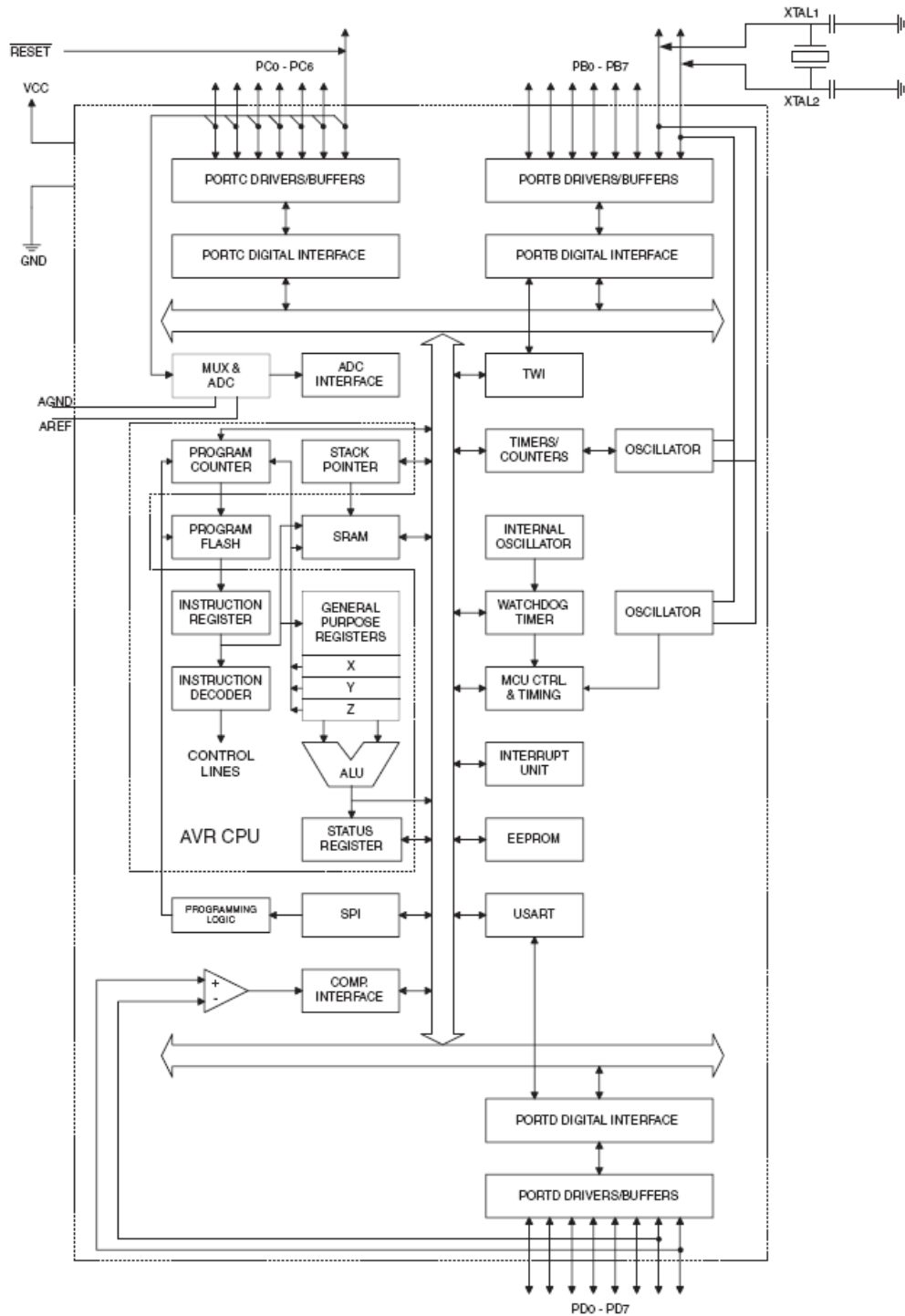
Port D merupakan 8-bit *bi-directional* I/O dengan internal *pull-up* resistor. Fungsi dari *port* ini sama dengan *port-port* yang lain. Hanya saja pada *port* ini tidak terdapat kegunaan-kegunaan yang lain. Pada *port* ini hanya berfungsi sebagai masukan dan keluaran saja atau biasa disebut dengan I/O.

