

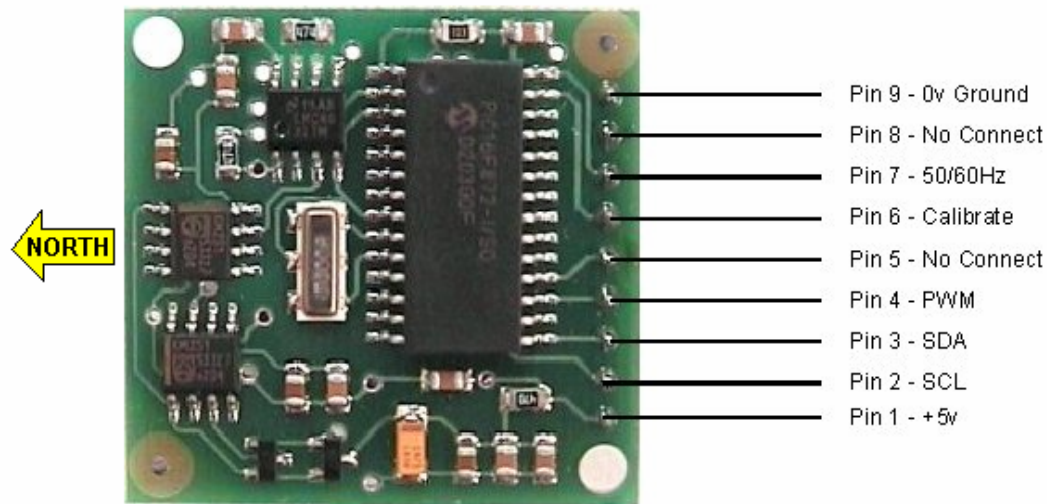
Digital Compass CMPS03

By : Hendawan Soebhakti

1. Karakteristik Digital Compass

Mobile robot, adalah istilah yang sering digunakan untuk menyebut sebuah robot yang memiliki kemampuan menjelajah. Tidak peduli apakah robot tersebut bergerak menggunakan roda, kaki, maupun kipas untuk berenang atau bahkan terbang sekalipun, maka robot semacam ini masuk ke dalam kategori mobile robot. Agar tidak nyasar, robot harus dilengkapi dengan sistem navigasi yang dapat memberikan informasi arah dengan baik. Sehingga robot dapat memutuskan dengan benar ke arah mana seharusnya bergerak untuk mencapai lokasi yang diinginkan.

Sistem navigasi yang cukup baik, efektif, mudah digunakan dan murah meriah adalah dengan kompas digital. Banyak jenis kompas digital yang diproduksi khusus untuk keperluan robotika, salah satu yang sangat populer adalah CMPS03 Magnetic Compass buatan Devantech Ltd. CMPS03 yang berukuran 4 x 4 cm ini menggunakan sensor medan magnet Philips KMZ51 yang cukup sensitif untuk mendeteksi medan magnet bumi.



Gambar 1. CMPS03 Digital Compass

Kompas digital ini hanya memerlukan suplai tegangan sebesar 5 Vdc dengan konsumsi arus 15mA. Pada CMPS03, arah mata angin dibagi dalam bentuk derajat yaitu : Utara (0°), Timur (90°), Selatan (180°) dan Barat (270°). Ada dua cara untuk mendapatkan informasi arah dari modul kompas digital ini yaitu dengan membaca sinyal PWM (*Pulse Width Modulation*) pada pin 4 atau dengan membaca data interface I2C pada pin 2 dan 3.

