

DT-AVR

DT-AVR *Application Note*

Monitoring Suhu Nirkabel

(Bahasa BASIC)

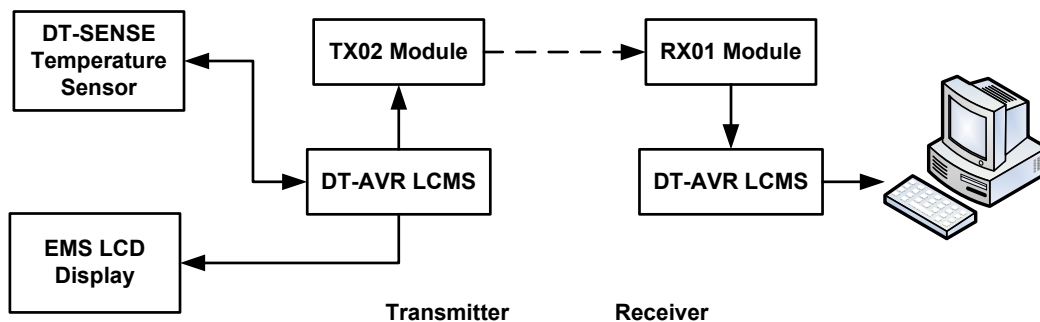
Oleh: Tim IE

Komunikasi data secara nirkabel (*wireless*) saat ini berkembang cukup cepat. Cukup banyak media komunikasi yang bisa digunakan untuk aplikasi komunikasi secara nirkabel seperti cahaya, suara, maupun RF (*Radio Frequency*). Monitoring suhu nirkabel pada aplikasi ini memanfaatkan media komunikasi RF dengan pita frekuensi 915 MHz menggunakan sepasang modul RF yaitu RX01 dan TX02. Sepasang modul ini digunakan untuk mengirimkan data pembacaan suhu oleh DT-SENSE Temperature Sensor. Aplikasi ini terdiri dari 2 bagian yaitu bagian *receiver* (penerima) yang terhubung dengan komputer dan bagian *transmitter* (pengirim) yang terhubung dengan DT-SENSE Temperature Sensor. *Transmitter* akan mengirimkan data secara terus menerus ke *receiver* dengan interval waktu tertentu. *Receiver* akan mengolah data yang diterima kemudian dikirimkan ke komputer. Bagian *transmitter* terdiri dari DT-AVR Low Cost Micro System (DT-AVR LCMS) dan modul TX02. Bagian *receiver* terdiri dari DT-AVR LCMS dan modul RX01. Bahasa pemrograman yang digunakan adalah bahasa BASIC dengan IDE (*Integrated Development Environment*) BASCOM AVR versi Demo.

Modul & komponen yang digunakan pada aplikasi ini adalah:

- 2x DT-AVR LCMS.
- 1x EMS LCD Display.
- 1x modul RX01.
- 1x modul TX02.
- 1x resistor 10 K/0.25W.
- 1 set kabel penghubung.

Adapun blok diagram aplikasi “Monitoring Suhu Nirkabel” adalah sebagai berikut :



Gambar 1
Blok Diagram Aplikasi Monitoring Suhu Nirkabel

Hubungan antara modul-modul tersebut adalah sebagai berikut:

DT-AVR LCMS	DT-SENSE Temperature Sensor
VCC (J11 pin 2)	VCC (J1 pin 2)
GND (J11 pin 1)	GND (J1 pin 1)
PB.0 (J11 pin 3)*	SCL (J1 pin 6)
PB.1 (J11 pin 4)*	SDA (J1 pin 5)

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

Tabel 1
Hubungan DT-AVR LCMS dengan DT-SENSE Temperature Sensor secara I2C

DT-AVR LCMS	RX01 Module
VCC (J10 pin 2)	VDD
GND (J10 pin 1)	GND
PA.4 (J10 pin 7)	NSEL
PA.5 (J10 pin 8)	SDI
PA.6 (J10 pin 9)	SDO
PA.7 (J10 pin 10)	SCK
PD.2 (J13 pin 5)	NIRQ

Tabel 2
Hubungan DT-AVR LCMS dengan RX01 Module.