□ AVCC

Pada *pin* ini memiliki fungsi sebagai *supply* tegangan untuk ADC. Untuk *pin* ini harus dihubungkan secara terpisah dengan VCC karena *pin* ini digunakan untuk *analog* saja. Bahkan jika ADC pada AVR tidak digunakan, tetap saja disarankan untuk menghubungkan secara terpisah dengan VCC. Cara menghubungkan AVCC adalah melewati *low-pass filter* setelah itu dihubungkan dengan VCC.

□ AREF

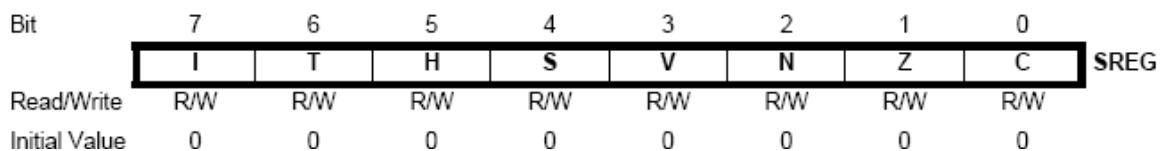
Merupakan *pin* referensi *analog* jika menggunakan ADC.



Gambar 2.2. Blok Diagram ATmega8

Pada AVR status *Register* mengandung beberapa informasi mengenai hasil dari kebanyakan hasil eksekusi instruksi aritmatik. Informasi ini dapat digunakan untuk *altering* arus program sebagai kegunaan untuk meningkatkan performa pengoperasian. Perlu diketahui bahwa *register* ini di-*update* setelah semua operasi

ALU (*Arithmetic Logic Unit*). Hal tersebut seperti yang telah tertulis dalam *datasheet* khususnya pada bagian *Instruction Set Reference*. Dalam hal ini untuk beberapa kasus dapat membuang kebutuhan penggunaan instruksi perbandingan yang telah didedikasikan serta dapat menghasilkan peningkatan dalam hal kecepatan dan kode yang lebih sederhana dan singkat. *Register* ini tidak secara otomatis tersimpan ketika memasuki sebuah rutin interupsi dan juga ketika menjalankan sebuah perintah setelah kembali dari interupsi. Namun hal tersebut harus dilakukan melalui *software*. Berikut adalah gambar *Status Register*.



Gambar 2.3. *Status Register* ATmega8

Akan dijelaskan kegunaan dari masing-masing *bit* yang terlihat di atas :

Bit 7 (I)

Merupakan *bit Global Interrupt Enable*. *Bit* ini harus di-*set* supaya semua perintah interupsi dapat dijalankan. Untuk fungsi interupsi individual akan dijelaskan pada bagian yang lain. Jika *bit* ini di-*reset*, maka semua perintah interupsi baik yang individual maupun yang secara umum akan diabaikan. *Bit* ini akan dibersihkan atau *cleared* oleh *hardware* setelah sebuah interupsi dijalankan dan akan di-*set* kembali oleh perintah RETI. *Bit* ini juga dapat di-*set* dan di-*reset* melalui aplikasi dengan instruksi SEI dan CLI.

Bit 6 (T)

Merupakan *bit Copy Storage*. Instruksi *bit Copy Instructions BLD (Bit Load) and BST (Bit Store)* menggunakan *bit* ini sebagai asal atau tujuan untuk *bit* yang telah dioperasikan. Sebuah *bit* dari sebuah *register* dalam *Register File* dapat disalin ke dalam *bit* ini dengan menggunakan instruksi BST, dan sebuah *bit* di dalam *bit* ini dapat disalin ke dalam sebuah *bit* di dalam *register* pada *Register File* dengan menggunakan perintah BLD.

□ *Bit 5 (H)*

Merupakan *bit Half Carry Flag*. *Bit* ini menandakan sebuah *Half Carry* dalam beberapa operasi aritmatika. *Bit* ini berfungsi dalam aritmatik BCD.