PWM Interface

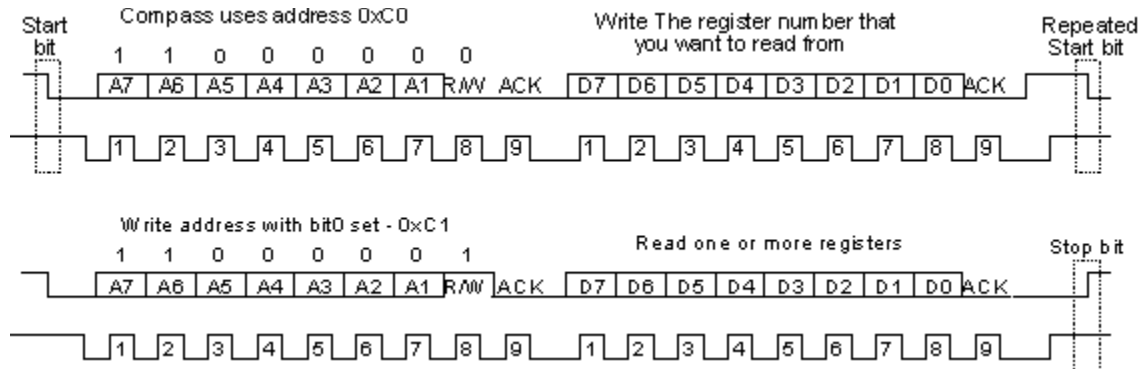
Sinyal PWM adalah sebuah sinyal yang telah dimodulasi lebar pulsanya. Pada CMPS03, lebar pulsa positif merepresentasikan sudut arah. Lebar pulsa bervariasi antara 1mS (0°) sampai 36.99mS (359.9°). Dengan kata lain lebar pulsa berubah sebesar 100uS setiap derajatnya. Sinyal akan low selama 65mS diantara pulsa, sehingga total periodanya adalah 65mS + lebar pulsa positif (antara 66mS sampai 102mS). Pulsa tersebut dihasilkan oleh timer 16 bit di dalam prosesor, yang memberikan resolusi 1uS.

Selain PWM, CMPS03 juga dilengkapi dengan interface I2C yang dapat digunakan untuk membaca data arah dalam bentuk data serial. Pada mode 8 bit, arah utara ditunjukkan dengan data 255

dengan resolusi 1,40625 derajat/bit. Pada mode 16 bit, arah utara ditunjukkan dengan data 65535 sehingga resolusinya menjadi 0,0055 derajat/bit.

I2C Interface

Pin 2 dan 3 adalah jalur komunikasi I2C dan dapat digunakan untuk membaca data arah (bearing). Jika jalur I2C tidak digunakan, maka pin ini harus di pull up (ke +5V) melalui resistor yang nilainya sekitar 47K, nilai resistor tidak kritikal.



Gambar 2. I2C Communication Protocol

I2C communication protocol dimulai dengan mengirimkan start bit, address modul digital compass dengan read/write low (0xC0), kemudian nomor register yang akan dibaca. Selanjutnya diikuti dengan start bit lagi, address modul digital compass dengan read/write high (0xC1). Selanjutnya anda bisa membaca satu atau dua register (8 bit atau 16 bit). Untuk register 16 bit, yang pertama kali dibaca adalah high byte. CMPS03 memiliki 16 register sesuai tabel berikut :

Tabel 1. Register CMPS03

Register	Function
0	Software Revision Number
1	Compass Bearing as a byte, i.e. 0-255 for a full circle
2,3	Compass Bearing as a word, i.e. 0-3599 for a full circle, representing 0-359.9 degrees.
4,5	Internal Test - Sensor1 difference signal - 16 bit signed word
6,7	Internal Test - Sensor2 difference signal - 16 bit signed word
8,9	Internal Test - Calibration value 1 - 16 bit signed word
10,11	Internal Test - Calibration value 2 - 16 bit signed word
12	Unused - Read as Zero
13	Unused - Read as Zero
14	Unused - Read as Zero
15	Calibrate Command - Write 255 to perform calibration step. See text.

Register 0 adalah Software revision number. Register 1 adalah data arah yang diubah dalam nilai 0-255. Dalam aplikasinya ini lebih mudah dibandingkan nilai 0-360 karena memerlukan dua byte. Untuk resolusi yang lebih tinggi pada Register 2 dan 3 akan menyimpan data arah 16 bit dengan nilai 0-3599. Ini menunjukkan arah 0-359.9°. Register 4 sampai 11 adalah internal test register. Register 12, 13 dan 14 tidak digunakan. Register 15 digunakan untuk melakukan kalibrasi kompas.