DT-AVR LCMS	TX02 Module
VCC (J10 pin 2)	VDD
GND (J10 pin 1)	GND
PA.1 (J10 pin 4)	FSK
PA.4 (J10 pin 7)	NSEL
PA.5 (J10 pin 8)	SDI
PA.6 (J10 pin 9)	NIRQ
PA.7 (J10 pin 10)	SCK

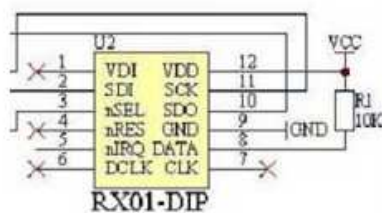
Tabel 3
Hubungan DT-AVR LCMS dengan TX02 Module.

DT-AVR LCMS	EMS LCD Display
VCC (J12 pin 2)	VCC (J3 pin 2)
GND (J12 pin 1)	GND (J3 pin 1)
PA.0 (J12 pin 3)	RS (J3 pin 3)
PA.1 (J12 pin 4)	R/W (J3 pin 4)
PA.2 (J12 pin 5)	EN (J3 pin 5)
PA.3 (J12 pin 6)	-
PA.4 (J12 pin 7)	DB4 (J3 pin 7)
PA.5 (J12 pin 8)	DB5 (J3 pin 8)
PA.6 (J12 pin 9)	DB6 (J3 pin 9)
PA.7 (J12 pin 10)	DB7 (J3 pin 10)

Tabel 4
Hubungan DT-AVR LCMS dengan EMS LCD Display

DT-SENSE Temperature Sensor terhubung dengan DT-AVR LCMS melalui antarmuka I2C. Data temperatur yang dibaca pada DT-SENSE Temperature Sensor adalah data temperatur dalam satuan derajat Celcius.

Pembacaan temperature dimungkinkan juga dalam satuan derajat Fahrenheit dengan melakukan perubahan program. Selanjutnya data pembacaan temperatur oleh DT-SENSE Temperature Sensor akan diolah oleh DT-AVR LCMS dan dipaket menjadi sebuah *frame data* (Gambar 4) sebelum dikirimkan melalui modul TX02. *Update* data pembacaan temperatur dan pengiriman data secara nirkabel dilakukan setiap 500 ms. Data yang dikirim secara nirkabel via modul TX02 akan diterima oleh modul RX01. Pada rangkaian modul RX01, perlu ditambahkan resistor sebesar 10K/0.25W dengan pemasangan sebagai berikut



Gambar 3
Letak pemasangan resistor 10K/0.25W

Pin NIRQ pada modul RX01 dimanfaatkan sebagai indikator untuk mengetahui ada tidaknya data yang masuk ke *buffer* penerima modul RX01. Selanjutnya *frame data* yang diterima akan dikirimkan ke komputer untuk diurai (*di-parsing*) sehingga mendapatkan beberapa informasi yang diperlukan. Proses penguraian data yang diterima ini dilakukan oleh program pada komputer yang ditulis menggunakan bahasa BASIC dengan IDE Visual BASIC 6.0. DT-AVR LCMS pada bagian *receiver* ini berfungsi untuk melakukan inialisasi awal pada modul RX01 sebelum dapat menerima data dan meneruskan *frame data* yang diterima oleh modul RX01 ke komputer untuk pemrosesan lebih lanjut. DT-AVR LCMS terhubung dengan komputer melalui antarmuka serial UART RS-232, sehingga *jumper* J4 & J5 harus diatur pada posisi 1-2 (UART RS-232). Jika komputer yang digunakan tidak memiliki *port serial*, maka dapat digunakan *USB to Serial Converter* sehingga DT-AVR LCMS dapat terhubung dengan komputer melalui *port USB*. Setelah semua modul dihubungkan sesuai dengan tabel koneksi antar modul, hubungkan DT-AVR LCMS dengan catu daya +12 VDC pada J2 DT-AVR LCMS dan isikan program **tx.hex** ke DT-AVR LCMS bagian pengirim (*transmitter*) dan program **rx.hex** ke DT-AVR LCMS bagian penerima (*receiver*) menggunakan DT-AVR ISP atau divais *programmer* lain yang kompatibel. Setelah proses pengisian program selesai, tekan tombol *reset* pada kedua modul untuk memastikan program dieksekusi mulai awal. *Frame data* yang dikirimkan oleh *Transmitter* memiliki susunan sebagai berikut:

STX [byte 1]	NID [byte 2]	TC [byte 3]	CS [byte 4]	ETX [byte 5]
'@'	'1'	[0x01 – 0x64]	[0x01 – 0xFF]	'\$'