□ *Bit 4 (S)*

Merupakan *Sign bit*. *Bit* ini selalu merupakan sebuah eksklusif di antara *Negative Flag (N)* dan *Two's Complement Overflow Flag (V)*. $S = N \oplus V$.

□ *Bit 3 (V)*

Merupakan *bit Two's Complement Overflow Flag*. *Bit* ini menyediakan fungsi aritmatika dua komplemen.

□ *Bit 2 (N)*

Merupakan *bit Negative Flag*. *Bit* ini mengindikasikan sebuah hasil negative di dalam sebuah fungsi logika atau aritmatika.

□ *Bit 1 (Z)*

Merupakan *bit Zero Flag*. *Bit* ini mengindikasikan sebuah hasil nol "0" dalam sebuah fungsi aritmatika atau logika.

□ *Bit 0 (C)*

Merupakan *bit Carry Flag*. *Bit* ini mengindikasikan sebuah *Carry* atau sisa dalam sebuah fungsi aritmatika atau logika.

2.3 SENSOR ULTRASONIK

Ultrasonik, sebutan untuk jenis suara diatas batas suara yang bisa didengar manusia. Seperti diketahui, telinga manusia hanya bisa mendengar suara dengan frekuensi 20 Hz sampai 20KHz. Lebih dari itu hanya beberapa jenis binatang yang mampu mendengarnya, seperti kelelawar dan lumba-lumba. Lumba-lumba bahkan memanfaatkan ultrasonik untuk mengindra benda-benda di laut. Prinsip ini kemudian ditiru oleh sistem pengindra kapal selam. Dengan cara mengirimkan sebuah suara dan mengitung lamanya pantulan suara tersebut maka dapat diketahui jarak kapal selam dengan benda tersebut. Mula-mula suara dibunyikan, kemudian dihitung lama waktu sampai terdengar suara pantulan. Jarak dapat dihitung dengan mengalikan kecepatan suara dengan waktu pantulan. Kemudian hasilnya dibagi 2. Misalnya lama waktu pantulan adalah 1 detik, maka jaraknya adalah $(344,424\text{m/detik} \times 1 \text{ detik})/2 = 172\text{m}$.

Ping Ultrasonic Range Finder, adalah modul pengukur jarak dengan ultrasonic buatan Parallax Inc. yang didesain khusus untuk teknologi robotika. Dengan ukurannya yang cukup kecil (2,1cm x 4,5cm), sensor ini dapat mengukur jarak antara 3 cm sampai 300 cm. Keluaran dari Ping berupa pulsa yang lebarnya merepresentasikan jarak. Lebar pulsanya bervariasi dari 115 uS sampai 18,5 mS.



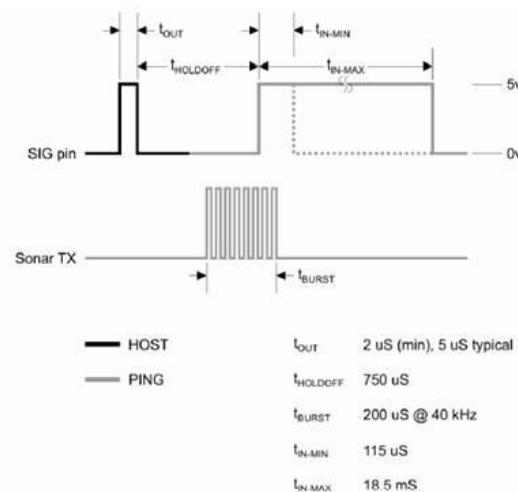
Gambar2.6. Ping Parallax Ultrasonic Range Finder

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. Pada modul Ping terdapat 3 pin yang digunakan untuk jalur power

supply (+5V), ground dan signal. Pin signal dapat langsung dihubungkan dengan mikrokontroler tanpa tambahan komponen apapun. Ping mendeteksi objek dengan cara mengirimkan suara ultrasonik dan kemudian “mendengarkan” pantulan suara

tersebut. 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. Selanjutnya mikrokontroler cukup mengukur lebar pulsa tersebut dan mengkonversinya dalam bentuk jarak dengan perhitungan sebagai berikut

$$\text{Jarak} = (\text{Lebar Pulsa} / 29.034\mu\text{S}) / 2 \text{ (dalam cm)} \text{ atau } \text{Jarak} = (\text{Lebar Pulsa} \times 0.034442) / 2 \text{ (dalam cm)}$$



Karena $1/29.034 = 0.34442$

Gambar 2.7 Pulsa Ping Parallax Ultrasonic Range Finder

Satu hal yang perlu diperhatikan adalah bahwa ping tidak dapat mengukur objek yang permukaannya dapat menyerap suara, seperti busa atau sound damper lainnya. Pengukuran jarak juga akan kacau jika permukaan objek bergerigi dengan sudut tajam (meruncing).