Pin 7 adalah input untuk memilih mode operasi 50Hz (low) atau 60Hz (high). Pilihan ini diperlukan untuk melakukan sinkronisasi dengan frekuensi jala-jala PLN jika digital compass menggunakan power supply yang terhubung ke jala-jala PLN. Hal ini dapat menurunkan deviasi pembacaan data arah. Proses konversi didalam sensor, selesai dalam waktu 40mS (50Hz) atau 33.3mS (60Hz). Pin

ini memiliki resistor pullup on board dan dapat dibiarkan tidak terhubung (unconnected) untuk operasi 60Hz.

Pin 6 digunakan untuk kalibrasi digital compass. Pin ini memiliki resistor pullup on board dan dapat dibiarkan tidak terhubung setelah proses kalibrasi.

2. Kalibrasi

Kalibrasi hanya perlu dilakukan sekali, data kalibrasi akan disimpan didalam EEPROM pada chip PIC16F872. Anda tidak perlu melakukan kalibrasi lagi setiap kali modul diaktifkan.

Modul digital compass harus tetap dalam posisi horizontal terhadap permukaan bumi dengan sisi komponen berada dibagian atas. Jauhkan modul dari metal, terlebih lagi dari objek yang mengandung magnet.

Kalibrasi Rev 3 Software

▪ **Kalibrasi Dengan Metode I2C**

Untuk mengkalibrasi kompas menggunakan I2C, anda hanya perlu menuliskan data 255 ke register 15 dan memutar modul kompas secara perlahan sampai 360 derajat. Menuliskan 0 pada register 15 akan menyimpan nilai kalibrasi kedalam EEPROM internal prosesor. Pembacaan dilakukan oleh prosesor pada empat titik arah kompas dan nilai ini digunakan untuk menghasilkan nilai kalibrasi. Register 14 membaca 255 selama operasi normal. Dan membaca 0 pada mode kalibrasi dan 255 lagi ketika 4 titik arah kompas telah diukur. Register 14 kemudian akan menunjukkan bahwa 4 titik telah didapat. Kemudian nilai 0 dapat dituliskan ke register 15 untuk menyimpan data kalibrasi dan kompas akan kembali ke operasi normal.

Sangat penting untuk memutar kompas secara sangat perlahan selama proses kalibrasi untuk menghindari kehilangan titik arah kompas yang diperlukan dan kompas harus tetap horizontal untuk memastikan akurasi proses kalibrasi.

▪ **Kalibrasi Dengan Metode Pin**

Pin 5 dan 6 digunakan untuk mengkalibrasi kompas. Pin 6 dibuat low dan putar kompas 360 derajat secara sangat perlahan. Setelah itu Pin 6 dibuat high agar data kalibrasi tersimpan didalam EEPROM internal prosesor. Pin 5 akan high dalam kondisi normal, dan akan low selama proses kalibrasi. Pin 5 akan high kembali ketika 4 titik arah kompas telah diukur.

Kalibrasi Rev 7 Software

Pin 5 (CalDone) dan register 14 (Calibration Done Flag) tidak digunakan pada Rev 7 software atau CMPS03. Pin 5 harus dibiarkan tidak terhubung dan register 14 diabaikan. Pada saat mengkalibrasi kompas, anda harus tahu dengan pasti arah Utara, Timur, Selatan dan Barat. Jangan menebak-nebak. Gunakan magnetic compass.

▪ **Kalibrasi Dengan Metode I2C**

Untuk mengkalibrasi dengan I2C, anda hanya perlu menuliskan 255 (0xFF) ke register 15 untuk tiap 4 titik arah, Utara, Timur, Selatan dan Barat. Secara internal, nilai 255 akan clear secara otomatis setelah semua titik terkalibrasi. Berikut adalah langkah-langkah kalibrasi :

1. Pastikan kompas pada posisi rata, hadapkan kearah Utara, tuliskan 255 ke register 15.
2. Pastikan kompas pada posisi rata, hadapkan kearah Timur, tuliskan 255 ke register 15.
3. Pastikan kompas pada posisi rata, hadapkan kearah Selatan, tuliskan 255 ke register 15.
4. Pastikan kompas pada posisi rata, hadapkan kearah Barat, tuliskan 255 ke register 15.