Keterangan:

- STX, merupakan awal *frame data*. STX menggunakan karakter '@'.
- NID, merupakan nomor ID sensor. Pada aplikasi ini digunakan nomor ID sensor '1'.
- TC, merupakan data suhu dalam satuan derajat Celcius. *Range* nilainya antara 0x01 sampai 0x64 atau 0 sampai 100 derajat Celcius.
- CS, merupakan nilai Check Sum dari *frame data*. *Range* nilai Check Sum antara 0x01 sampai 0xFF. Nilai Check Sum merupakan nilai penjumlahan data *byte* mulai awal *frame data* sampai TC dengan mengambil 1 *byte* data saja.
- ETX, merupakan akhir *frame data*. ETX menggunakan karakter '\$'.

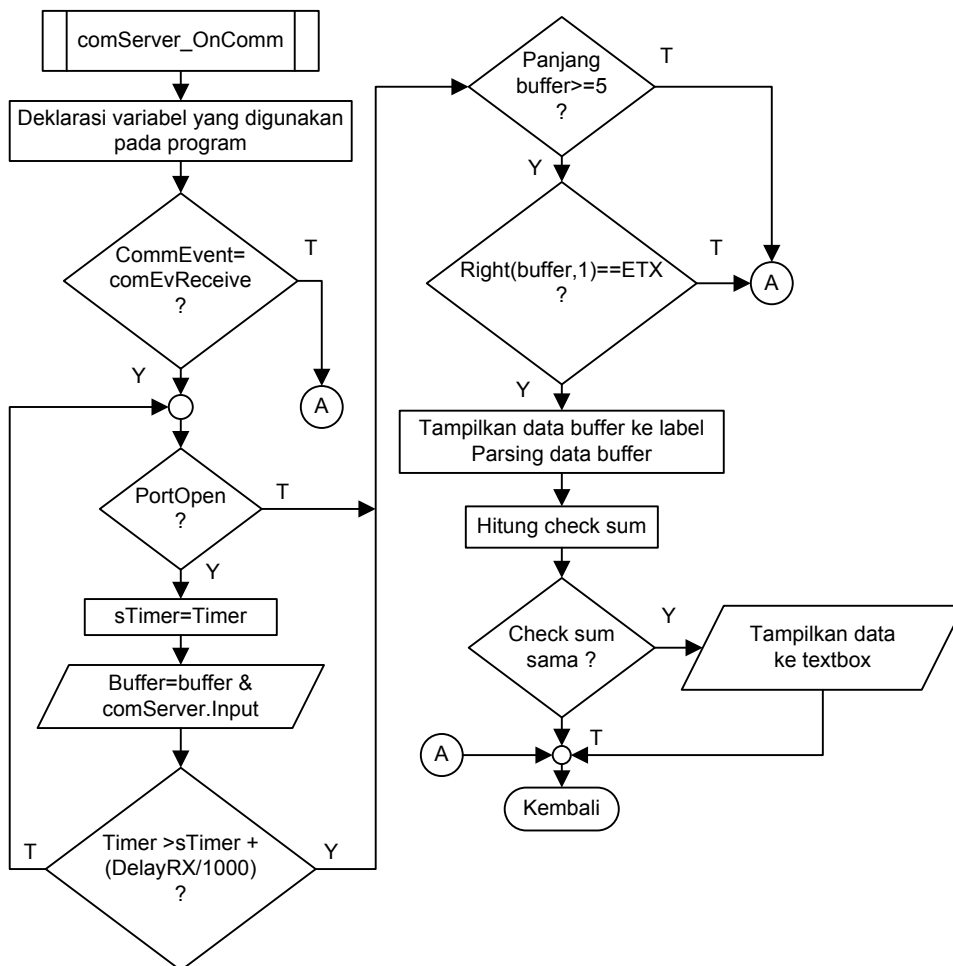
Program pada komputer berupa GUI (*Graphical User Interface*) yang akan menampilkan *frame data* yang dikirimkan oleh modul TX02 dan diterima oleh modul RX01. Program ini memanfaatkan komponen MSCOMM32.OCX pada Visual BASIC 6.0. Penerimaan data yang masuk ke komputer secara serial memanfaatkan *event* comEvReceive (*RS-232 Interrupt Service Routine*) pada MSCOMM. Pada pemrosesan data di komputer juga dilakukan pengecekan validitas data kembali. Prinsip proses pengecekannya sama dengan proses pengecekan pada mikrokontroler. Hanya data yang valid saja yang akan ditampilkan pada *textbox*. Data pada *textbox* terdiri dari beberapa informasi seperti:

- Informasi waktu saat data diterima oleh komputer (berdasarkan waktu jam pada BIOS).
- Informasi Node ID. Informasi ini akan bermanfaat jika DT-SENSE Temperature Sensor yang terpasang lebih dari 1.
- Informasi data suhu dalam satuan derajat celcius.
- Informasi nilai *check sum* dari *frame data* yang diterima.



