

CMPS03 – Devantech Magnetic Compass

AN-09

Oleh: Tim Digiware dan Hadid T.B. - Sihmanto - Idam F.R.
(Institut Teknologi Sepuluh Nopember)

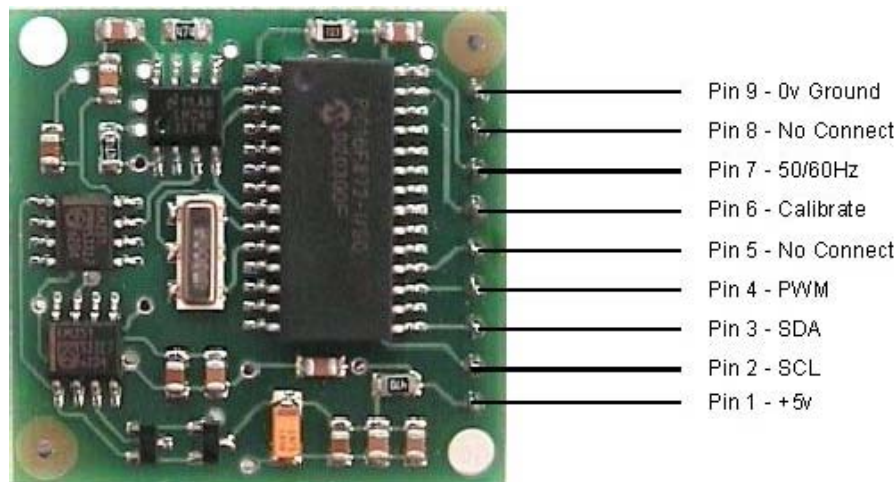
Navigasi sangatlah penting untuk semua benda bergerak, baik manusia maupun robot. Salah satu alat untuk membantu navigasi adalah kompas. Dan aplikasi kali ini akan membuat sebuah kompas digital *portable* dengan menggunakan CMPS03 – Devantech Magnetic Compass yang memiliki resolusi hingga 0,1 derajat dan 2 pilihan antarmuka yaitu I²C atau PWM. Modul CMPS03 ini dapat juga digunakan untuk keperluan robotika.

Komponen yang diperlukan:

- 1 DT-51 Low Cost Nano System
- 1 Character LCD 8x2 (atau ukuran lain dengan driver HD44780 / kompatibel)
- 1 CMPS03 – Devantech Magnetic Compass
- 1 Resistor variabel 2k ohm
- 2 Resistor 1k ohm
- 2 Resistor 47k ohm
- 1 *Tactile switch*
- 2 Jumper
- 1 Header 2x3

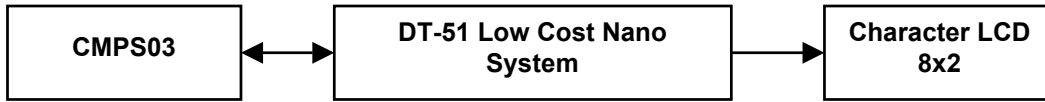
Spesifikasi untuk modul CMPS03 – Devantech Magnetic Compass, yaitu:

- Catu daya : +5 VDC,
- Konsumsi arus : 15 mA,
- Antarmuka : I²C atau PWM,
- Akurasi : 3-4 derajat,
- Resolusi : 0,1 derajat,
- Waktu konversi : 40ms atau 33,3ms dapat dipilih,
- Telah dikalibrasi pada daerah dengan sudut inklinasi 67 derajat.