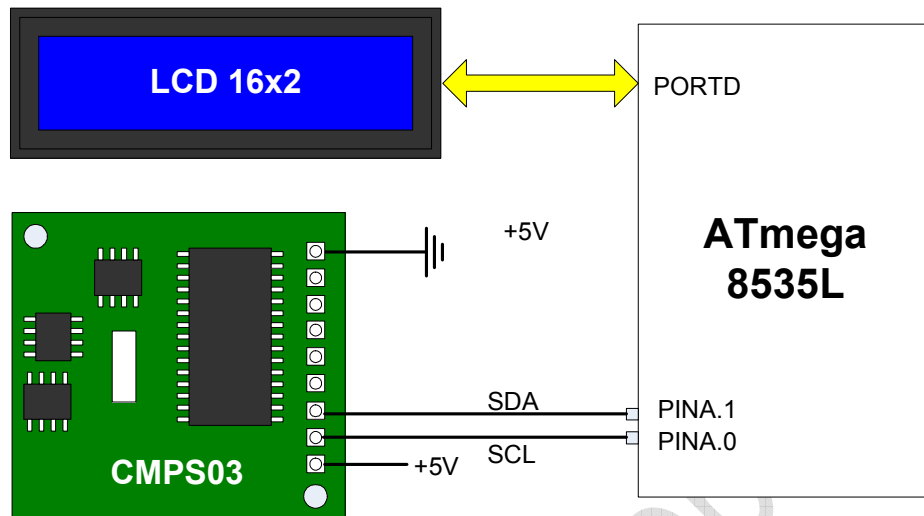
▪ **Kalibrasi Dengan Metode Pin**

Pin 6 dapat dihubungkan ke 0V (Ground) dengan sebuah push button switch. Dan alakukan langkah-langkah berikut :

1. Pastikan kompas pada posisi rata, hadapkan kearah Utara, tekan switch dan lepaskan.
2. Pastikan kompas pada posisi rata, hadapkan kearah Timur, tekan switch dan lepaskan.
3. Pastikan kompas pada posisi rata, hadapkan kearah Selatan, tekan switch dan lepaskan.
4. Pastikan kompas pada posisi rata, hadapkan kearah Barat, tekan switch dan lepaskan.

3. Contoh Aplikasi Pembacaan Data 8 bit Digital Compass

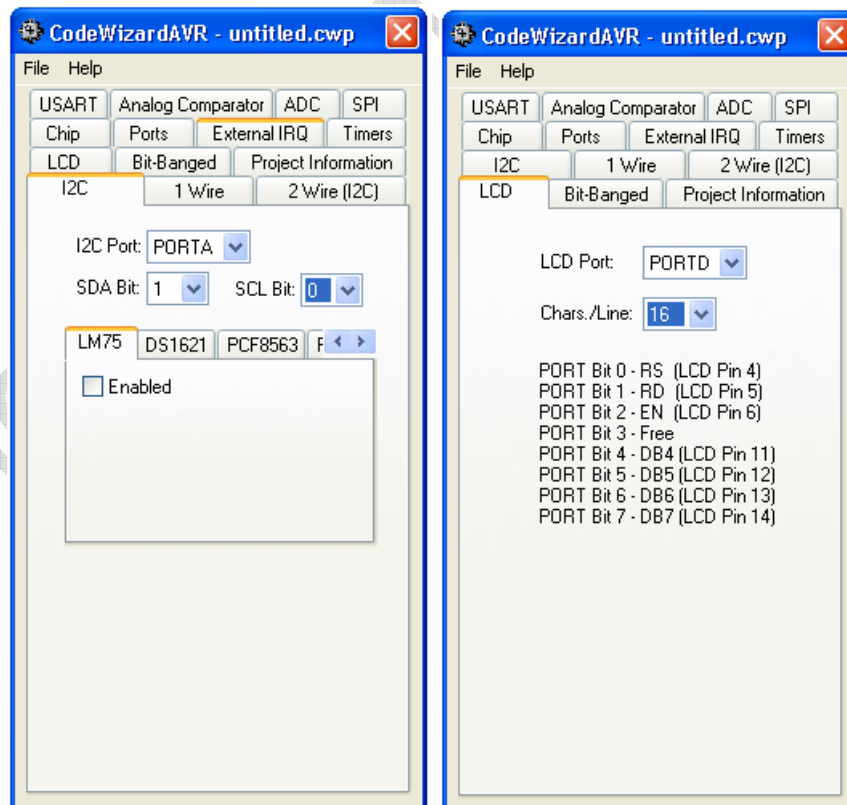
Pada aplikasi ini mikrokontroler akan membaca data arah dari CMPS03 dan ditampilkan ke LCD.