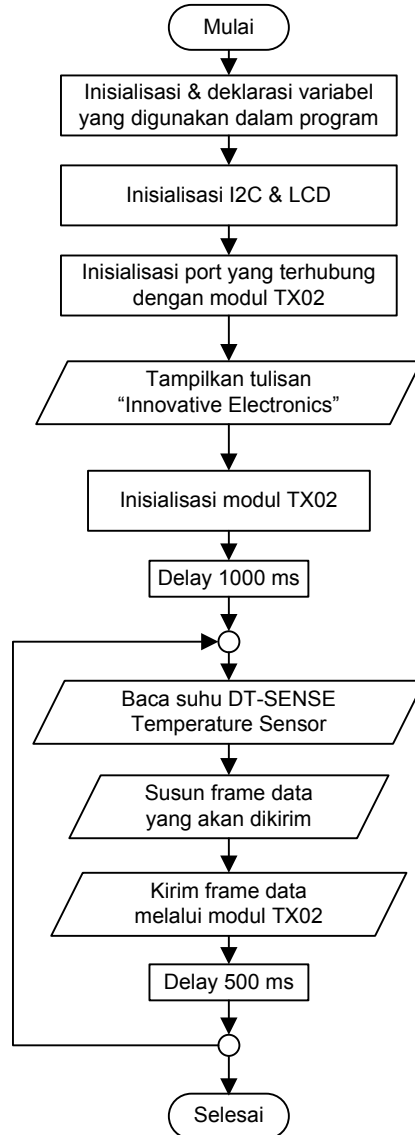
Gambar 4
Tampilan GUI pada komputer

Proses pengolahan data yang diterima saat terjadi event penerimaan data pada *port serial* komputer (*comEvReceive*) dapat diilustrasikan pada **flowchart** berikut ini:



Gambar 5
Flowchart event comServer_OnCom

Flowchart program tx.bas pada bagian transmitter secara garis besar adalah sebagai berikut:



Gambar 6
Flowchart program tx.bas pada bagian pengirim (transmitter)

Program tx.bas pada bagian pengiriman (*transmitter*) akan diproses sebagai berikut:

1. Pertama kali program akan mendeklarasikan dan melakukan inisialisasi dan deklarasi variabel-variabel yang digunakan pada program. Yang meliputi konfigurasi pin LCD, konfigurasi pin (SDA & SCL) untuk jalur komunikasi I2C dan deklarasi pin (NSEL, SDI, NIRQ, SCK, FSK) untuk jalur komunikasi dengan modul TX02.
2. Selanjutnya program melakukan inisialisasi I2C dan LCD dengan menambahkan kode program seperti gambar berikut ini:

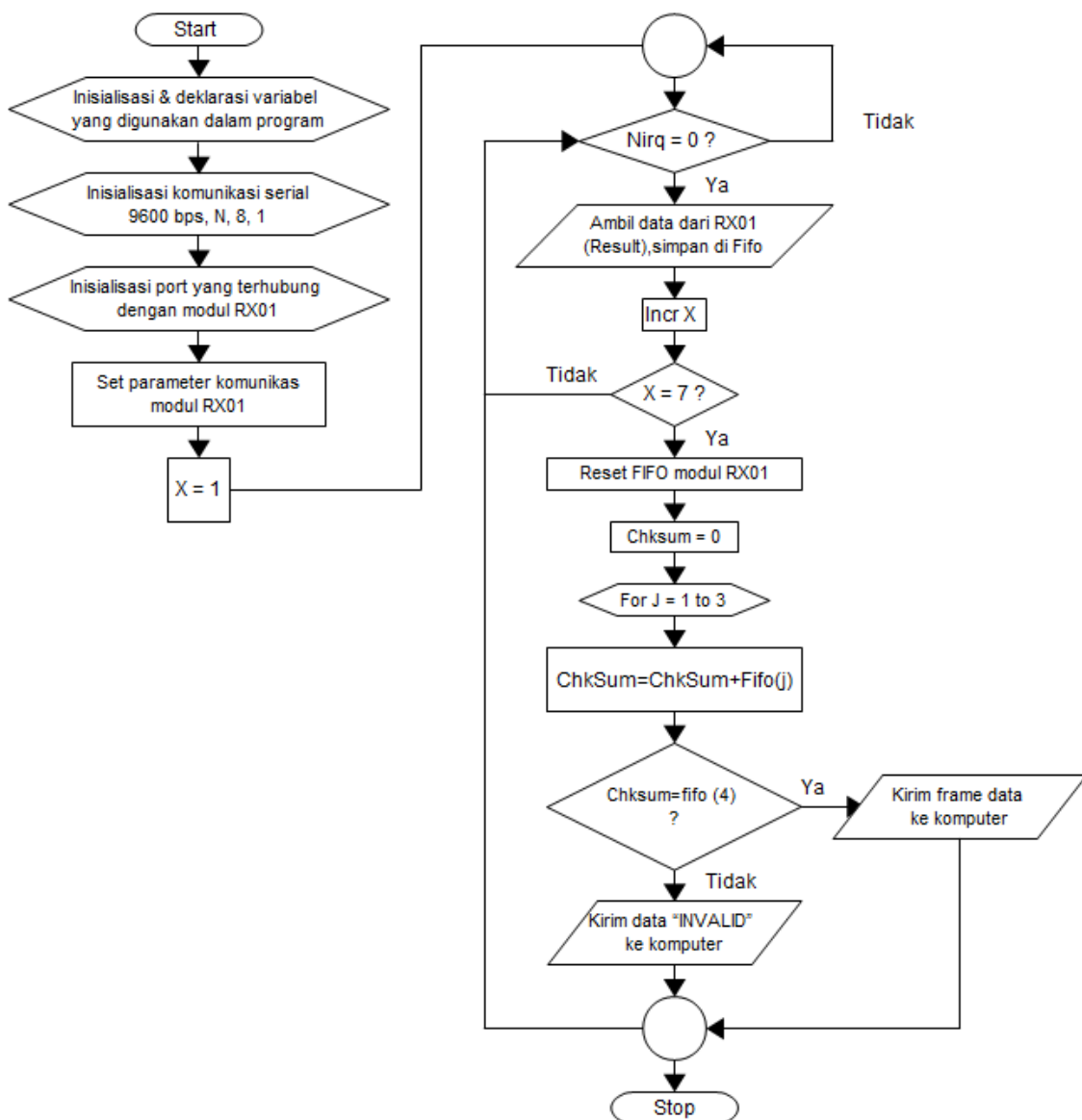
```
I2cinit  
Initlcd
```

Gambar 7
Kode program untuk inisialisasi I2C & LCD

3. Program menampilkan tulisan "INNOVATIVE ELECTRONICS" pada layar LCD EMS LCD Display.
4. Program melakukan *delay* selama 1 detik.

5. Program selanjutnya akan membaca data temperatur (dalam satuan derajat Celcius) dari modul DT-SENSE Temperature Sensor. Program diawali *start condition* kemudian diikuti dengan mengirimkan alamat tulis E0H dan perintah 00H yang digunakan untuk membaca data suhu dalam satuan derajat Celcius. Selanjutnya program mengirimkan *stop condition* dan program melakukan *delay* selama 20 ms. Selanjutnya program mengirimkan kembali *start condition* lagi dan alamat baca E1H untuk membaca respon (ACK) dari pengiriman perintah 00H (baca data suhu dalam derajat Celcius). Selanjutnya baca data yang diterima dan akhiri dengan *stop condition*.
6. Hasil pembacaan data suhu selanjutnya akan diproses oleh program dan dipaket menjadi sebuah *frame data* untuk dikirimkan melalui modul TX02. Selain itu data suhu yang sudah diolah juga ditampilkan pada layar LCD EMS LCD Display.
7. *Frame data* yang telah selesai dipaket selanjutnya dikirimkan oleh modul TX02 ke modul RX01. Kemudian program melakukan *delay* selama 500 ms.
8. Program kembali melakukan pembacaan data suhu.