Gambar 1
Alokasi Pin dari CMPS03

Adapun blok diagram sistem secara keseluruhan adalah sebagai berikut:



Gambar 2
Blok Diagram AN-09

Hubungan antara modul-modul tersebut adalah sebagai berikut:

DT-51 Low Cost Nano System	CMPS03
VCC	+ 5V
GND	GND
P3.0*	SCL
P3.2*	SDA
P3.4*	PWM

* Pin ini tidak mutlak dan dapat diganti pin lain dengan cara mengubah program

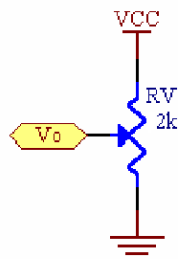
Tabel 1
Hubungan DT-51 Low Cost Nano System dengan CMPS03 secara I²C dan PWM

DT-51 Low Cost Nano System	Character LCD 8x2
VCC	V _{DD}
GND	V _{SS}
-	V _O **
GND	R/W
P1.6*	RS
P1.7*	E
P1.4*	DB4
P1.5*	DB5
P1.2*	DB6
P1.3*	DB7

* Pin ini tidak mutlak dan dapat diganti pin lain dengan cara mengubah program

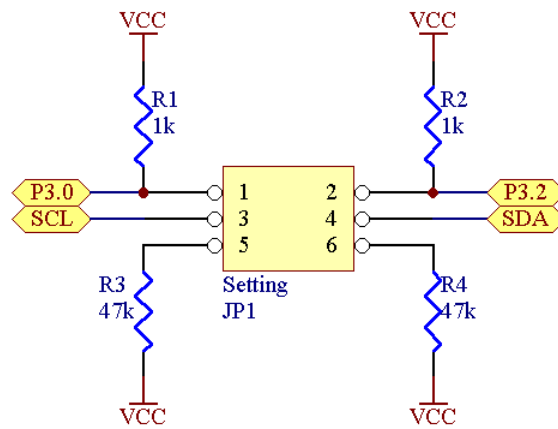
** Pin V_O dihubungkan ke resistor variabel seperti pada Gambar 3

Tabel 2
Hubungan DT-51 Low Cost Nano System dengan Character LCD 8x2



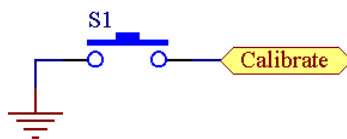
Gambar 3
Rangkaian Kontras untuk Character LCD 8x2

Selain mengikuti tabel hubungan di atas, buatlah rangkaian jumper seperti pada Gambar 4. Lalu pada saat menggunakan komunikasi I²C, hubungkan pin 2 & 4 serta pin 1 & 3 dari JP1 (Gambar 4) untuk memberi resistor *pullup* 1k ohm pada jalur SCL & SDA. Sedangkan pada saat menggunakan metode PWM, hubungkan pin 3 & 5 serta 4 & 6 dari JP1 (Gambar 4) untuk memberi resistor pullup sebesar 47k ohm pada jalur SCL & SDA.

