DT-AVR

AN203 – Kontrol AX-12 Dynamixel Servo dengan DT-AVR Low Cost Micro System

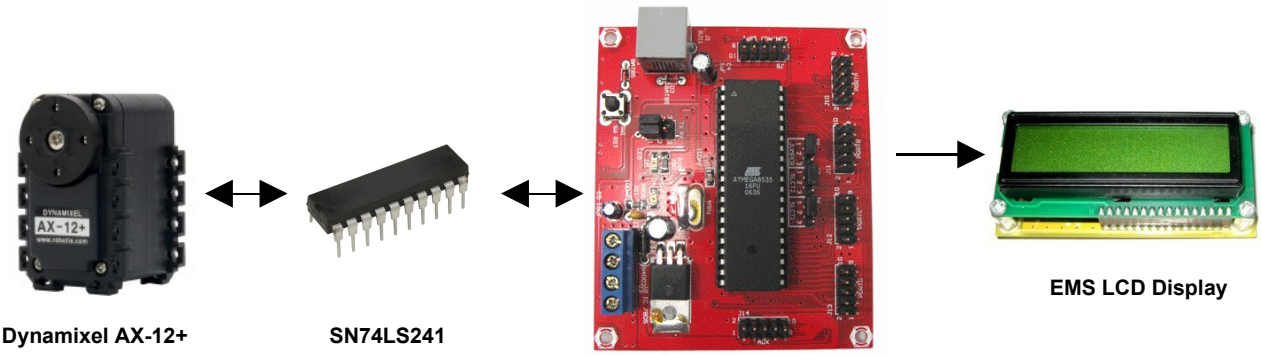
Oleh: Tim IE

Servo atau motor servo merupakan aktuator yang sangat penting dalam dunia robotika. AX-12/AX-12+/AX-12A termasuk dalam jenis servo pintar *(Smart Servo)* karena kelengkapan fitur dan kecanggihannya.Keluarga servo ini dapat melaporkan posisi *shaft,* tegangan input, kecepatan, beban, dan temperatur secara real-time. Pada aplikasi ini servo yang digunakan adalah AX-12+.
Sebagai kontroler utama pada aplikasi ini, digunakan DT-AVR Low Cost Micro System yang berbasis Mikrokontroler ATmega8535 dari Innovative Electronics. Sebagai media penampil data digunakan EMS LCD Display yang merupakan modul LCD Character 16x2 yang kompatibel dengan DT-AVR LCMS dan juga merupakan produk dari Innovative Electronics.
Baudrate komunikasi AX-12+ dapat diatur sesuai kebutuhan, namun secara *default* baudratenya adalah sebesar 1000000 bit per detik atau 1Mbps. Oleh karena itu, pada DT-AVR LCMS dilakukan penggantian *Crystal clock* dari 4Mhz menjadi 8Mhz agar dapat menggunakan baudrate UART sebesar 1Mbps dengan *error 0%*.
Jalur Data pada AX-12+ memiliki spesifikasi UART TTL *half duplex,* dimana data *Instruction packet dan Status Packet* berada pada satu kabel yang sama. Mekanisme pengarahan data yang dikirim dan data yang diterima dari DT-AVR LCMS ke AX-12+ dan sebaliknya dilakukan dengan bantuan IC *Octal Buffer and Line Driver* 74LS241.
AN203 ini dikembangkan dalam bahasa C dengan IDE Eclipse C/C++ yang telah terinstal plugin AVR dan terintegrasi dengan WinAVR (AVR-GCC).

Berikut adalah perlengkapan yang diperlukan dalam aplikasi ini :

- 1x DT-AVR Low Cost Micro System (dengan Crystal 8MHz)
- 1x EMS LCD Display
- 1x AX-12+ servo
- 1x IC SN74LS241 dan 1x Resistor 10K Ohm
- Beberapa kabel jumper dan 2x kabel servo konektor 3 pin Dynamixel

Adapun blok diagram dari aplikasi ini adalah sebagai berikut :



Gambar 1
Blok Diagram AN203

Hubungan antar modul adalah sebagai berikut :

DT-AVR Low Cost Micro System	SN74LS241
GND (J13 pin 1)	GND (Pin 10)
VCC (J13 pin 2)	VCC (Pin 20)
PORTD.0/RXD (J13 Pin 3) **	1Y1 (Pin 18)***
PORTD.1/TXD (J13 Pin 4) **	2A4 (Pin 17)***
PORTD.7/Direction Control (J13 Pin 10)*	1G (Pin 1)**
	2G (Pin 19)**

*) Pin tidak mutlak, dapat diganti dengan pin lain dengan penyesuaian program
**) Pin ini mutlak, tidak dapat diganti dengan pin yang lain.
***) Pin tidak mutlak, dapat diganti dengan pin lain yang sesuai fungsinya.

