

DT-51

DT-51 *Application Note*

AN136 – Type With SPC Keymatic

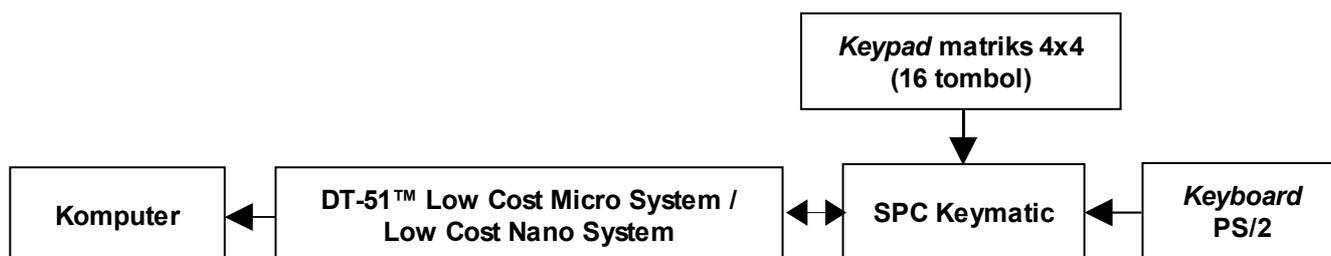
Oleh: Tim IE

Aplikasi berikut memberikan contoh sederhana mengenai penggunaan modul DT-51™ dengan bahasa pemrograman C ($\mu\text{C}/51^{\circ}$, Wickenh euser). DT-51™ Low Cost Micro System akan membaca data *keyboard* dan *keypad* yang sudah diterjemahkan oleh SPC Keymatic dan mengirimkan data tersebut ke komputer untuk ditampilkan melalui program Terminal[ ]. Kedua modul tersebut dihubungkan dengan antarmuka paralel maupun antarmuka I²C, kedua antarmuka ini didukung penuh oleh modul SPC Keymatic. AN Ini merupakan suatu contoh sederhana penggunaan DT-51™ Low Cost Micro System untuk membaca penekanan tombol *keypad* maupun *keyboard* melalui SPC Keymatic dengan pemrograman berbahasa C.

Komponen yang diperlukan:

- 1 buah DT-51™ Low Cost Micro System / Low Cost Nano System,
- 1 buah SPC Keymatic,
- 1 buah *Keypad* matriks 4x4 (16 tombol),
- 1 buah *Keyboard* PS/2.

Adapun blok diagram sistem secara keseluruhan adalah sebagai berikut:



Gambar 1
Blok Diagram AN136

Hubungan antara modul-modul tersebut adalah sebagai berikut:

DT-51™ Low Cost Micro System / Low Cost Nano System	SPC Keymatic
GND (J10/J5)	GND (J5)
VOOUT (J10/J5)	+5V (J5)
P1.0* (pin 3 - J4/J6)	Data bit 0 (pin 3 - J4)
P1.1* (pin 4 - J4/J6)	Data bit 1 (pin 4 - J4)
P1.2* (pin 5 - J4/J6)	Data bit 2 (pin 5 - J4)
P1.3* (pin 6 - J4/J6)	Data bit 3 (pin 6 - J4)
P1.4* (pin 7 - J4/J6)	RD (pin 1 - J4)
P3.2* (pin 5 - J6/J7)	INT (pin 2 - J4)

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

Tabel 1

Hubungan DT-51™ Low Cost Micro System / Low Cost Nano System dengan SPC Keymatic Secara Paralel

DT-51™ Low Cost Micro System / Low Cost Nano System	SPC Keymatic
GND (J10/J5)	GND (J5)
VOOUT (J10/J5)	+5VDC (J5)
P3.4* (pin 7 - J6/J7)	SDA (J1)
P3.5* (pin 8 - J6/J7)	SCL (J1)
P3.2* (pin 5 - J6/J7)	INT (J1)

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

Tabel 2
Hubungan DT-51™ Low Cost Micro System / Low Cost Nano System Dengan SPC Keymatic Secara I²C