2.4 Pemancar dan Penerima (*Transmitter and Receiver*)

Pemancar dan penerima merupakan perangkat yang digunakan dalam sistem komunikasi dengan media udara untuk mengirimkan gambar, suara ataupun data. Selain penggunaannya dalam penyiaran, pemancar juga merupakan bagian komponen penting dari perangkat elektronik lainnya yang berkomunikasi dengan menggunakan gelombang radio, seperti ponsel, jaringan *wifi*, *walkie-talkie*, *radar*, dan komunikasi

via satelit luar angkasa. Pemancar merupakan perangkat yang mengubah pengukuran dari sensor menjadi sinyal elektromagnetik, dan mengirimkannya melalui antena sehingga dapat diterima oleh perangkat penerima. Sebuah penerima mengkonversi sinyal dari antena radio ke bentuk yang dapat digunakan. Menggunakan penyaring data (*data filtering*) dengan menggunakan IC untuk memisahkan data yang dikirim dari pembawa sinyal (*carrier*) dan derau (*noise*), dan dengan beberapa penguat (*amplifier*), kualitas data tersebut dapat dikembalikan seperti data yang asli, beberapa proses tersebut lebih dikenal dengan istilah demodulasi dan decoding.

2.5 Pengontrol Kecepatan (*Speed Controller*)

Speed Controller atau pengontrol kecepatan merupakan sinyal atau tegangan yang dapat mengatur kecepatan putaran motor listrik, arah dan bisa juga sebagai rem dinamis, *Speed Controller* biasanya diaplikasikan pada driver motor yang tersusun dari beberapa transistor atau lebih dikenal dengan istilah *H-Bridge*. Salah satu jenis *Speed Controller* yang biasa digunakan dalam pengendalian motor dc pada robot adalah Pwm (Pulse width modulation) yang merupakan tegangan keluaran dari mikrokontroler berupa gelombang kotak (*pulse*), bila nilai frekuensi tegangan pwm pada mikrokontroler dinaikkan maka kecepatan putaran pada motor akan bertambah.

2.6 Giroskop (*Gyroscope*)

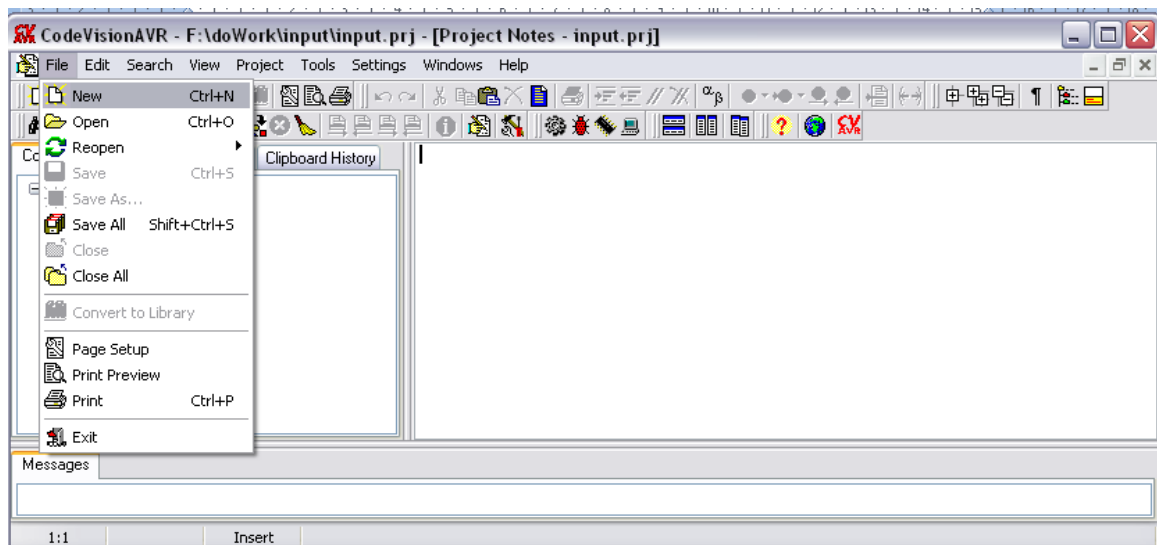
Giroskop merupakan alat untuk mengukur atau mempertahankan orientasi/penyesuaian arah yang terdiri atas piringan dan beberapa pegangan piringan yang dapat berputar dan bekerja berdasarkan prinsip-prinsip kekekalan momentum sudut. Pada dasarnya, sebuah giroskop adalah mekanik berupa roda yang berputar atau piringan yang porosnya bebas untuk mengambil setiap orientasi. Meskipun orientasi ini tidak tetap, namun perubahan torsi eksternal jauh lebih sedikit dalam arah yang berbeda (pegangan piringan) dan dari putaran itu akan ada momentum sudut besar yang terkait dengan putaran pegangan piringan lainnya dengan kecepatan yang sesuai momen inersia.