Tabel 1
Hubungan DT-AVR LCMS dengan SN74LS241

SN74LS241	AX-12+
GND (Pin 10)	GND
1A1 (Pin 2)***	DATA (dengan *pull up* eksternal Resistor 10K Ohm)
2Y4 (Pin 3)***	

***) Pin tidak mutlak, dapat diganti dengan pin lain yang sesuai fungsinya.

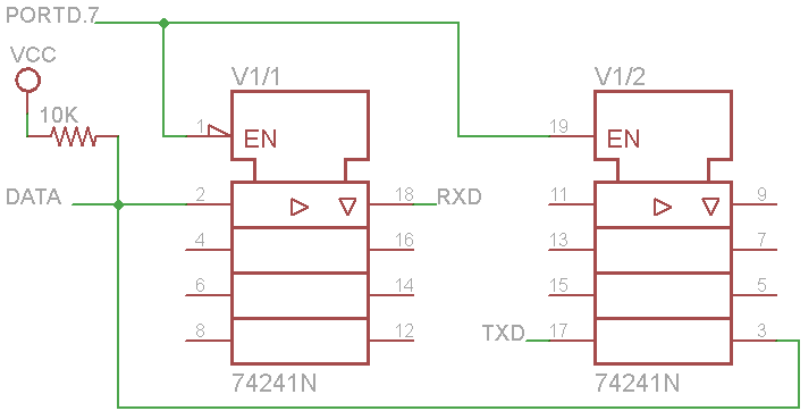
Tabel 2
Hubungan SN74LS241 dengan AX-12+

DT-AVR Low Cost Micro System	EMS LCD Display
GND (J10 pin 1)	GND (J3 pin 1)
VCC (J10 pin 2)	+5V (J3 pin 2)
PORTA.0 (J10 Pin 3)*	RS (J3 pin 3)
PORTA.1 (J10 Pin 4)*	R/W (J3 pin 4)
PORTA.2 (J10 Pin 5)*	E (J3 pin 5)
PORTA.3 (J10 Pin 6)*	BL (J3 pin 6)
PORTA.4 (J10 Pin 7)*	DB4 (J3 pin 7)
PORTA.5 (J10 Pin 8)*	DB5 (J3 pin 8)
PORTA.6 (J10 Pin 9)*	DB6 (J3 pin 9)
PORTA.7 (J10 Pin 10)*	DB7 (J3 pin 10)

*) Pin tidak mutlak, dapat diganti dengan pin lain dengan penyesuaian program

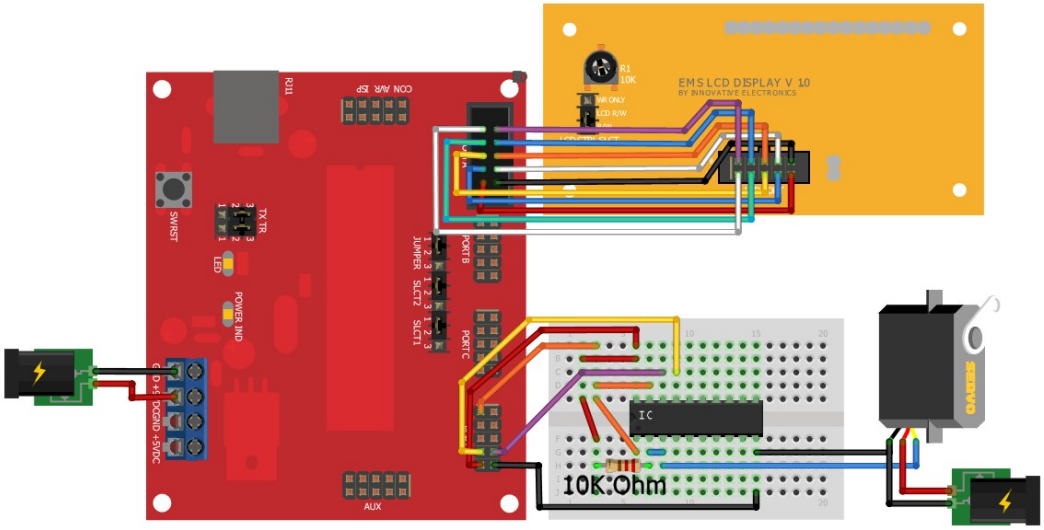
Tabel 1
Hubungan DT-AVR LCMS dengan EMS LCD Display