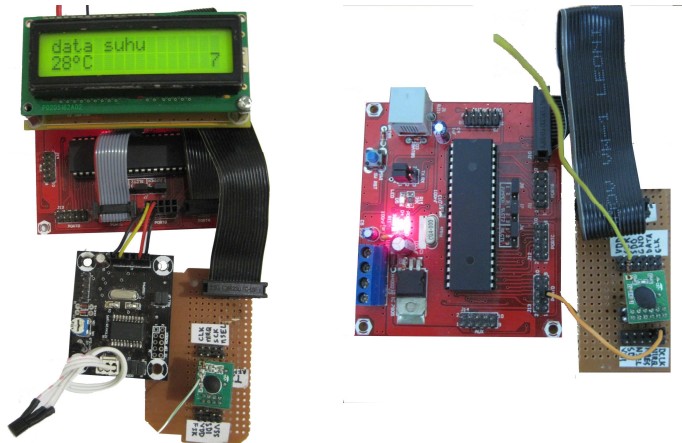
Flowchart program rx.bas pada bagian *Receiver* secara garis besar adalah sebagai berikut:



Gambar 8
Flowchart program rx.bas pada bagian penerima (receiver)

Program rx.bas pada bagian penerima (*receiver*) akan diproses sebagai berikut:

1. Pertama kali program akan melakukan deklarasi dan inialisasi variabel-variabel yang digunakan dalam program.
2. Program melakukan inialisasi komunikasi serial yang nantinya digunakan untuk komunikasi antara komputer dengan DT-AVR LCMS. Konfigurasi komunikasi serial yang digunakan adalah 9600 bps, N, 8, 1. Komputer terhubung dengan DT-AVR LCMS melalui komunikasi serial UART RS-232.
3. Program melakukan inialisasi *port I/O* mikrokontroler ATmega8535 pada DT-AVR LCMS yang terhubung dengan modul RX01. Pin-pin modul RX01 yang terhubung dengan modul RX01 adalah VDD, GND, FSK, NSEL, SDI, SCK, dan NIRQ. Pin-pin modul RX01 tersebut terhubung dengan PORT A ATmega8535 pada DT-AVR LCMS.
4. Kemudian program melakukan *setting* komunikasi modul RX01. *Setting* konfigurasi yang dilakukan meliputi:
 - Pengaturan frekuensi kerja pada 915 MHz dan *bandwidth* komunikasi 134 kHz.
 - Pengaturan *data rate* sebesar 4800 bps.
 - Aktifasi fungsi FIFO dan *receiver mode* untuk penerimaan data.
5. Program melakukan inialisasi variabel $x=0$, yang nantinya digunakan untuk *pooling* penerimaan data.
6. Program selanjutnya melakukan pengecekan PIND.2 atau Nirq. Jika bernilai 0 berarti pin NIRQ aktif. Hal ini menandakan bahwa ada data yang masuk ke *buffer* FIFO modul RX01.
7. Program mengambil data yang masuk ke *buffer* FIFO dan selanjutnya disimpan ke variabel *array* Fifo dengan *indeks* sesuai nilai x . Selanjutnya nilai x di-*increment*.
8. Program melakukan pengecekan nilai x . Jika nilai $x=7$ maka program akan me-*reset* *buffer* FIFO agar nantinya siap untuk menerima data yang baru. Selanjutnya dilakukan pemrosesan data yang diterima. Program melakukan inialisasi variabel $ChkSum=0$ dan melakukan perhitungan *check sum* untuk mengetahui validitas data. Nilai $ChkSum$ merupakan hasil penjumlahan seluruh data yang disimpan di variabel Fifo (1-3) dengan mengambil 8 bit data LSB.
9. Selanjutnya program akan membandingkan hasil perhitungan *check sum* yang tersimpan di variabel $ChkSum$ dengan nilai data Fifo(4). Jika nilainya sama maka data valid, jika tidak sama maka data tidak valid.
10. Jika data valid selanjutnya data tersebut akan dikirimkan ke komputer. Jika data tidak valid maka program akan mengirimkan data "Invalid" ke komputer. Variabel x kemudian diset menjadi 0.
11. Selanjutnya program akan melakukan pengecekan kembali nilai PIND.2 untuk memulai siklus pengecekan data kembali.



Gambar 9
Tampilan rangkaian pada AN189

Listing program aplikasi ini terdapat pada **AN189.ZIP**

Selamat berinovasi!

*All trademarks, trade names, company names, and product names are the property of their respective owners.
All softwares are copyright by their respective software publishers and/or creators.*