2.7 CodeVisionAVR COMPILER

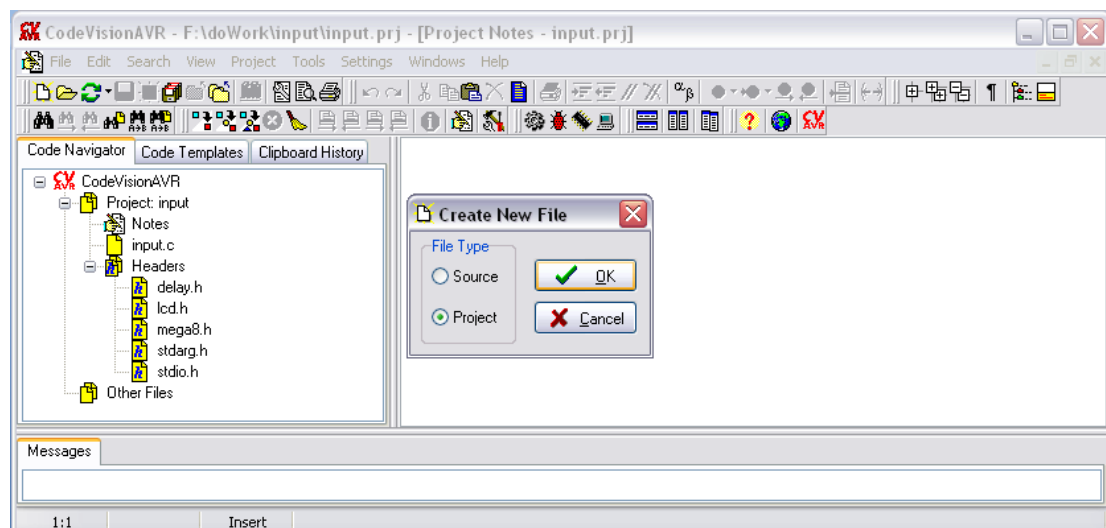
Pemrograman Bahasa C untuk AVR digunakan untuk pemrograman berbagai jenis perangkat, termasuk mikrokontroler. Bahasa ini sudah merupakan high level language, dimana memudahkan programmer menuangkan algoritmanya.