Skema koneksi SN74LS241 dengan modul-modul yang lain adalah sebagai berikut :



Gambar 2
Skema koneksi SN74LS241 dengan AX-12+ dan DT-AVR LCMS

Ilustrasi koneksi antar modul pada AN203 ini adalah sebagai berikut :



Gambar 3
Ilustrasi AN203

Modul – modul diatas perlu dikonfigurasi terlebih dahulu agar dapat bekerja sesuai dengan yang diharapkan. Berikut ini adalah langkah – langkah konfigurasi yang perlu dilakukan :

- DT-AVR Low Cost Micro System
  - Mikrokontroler ATmega8535 pada DT-AVR Low Cost Micro System menggunakan osilator eksternal berupa crystal dengan nilai frekuensi 4 Mhz. Untuk mendukung baudrate komunikas 1Mbps maka diperlukan penggantian Crystal Oscillator dengan frekuensi 8MHz maupun 16MHz, pada AN ini digunakan XTAL 8MHz. Silahkan melakukan pengaturan *fusebit* pada ATmega8535 dahulu agar dapat bekerja dengan osilator eksternal 8 MHz. Informasi lebih detil mengenai pengaturan fusebit terdapat pada AN177.
  - Komunikasi antara ATmega8535 dengan AX-12+ dilakukan menggunakan komunikasi serial UART TTL. Maka dari itu perlu dilakukan pengaturan jumper J4 dan J5 pada posisi 2-3.

1	2	3	J4
1	2	3	J5

Gambar 4
Pengaturan Jumper J4 dan J5 pada DT-AVR Low Cost Micro System

Setelah menghubungkan modul-modul tersebut menggunakan kabel jumper, lakukan pengecekan kembali menggunakan *multimeter* untuk memastikan koneksi antar modul telah terhubung dengan baik. Pastikan juga bahwa tidak terjadi hubungan singkat antara VCC dan GND sebelum memberikan catu daya.
Apabila konfigurasi di atas telah selesai, lakukan langkah – langkah berikut ini :

1. Berikan catu daya +9V DC s.d. +12V DC pada terminal biru J2 DT-AVR LCMS (**perhatikan polaritas catu daya, Jika polaritas terbalik dapat menyebabkan kerusakan modul**).
2. Hubungkan *programmer* mikrokontroler AVR yang mendukung fitur ISP dengan DT-AVR Low Cost Micro System, seperti DT-HiQ AVR In System Programmer, DT-HiQ AVR USB ISP, atau *programmer* lainnya.
3. *Download* file dengan ekstensi .hex (DynamixelControl.hex) yang berada di dalam folder (AN203\DynamixelControl\Release) pada DT-AVR Low Cost Micro System.
4. Berikan input catu daya untuk servo AX-12+ melalui konektor 3 pin yang ada pada *body* servo. Catu daya disarankan memiliki spesifikasi tegangan yang stabil dan kemampuan *supply* arus yang cukup. Tegangan Operasional AX-12+ adalah rentang 9V s.d. 12V.
5. Konsumsi Arus AX-12+ saat standby sekitar 50mA dan saat beroperasi mengkonsumsi arus maksimal hingga 900mA.

Protokol komunikasi Dynamixel AX-12+

Protokol komunikasi pada Dynamixel Ax-12+ terbagi menjadi dua jenis yaitu data yang dikirimkan dari kontroler ke servo *(Instruction Packet)* dan data respon yang dikirim balik oleh servo kepada kontroler *(Status packet)*. Keterangan lengkap mengenai paket *Instruction*, *Status*, dan *Parameter* dapat dilihat pada manual dari produk AX-12+. Kedua paket data tersebut memiliki format sebagai berikut.

