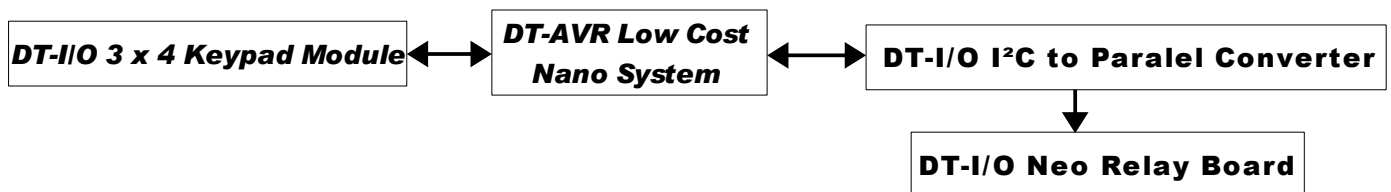


Mengembangkan aplikasi yang cukup besar dan kompleks tentu membutuhkan pin untuk I/O yang banyak. Kendala yang sering kali dihadapi adalah permasalahan *port* I/O yang disediakan di mikrokontroler/mikroprosesor kurang untuk memenuhi kebutuhan aplikasi tersebut. Solusi untuk permasalahan tersebut adalah dengan cara melakukan ekspansi dari *port* yang tersedia. Terdapat beberapa teknik yang dapat digunakan untuk pengolahan ekspansi *port* tersebut antara lain dengan menggunakan bantuan IC *multiplexer*. Selain menggunakan teknik tersebut, dapat juga menggunakan DT-I/O I²C to Parallel Converter. Modul ini berfungsi sebagai modul ekspansi 2 pin I/O menjadi 16 pin I/O. Pada aplikasi kali ini akan dibahas mengenai cara penggunaan DT-I/O I²C to Parallel Converter dan contoh aplikasi yang dapat dikembangkan dengan tambahan beberapa modul lain menggunakan bantuan kompiler CodeVisionAVR yang berbasis *C-Language*.

Adapun modul yang diperlukan dalam aplikasi kali ini :

- DT-AVR Low Cost Nano System/DT-AVR Low Cost Micro System.
- DT-I/O I²C to Parallel Converter.
- DT-I/O Neo Relay Board.
- DT-I/O 4 x 4 Keypad Module.

Blok diagram dari sistem secara keseluruhan adalah sebagai berikut :



Gambar 1
Blok Diagram AN151

Hubungan antara modul-modul tersebut adalah sebagai berikut :

| DT-AVR Low Cost Nano System | DT-I/O I²C to Paralel Converter |
|------------------------------------|---|
| GND | GND |
| VCC | +5 VDC |
| * PORTD.2 (J8.5) | SDA (J1) |
| * PORTD.3 (J8.6) | SCL (J1) |

* Pin ini tidak mutlak dan dapat diganti pin lain tetapi harus mengubah program

Tabel 1
Hubungan DT-AVR Low Cost Nano System dengan DT-I/O I²C to Paralel Converter

| DT-I/O I²C to Paralel Converter | DT-I/O Neo Relay Board |
|---|-------------------------------|
| Pin 0 (J5.3) | Relay 1 (J1.3) |
| Pin 1 (J5.4) | Relay 2 (J1.4) |
| Pin 2 (J5.5) | Relay 3 (J1.5) |
| Pin 3 (J5.6) | Relay 4 (J1.6) |
| Pin 4 (J5.7) | Relay 5 (J1.7) |
| Pin 5 (J5.8) | Relay 6 (J1.8) |
| Pin 6 (J5.9) | Relay 7 (J1.9) |
| Pin 7 (J5.10) | Relay 8 (J1.10) |

Tabel 2
Hubungan DT-I/O I²C to Paralel Converter dengan DT-I/O Neo Relay Board

| DT-AVR Low Cost Nano System | DT-I/O 4x4 Keypad Module |
|------------------------------------|---------------------------------|
| VCC (J7.2) | VCC |
| PORTB.0 (J7.3) | C1 |
| PORTB.1 (J7.4) | C2 |
| PORTB.2 (J7.5) | C3 |
| PORTB.3 (J7.6) | C4 |
| PORTB.4 (J7.7) | R1 |
| PORTB.5 (J7.8) | R2 |
| PORTB.6 (J7.9) | R3 |
| PORTB.7 (J7.10) | R4 |

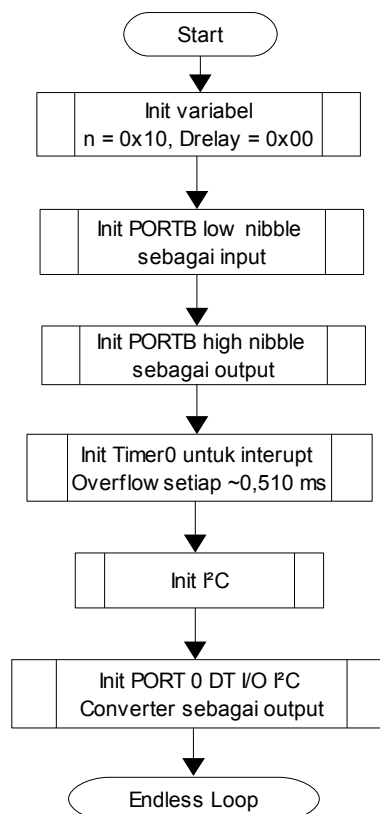
Tabel 3
Hubungan DT-AVR Low Cost Nano System dengan DT-I/O 4x4 Keypad Module