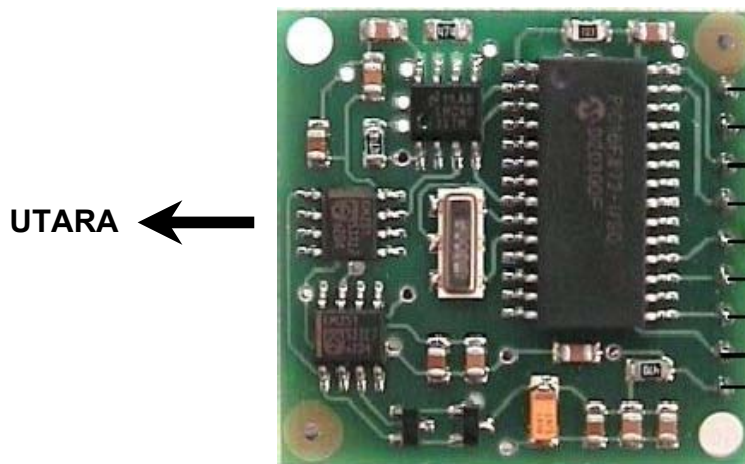


Gambar 4
Rangkaian Jumper untuk Pemilihan Resistor *Pullup* pada Jalur SCL & SDA

Lepaslah AT89C2051 dari *board* DT-51 Low Cost Nano System. Dan programlah i2c_final.hex (untuk metode komunikasi I²C) atau pwm_final.hex (untuk metode PWM) ke dalam mikrokontroler AT89C2051 menggunakan TOP2004 Universal Programmer atau DT-HiQ Programmer atau DT-51 MinSys v3.0 + DT-51 ProgPAL atau divais paralel programmer lain yang mendukung AT89C2051. Pasanglah kembali AT89C2051 yang telah diprogram ke *board* DT-51 Low Cost Nano System dan berilah catu daya yang sesuai pada rangkaian.



Gambar 5
Rangkaian *Tactile Switch* untuk Proses *Calibrate*



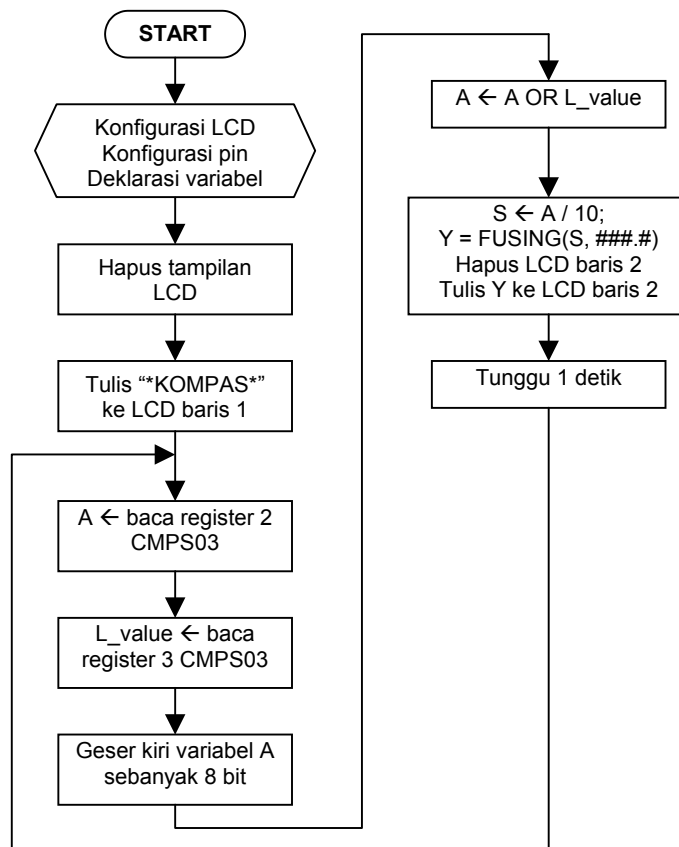
Gambar 6
Orientasi CMPS03 yang Menghasilkan Pembacaan Sudut 0°

Modul CMPS03 telah terkalibrasi di pabriknya namun karena lokasi pabriknya berbeda dengan Indonesia dalam hal sudut inklinasinya, maka modul ini perlu dikalibrasi ulang. Cara mengkalibrasi CMPS03 ada dua cara, yaitu dengan metode I²C atau pin (manual). Dalam aplikasi ini dipilih kalibrasi dengan metode pin (manual) karena dinilai lebih mudah dan efisien. Berikut ini adalah langkah-langkahnya:

1. Gunakan rangkaian *tactile switch* seperti pada Gambar 5.
2. Posisikan orientasi utara dari CMPS03 (Gambar 6) ke arah utara bumi yang sebenarnya lalu tekan *tactile switch*.
3. Putar secara perlahan-lahan sampai orientasi utara dari CMPS03 menuju ke arah timur bumi, lalu tekan *tactile switch*.
4. Putar secara perlahan-lahan sampai orientasi utara dari CMPS03 menuju ke arah selatan bumi, lalu tekan *tactile switch*.
5. Putar secara perlahan-lahan sampai orientasi utara dari CMPS03 menuju ke arah barat bumi, lalu tekan *tactile switch*.
6. Periksa apakah kompas telah menampilkan arah yang benar sesuai dengan arah sebenarnya. Jika belum sesuai ulangi lagi mulai langkah 1.
7. Jika penunjukan sudah sesuai dengan arah sebenarnya, maka CMPS03 dinyatakan telah terkalibrasi dengan baik.