- **Instruction Packet**

Merupakan paket yang dikirimkan oleh kontroler utama kepada servo untuk mengirimkan perintah-perintah. Struktur dari *Instruction packet* adalah sebagai berikut:

0XFF	0XFF	ID	LENGTH	INSTRUCTION	PARAMETER1	…	PARAMETER N	CHECK SUM

Keterangan :

0xFF 0xFF : Merupakan indikator awal dari sebuah paket.

ID : Merupakan nomor ID/Identitas unique dari servo dynamixel.
Terdapat 254 nilai ID yang tersedia mulai dari 0x00 sampai dengan 0xFD. Ada nomor ID khusus yang disebut broadcasting ID yaitu nomor 0xFE. Paket yang dikirmkan dengan ID Broadcasting akan diterima dan dilaksanakan oleh semua servo yang terhubung. Paket yang terkirim dengan Broadcasting ID tidak akan menghasilkan Respon Status Packet.

LENGTH : Merupakan panjang dari paket dimana nilainya adalah “Jumlah parameter(N) + 2”.

INSTRUCTION : Instruksi untuk aktuator Dynamixel yang akan dilakukan.

PARAMETER0...N : Digunakan jika ada informasi tambahan yang butuh untuk dikirmkan selain dari Instruksi itu sendiri.

CHECK SUM : Merupakan data yang dihitung sebagai error checking. Metode penghitungan check sum adalah sebagai berikut :

$$\text{Check Sum} = \sim(\text{ID} + \text{Length} + \text{Instruction} + \text{Parameter1} + \ldots + \text{Parameter N})$$

Jika nilai yang terhitung lebih besar dari 255, maka byte terbawahlah yang ditentukan sebagai nilai check sum.
~ mewakili operasi logic NOT.

- **Status Packet/Return Packet**

Merupakan paket respon dari servo kepada kontroler utama setelah menerima sebuah *Instruction Packet*. Struktur dari *Status Packet* adalah sebagai berikut.

0XFF	0XFF	ID	LENGTH	ERROR	PARAMETER1	PARAMETER2	…	PARAMETER N	CHECK SUM

Keterangan :

0xFF 0xFF : Merupakan indikator awal dari sebuah paket.

ID : Merupakan nomor ID/Identitas unique dari servo dynamixel.
Terdapat 254 nilai ID yang tersedia mulai dari 0x00 sampai dengan 0xFD. Ada nomor ID khusus yang disebut broadcasting ID yaitu nomor 0xFE. Paket yang dikirmkan dengan ID

Broadcasting akan diterima dan dilaksanakan oleh semua servo yang terhubung. Paket yang terkirim dengan Broadcasting ID tidak akan menghasilkan Respon Status Packet.

LENGTH : Merupakan panjang dari paket dimana nilainya adalah "Jumlah parameter(N) + 2".
ERROR : Merupakan *Byte* yang yang merepresentasikan error yang dikirimkan oleh Servo.
INSTRUCTION : Instruksi untuk aktuator Dynamixel yang akan dilakukan.
PARAMETER0...N : Digunakan jika ada informasi tambahan yang butuh untuk dikirmkan selain dari Instruksi itu sendiri.
CHECK SUM : Merupakan yang dihitung sebagai error checking. Metode penghitungan 'Check Sum' adalah sebagai berikut :

Check Sum = ~(ID +Length +Instruction + Parameter1+... Parameter N)

Jika nilai yang terhitung lebih besar dari 255, maka byte terbawahlah yang ditentukan sebagai nilai checksum.
~ mewakili operasi logic NOT.

Library AX-12+ Dynamixel servo

Kumpulan dari *instruction-instruction* Dynamixel AX-12+ telah dirangkum kedalam library program dengan nama *myDynamixel.c dan* file header *myDynamixel.h*. Library ini disusun berdasarkan *library* Arduino *Dynamixel.cpp* dan *Dynamixel.h* yang dibuat oleh Josué Alejandro Savage.