Keterangan :

- ◆ Resistor *pull-up* untuk SDA dan SCL di DT-I/O I²C to Parallel Converter harus dipasang, caranya dengan memasang *jumper* pin 1-2 di J4 dan *jumper* pin 3-4 di J4.
- ◆ Pada DT-I/O Neo Relay Board, atur *jumper* J4 pada posisi 1-2.
- ◆ DT-I/O I²C to Paralel Converter memerlukan catu daya +5 Volt DC.
- ◆ DT-I/O Neo Relay Board memerlukan catu daya +5 Volt untuk rangkaian relay dan +12 Volt untuk relay, pastikan tidak salah antara catudaya +5 Volt dengan +12 Volt.
- ◆ Hubungan GND antara semua modul harus saling terhubung.

Setelah semua pengaturan selesai, isikan kode program **k.Hex** ke dalam DT-AVR Low Cost Nano System dengan menggunakan bantuan DT-HiQ AVR In System Programmer / DT-HiQ AVR USB ISP ataupun alat programmer yang lain yang memiliki fitur In-System Programming. Secara singkat, program **k.hex** akan membaca input dari penekanan *keypad*. Jika tombol 1 ditekan, maka program akan mengirimkan perintah untuk menghidupkan relay ke-1. Demikian juga untuk tombol 2 – 8 akan menghidupkan relay ke 2 – 8 dan tombol 9 akan mematikan semua relay.

Flowchart dari main program **k.Hex** sebagai berikut :