Kalibrasi ini hanya dilakukan sekali saja, karena hasil dari pengkalibrasian disimpan dalam EEPROM yang terdapat pada CMPS03. Untuk penggunaan selanjutnya (pada lokasi dengan sudut inklinasi sama), tidak perlu dilakukan kalibrasi ulang.

Flowchart program i2c_final.bas untuk metode I²C adalah sebagai berikut:



Gambar 7
Flowchart Program i2c_final.bas

Program utama `i2c_final.bas` akan diproses sebagai berikut:

1. Proses yang pertama dilakukan adalah konfigurasi LCD, konfigurasi pin I/O (Sdl & Sca) untuk jalur komunikasi I²C, serta deklarasi variabel yaitu antara lain:
 - `Adress` = variabel I/O untuk menampung alamat *register* yang akan dibaca pada pemanggilan prosedur `Read_compass`,
 - `Value` = variabel I/O untuk menampung hasil pembacaan *register* pada pemanggilan prosedur `Read_compass`,
 - `L_value` = variabel bertipe *byte* untuk menampung 8 bit data LSB hasil pembacaan CMPS03,
 - `A` = variabel bertipe *word* untuk menampung 8 / 16 bit data hasil pembacaan CMPS03,
 - `Y` = variabel bertipe *string* untuk menampung data yang akan dituliskan ke LCD.
 - `S` = variabel bertipe *single* untuk menampung data bernilai *real* yang merupakan hasil bagi dari proses pembagian variabel `A`.
2. Program akan menghapus tampilan LCD, lalu menampilkan “KOMPAS” pada LCD baris 1.
3. Setelah itu dilakukan pembacaan data pada *register* 2 dan 3 dari modul CMPS03 (Tabel 3) menggunakan prosedur `Read_compass`. Lalu hasil pembacaan tersebut diletakkan pada variabel `A` (data dari *register* 2 – MSB) dan `L_value` (data dari *register* 3 – LSB).
4. Program melakukan penggeseran nilai dalam variabel `A` sebanyak 8 bit ke kiri. Lalu variabel `A` di-OR-kan dengan variabel `L_value` dan hasilnya disimpan ke dalam variabel `A`.
5. Variabel `A` dibagi dengan 10 dan hasil baginya disimpan ke dalam variabel `S`. Lalu dengan menggunakan fungsi `FUSING`, dilakukan perubahan format data dari *real* (variabel `S`) ke dalam bentuk *string* dan disimpan ke dalam variabel `Y`.
6. Tampilan LCD pada baris 2 dihapus, lalu ditulisi data hasil pembacaan kompas (data dari variabel `Y`).
7. Tunggu selama 1 detik lalu kembali ke langkah .

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 Undefined
15	Calibrate Command - Write 255 to perform calibration step. See text.