Beberapa fungsi-fungsi yang ada didalamnya sebagai contoh :

- **Dxl_begin()**
  Deskripsi :Inisialisasi komunikasi serial pada kontroler (DT-AVR Low Cost Micro System)
  Sintaks : Dxl_begin(baudrate);
  Parameter: baudrate – kecepatan transmisi serial dalam bps *(bit per second)*
  Contoh :
  Dxl_begin(1000000);
  Akan melakukan inisialisasi komunikasi dengan kecepatan transmisi 1000000 bps.

- **Dxl_move()**
  Deskripsi : Menggerakkan servo kepada posisi yang dikirimkan.
  Sintaks : Dxl_move(ID, Position);
  Parameter:
  ID – Nomor identitas unik yang dimiliki oleh servo AX-12+ Dynamixel.
  Position – Posisi *saft* servo dengan rentang nilai 0 sampai dengan 1023 (0 derajat hingga 300 derajat).
  Contoh :
  Dxl_move(1,400);
  Akan mengirimkan perintah untuk menggerakkan servo dengan ID = 1 pada posisi = 400.

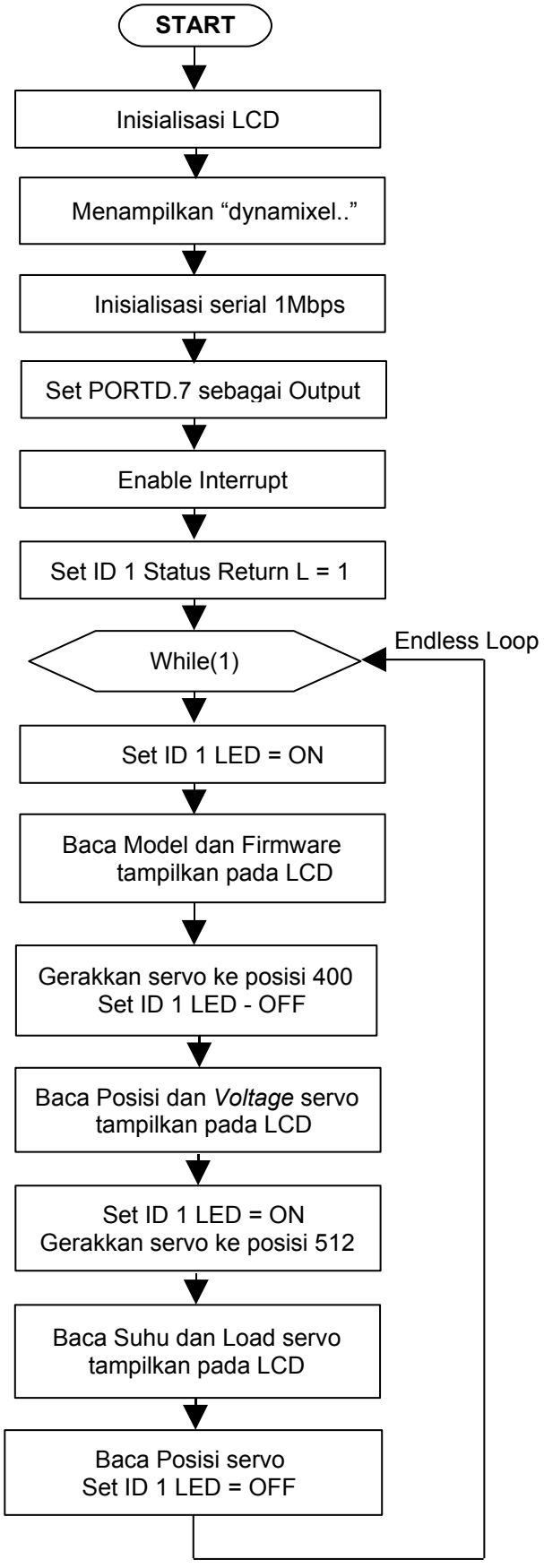
- **Dxl_readPosition()**
  Deskripsi : Membaca posisi servo pada saat itu.
  Sintaks : Dxl_readPosition(ID);
  Parameter: ID – Nomor identitas unik yang dimiliki oleh servo AX-12+ Dynamixel.
  Contoh :
  Dxl_readPosition(1);
  Akan mengirimkan perintah untuk membaca posisi servo dengan ID = 1.
  Respon Status Packet :
  Berisi nilai posisi servo dalam format integer.