Aplikasi CodeVision AVR C Compiler :

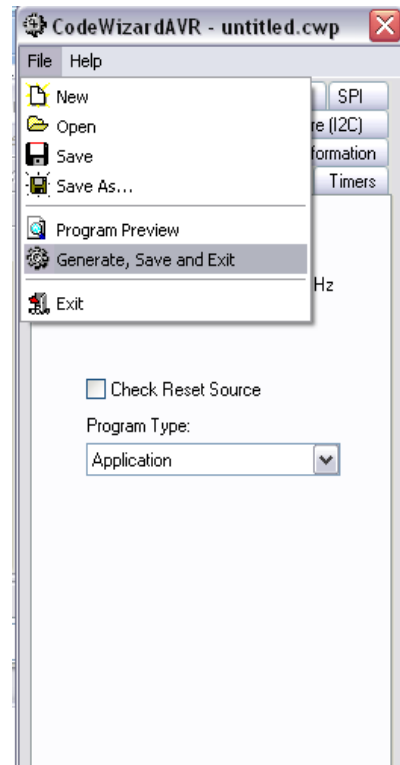
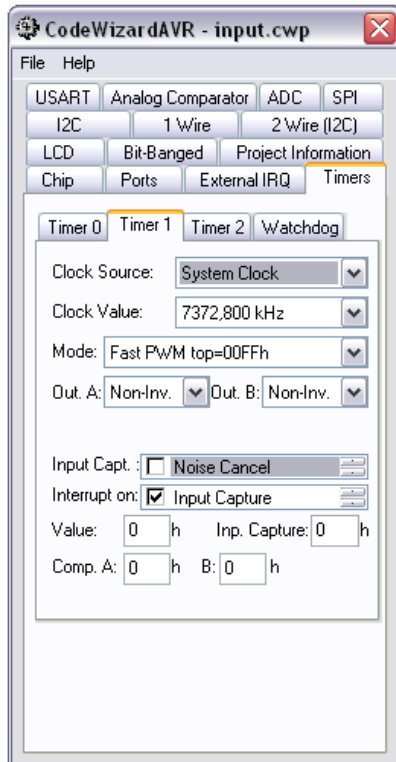
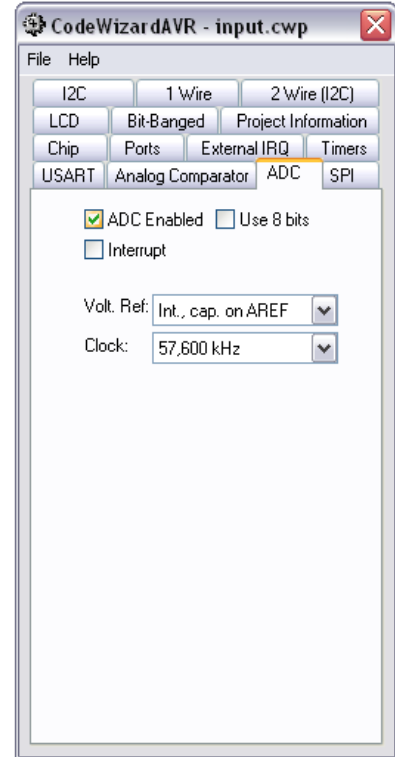
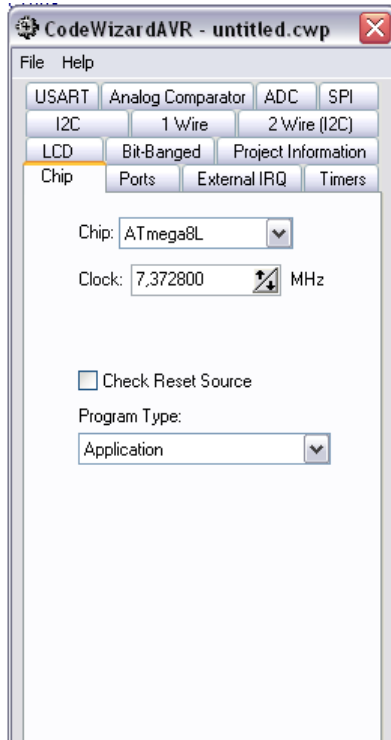
- Klik **File** -> **New** muncul seperti gambar di bawah ini



- Pilih project OK



- Lakukan Settingan Sesuai karakteristik Mikrokontroler dan Kinerja yang diinginkan, dan **“Generate, Save and Exit”** pada menu **File** untuk menyimpan program.



NB: Untuk **Chip** dan **Clock** sesuaikan dengan yang anda gunakan
Simpan sesuai nama yang anda inginkan

Example : ach.c (untuk file source C)
 ach.prj (untuk file project)
 ach.cwp (untuk file Code Wizard Project)