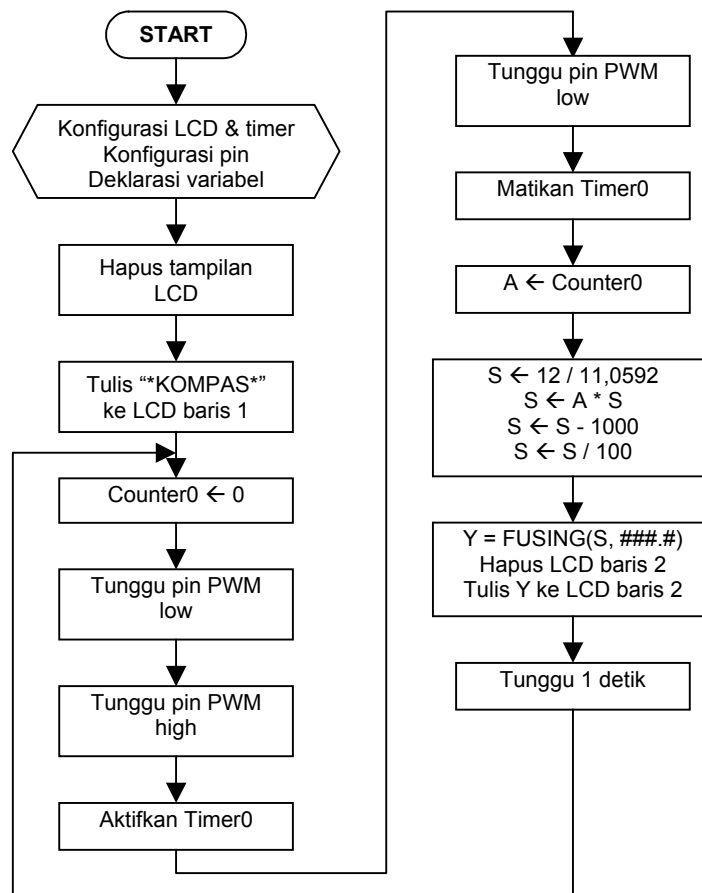
Tabel 3
Alokasi Internal Register CMPS03

Program `pwm_final.bas` secara garis besar akan diproses sebagai berikut:

1. Proses yang pertama dilakukan adalah konfigurasi LCD (alokasi pin dan ukuran LCD); konfigurasi Timer0 (mode 1 dan *gate-internal*); konfigurasi pin I/O (input sinyal PWM); serta deklarasi variabel yaitu antara lain:
 - `A` = variabel bertipe *word* untuk menampung 16 bit data Timer0 yang merupakan hasil pembacaan CMPS03 secara PWM,
 - `Y` = variabel bertipe *string* untuk menampung data yang akan dituliskan ke LCD.
 - `S` = variabel bertipe *single* untuk menampung data bernilai *real* yang merupakan hasil pengolahan data dari variabel `A`.
2. Program akan menghapus tampilan LCD, lalu menampilkan “KOMPAS” pada LCD baris 1.

3. Setelah itu program mengisi *register* Timer0 dengan data bernilai 0. Lalu menunggu perubahan transisi naik pada pin input sinyal PWM. Setelah terdeteksi perubahan tersebut Timer0 diaktifkan, lalu menunggu lagi pin PWM berubah menjadi logika 0 dan kemudian mematikan Timer0.
4. Program membaca data dari register Timer0 dan diletakkan ke dalam variabel A.
5. Program melakukan normalisasi data hasil pengukuran Timer0 (variabel A) disesuaikan dengan nilai *crystal* yang digunakan mikrokontroler dan hasilnya disimpan ke dalam variabel S. Lalu variabel S dikurangi dengan nilai *offset* yaitu 1000 μ s dan kemudian dibagi 100.
6. Lalu dengan menggunakan fungsi FUSING, dilakukan perubahan format data dari *real* (variabel S) ke dalam bentuk *string* dan disimpan ke dalam variabel Y.
7. Tampilan LCD pada baris 2 dihapus, lalu ditulisi data hasil pembacaan kompas (data dari variabel Y).
8. Tunggu selama 1 detik lalu kembali ke langkah .

Flowchart pogram pwm_final.bas untuk metode PWM adalah sebagai berikut:



Gambar 8
Flowchart Program pwm_final.bas

Catatan:

Pada metode PWM akan diperoleh pembacaan sudut maksimum adalah 357 derajat bukan 359 derajat, hal ini karena pada program pwm_final.bas belum diberi "fiddle factor" yang berguna untuk mengurangi selisih antara modul CMPS03 dan pengukuran / osilator mikrokontroler.

Listing program terdapat pada AN-09.ZIP.