Tambahan pada *library* UART

Untuk mendukung *library myDynamixel* yang sumber aslinya berjalan sebagai *library* Arduino diperlukan fungsi khusus pada program UART, beberapa fungsi tersebut antara lain:

- **uart_available()** = berfungsi untuk melihat jumlah byte yang diterima oleh receiver buffer
- **uart_peek()** = berfungsi untuk mengintip *(peek) byte* buffer teratas dari data diterima yang belum dibaca.
- **uart_flush()** = berfungsi membersihkan *Rx buffer* dari data diterima yang belum dibaca.

Adapun alur program dari DynamixelControl.exe adalah sebagai berikut:

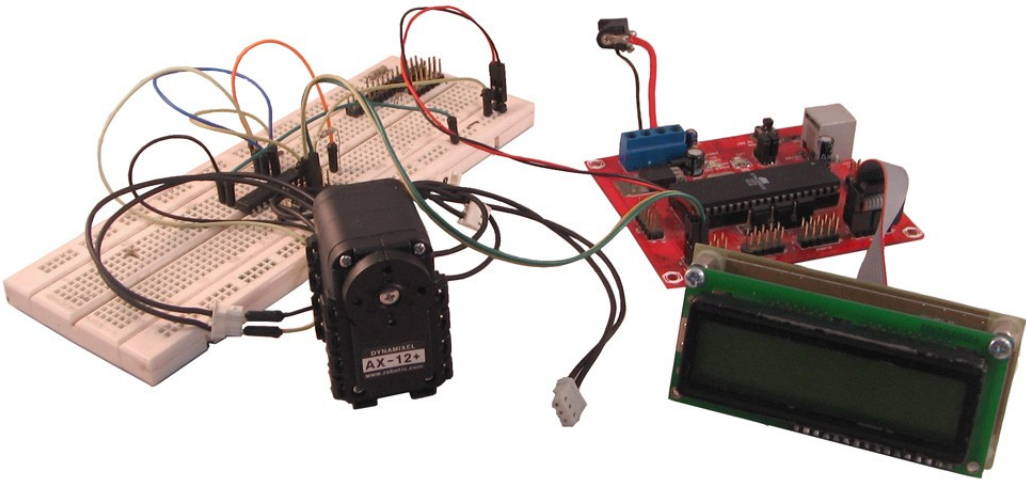


Gambar 5
Diagram alir program DynamixelControl

Penjelasan urutan kerja dari program diatas adalah sebagai berikut :

1. Program melakukan inisialisasi LCD 16x2 pada PORTA.
2. Menampilkan Tampilan pembuka pada LCD.
3. Inisialisasi komunikasi Serial UART dengan baudrate 1Mbps.
4. Set PORTD.7 sebagai output untuk pengontrol pin *data direction.*
5. Meng-*enable* kan *Global Interrupt.*
6. Mengirimkan *Instruction Packet* untuk melakukan set *Status Return Level* = 1, hal ini bertujuan agar Dynamixel AX-12+ hanya memberikan balasan respon untuk *Instruction Read Data*.
7. Masuk kedalam *Superloop,* mengirimkan Instruction Packet pada ID 1 agar LED servo ON.
8. Mengirim *Instruction Packet* untuk membaca Model dan Firmware kemudian menampilkannya ke LCD.
9. Mengirim *Instruction Packet* untuk menggerakkan servo ke posisi 400 dan set LED servo OFF.
10. Mengirim *Instruction Packet* untuk membaca posisi dan tegangan kemudian menampilkannya ke LCD.
11. Mengirim *Instruction Packet* untuk menggerakkan servo ke posisi 512 dan set LED servo ON.
12. Mengirim *Instruction Packet* untuk membaca suhu dan *load* servo kemudian menampilkannya ke LCD.
13. Mengirim *Instruction Packet* untuk membaca posisi servo dan set LED servo OFF.
14. Program berjalan terus dalam *Superloop*.

Gambar koneksi keseluruhan modul dapat dilihat pada **Gambar 6**.



Gambar 6
Rangkaian antar modul pada AN203

Listing program aplikasi ini terdapat pada **AN203.ZIP**

Selamat berinovasi!