Gambar 3. Rangkaian Aplikasi Digital Compass

Berikut adalah langkah percobaannya :

1. Lakukan setting codewizard sebagai berikut :



2. Buat program sebagai berikut :

```
/******
This program was produced by the
CodeWizardAVR V1.24.0 Standard
Automatic Program Generator
© Copyright 1998-2003 HP InfoTech s.r.l.
http://www.hpinfotech.ro
e-mail:office@hpinfotech.ro

Project :
Version :
Date    : 11/12/2007
Author  : hendawan
Company :
Comments:

Chip type      : ATmega8535L
Program type   : Application
Clock frequency : 11,059200 MHz
Memory model   : Small
External SRAM size : 0
Data Stack size : 128
*****/

#include <mega8535.h>
#include <stdio.h>
// I2C Bus functions
#asm
    .equ __i2c_port=0x1B
    .equ __sda_bit=1
    .equ __scl_bit=0
#endasm
#include <i2c.h>

// Alphanumeric LCD Module functions
#asm
    .equ __lcd_port=0x12
#endasm
#include <lcd.h>
unsigned char Msg1[16];
unsigned char Msg2[16];

// Declare your global variables here
/* read a byte from the COMPASS */
unsigned char data;
float posisi;
void compass_read()
{
    i2c_start();
    i2c_write(0xC0);
    i2c_write(0x01);
    i2c_start();
    i2c_write(0xC1);
    data=i2c_read(0);
    i2c_stop();
}
void main(void)
{
    // Declare your local variables here

    // Input/Output Ports initialization
    // Port A initialization
    // Func0=In Func1=In Func2=In Func3=In Func4=In Func5=In Func6=In Func7=In
    // State0=T State1=T State2=T State3=T State4=T State5=T State6=T State7=T
    PORTA=0x00;
    DDRA=0x00;

    // Port B initialization
    // Func0=In Func1=In Func2=In Func3=In Func4=In Func5=In Func6=In Func7=In
```

```

// State0=T State1=T State2=T State3=T State4=T State5=T State6=T State7=T
PORTB=0x00;
DDRB=0x00;

// Port C initialization
// Func0=In Func1=In Func2=In Func3=In Func4=In Func5=In Func6=In Func7=In
// State0=T State1=T State2=T State3=T State4=T State5=T State6=T State7=T
PORTC=0x00;
DDRC=0x00;

// Port D initialization
// Func0=In Func1=In Func2=In Func3=In Func4=In Func5=In Func6=In Func7=In
// State0=T State1=T State2=T State3=T State4=T State5=T State6=T State7=T
PORTD=0x00;
DDRD=0x00;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=FFh
// OC0 output: Disconnected
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer 1 Stopped
// Mode: Normal top=FFFFh
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=0x00;
MCUCSR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
// Analog Comparator Output: Off
ACSR=0x80;
SFIOR=0x00;

// I2C Bus initialization
i2c_init();

```

```
// LCD module initialization
lcd_init(16);

while (1)
{
    // Place your code here
    compass_read();
    posisi=data;
    sprintf(Msg1,"Bearing=%d ",data);
    sprintf(Msg2,"t=%f.%u\xdf",posisi%10);
    lcd_gotoxy(0,0);lcd_puts(Msg1);
    lcd_gotoxy(0,1);lcd_puts(Msg2);
};
}
```


DAFTAR PUSTAKA

Atmega8535(L) Datasheet, Atmel Corporation
CMPS03 Datasheet
<http://www.robot-electronics.co.uk/>

Hendawan Soebhakti