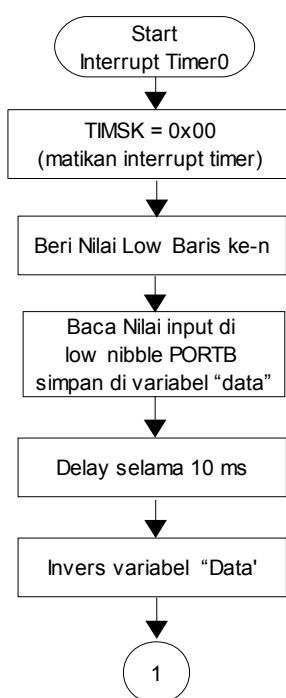


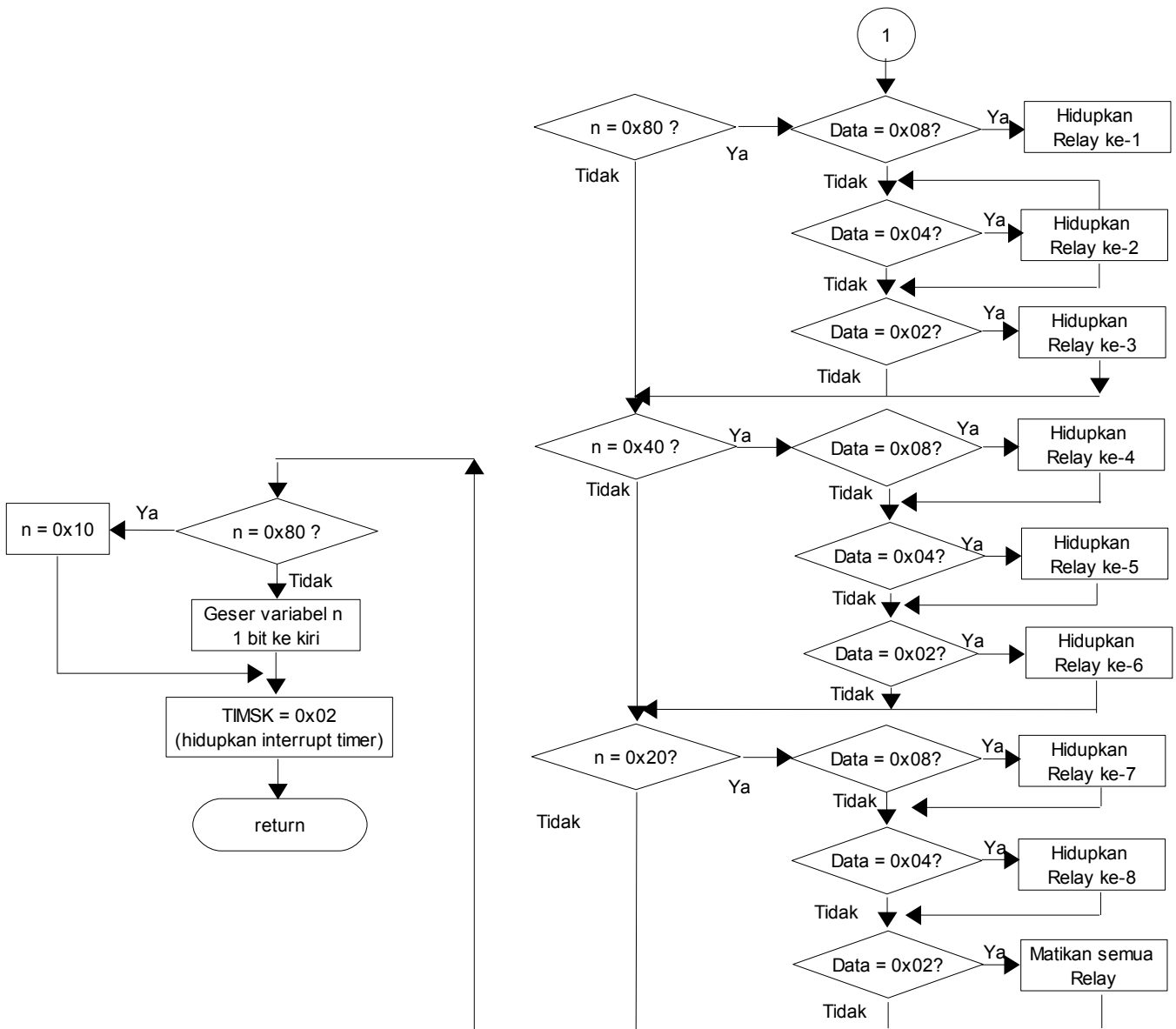
Gambar 2. Flowchart Main Program

Penjelasan program sebagai berikut :

1. Pertama kali program akan melakukan inisialisasi variabel, *port* dan timer DT-AVR Low Cost Nano System, serta inisialisasi komunikasi I²C dengan DT-I/O I²C to Parallel Converter.
2. Variabel "n" digunakan untuk nilai *scanning* baris DT-I/O 4 x 4 Keypad Module. Nilai awal = 0x10.
3. Variabel "Drelay" digunakan untuk menampung data keadaan relay. Nilai awal Drelay = 0x00 (semua relay aktif pada awal program).
4. PORTB *high nibble* adalah PORTB.4 s.d PORTB.7, sedangkan PORTB *low nibble* adalah PORTB.0 s.d PORTB.3. PORTB ini digunakan untuk keperluan *scanning keypad*.
5. *Interrupt* Timer0 digunakan untuk men-*scan keypad* setiap sekitar 0,51 ms. Pengaturan nilai 0,51 ms ini ada di register TCCR0B (dapat dibaca di *datasheet* ATtiny2313). Sedangkan pengaktifan *interrupt overflow* timer0 ada di register TIMSK. Pengaturan untuk program ini:
TCCR0B = 0x02
TIMSK = 0x02
6. Inisialisasi I²C digunakan untuk mengatur port 0 di DT-I/O I²C to Parallel Converter sebagai output.
7. Program utama di aplikasi ini seluruhnya ada di *interrupt overflow timer* yang terjadi setiap 0,51 ms.

Flowchart dari subrutin Interrup Timer0 dari program **k.Hex** sebagai berikut :





Gambar 3

Flowchart Subrutin Interrupt Timer0

Penjelasan program sebagai berikut :

1. Setiap kali *interrupt* terjadi, program akan memberi nilai *LOW* baris ke-n dari DT-IO 4x4 Keypad Module. Setelah itu program akan membaca nilai kolom 1 s.d 4 dengan perintah :
`data=((PINB.3<<3) | (PINB.2<<2) | (PINB.1<<1) | (PINB.0<<0))`
 Data kolom ini ada di *low nibble* variabel kemudian data di inversi, karena data kolom yang dibaca sebelumnya adalah aktif *low* (tujuan inversi sebenarnya hanya untuk memudahkan pengolahan data selanjutnya).
2. Setelah data kolom diperoleh, langkah selanjutnya adalah mencocokkannya dengan baris yang sekarang di *-scan*. Kombinasi data baris yang sedang di *-scan* dengan data kolom yang dibaca inilah yang digunakan untuk menentukan relay mana yang akan diaktifkan/dimatikan.

3. Variabel *n* digeser untuk scanning baris selanjutnya.

```
if (n==0x80)  n=0x10;  else n<<=1;
```

Di aplikasi kali ini hanya menggunakan 9 tombol dari 16 tombol yang tersedia pada di DT-IO 4x4 Keypad Module, sedangkan di DT-IO I²C to Paralel Converter hanya menggunakan 8 port I/O dari 16 port I/O yang ada. Sehingga terdapat banyak *port* yang tersisa untuk aplikasi yang lebih kompleks.

Listing program terdapat pada file **AN151.ZIP**.

Selamat berinovasi!

CodeVisionAVR is copyright by Pavel Haiduc, HP InfoTech s.r.l.
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.