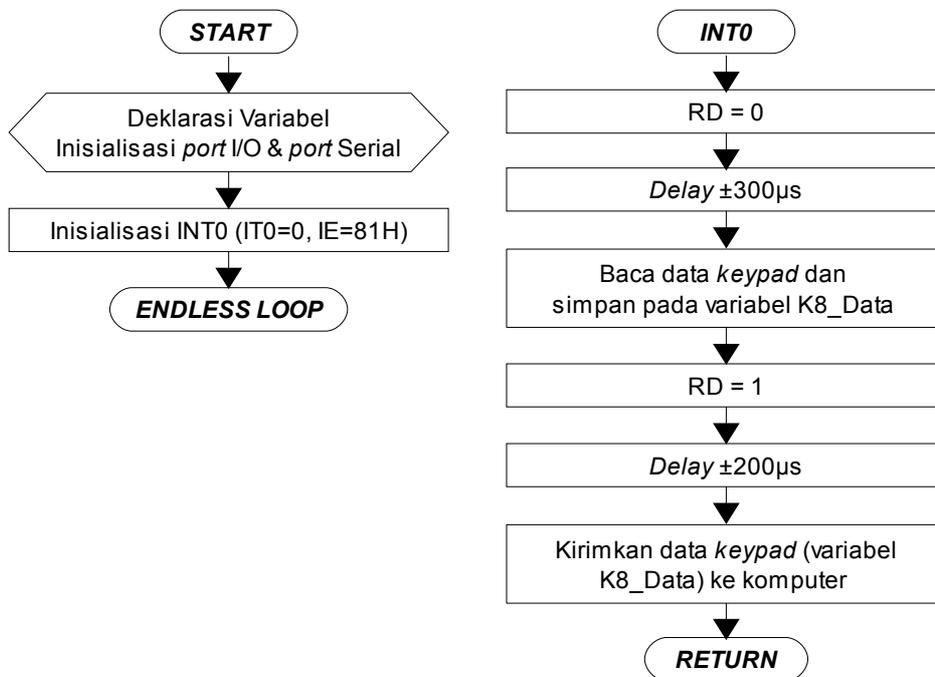
Hubungkan konektor PS/2 (*male*) keyboard ke konektor J8 (PS/2 *female*) pada SPC Keymatic, serta hubungkan juga *keypad* matriks 4x4 pada konektor J7. Keterangan lebih lanjut tentang susunan tombol *keypad* dan cara menghubungkannya ke SPC Keymatic dapat dibaca pada buku Manual SPC Keymatic. Pasang semua *jumper* pada pin "Addr." A0-A1 (J3) serta semua *jumper* pada pin SCL/SDA (J2). Alamat I²C untuk modul SPC Keymatic pada contoh program aplikasi ini adalah E8H.

Gunakan kabel serial DT-51™ Low Cost *series* (RJ11 ke DB9) untuk menghubungkan modul DT-51™ Low Cost *series* ke komputer. Kabel ini digunakan sebagai jalur komunikasi data serial antara DT-51™ Low Cost *series* dengan komputer. Atur konfigurasi serial UART pada *baud rate* 9600 bps, 8 bit data, tanpa bit *parity*, 1 bit stop, dan tanpa *flow control*.

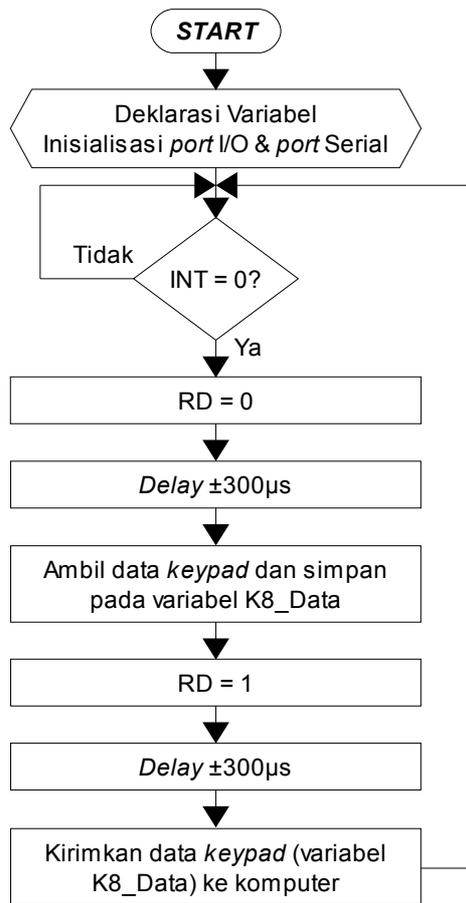
Setelah semua rangkaian dan sumber tegangan terhubung dengan benar, programlah PROGRAM.HEX pada direktori "Program_PARALEL" (untuk komunikasi secara paralel) atau pada direktori "Program_I2C" (untuk komunikasi secara I²C) ke DT-51™ Low Cost Micro System dengan bantuan DT-HiQ AT89S In System Programmer. Jika menggunakan DT-51™ Low Cost Nano System maka pemrogramannya dapat menggunakan DT-51™ ProgPAL + DT-51™ MinSys.

Catatan: untuk antarmuka paralel, SPC Keymatic hanya bisa membaca *input* dari tombol *keypad* dan tidak bisa membaca *input* dari tombol *keyboard*.

Flowchart program untuk komunikasi secara PARALEL adalah sebagai berikut:



Gambar 2
Flowchart Program Untuk Pengambilan Data Keypad Dengan Metode Interrupt



Gambar 3
Flowchart Program Untuk Pengambilan Data Keypad Dengan Metode Pooling

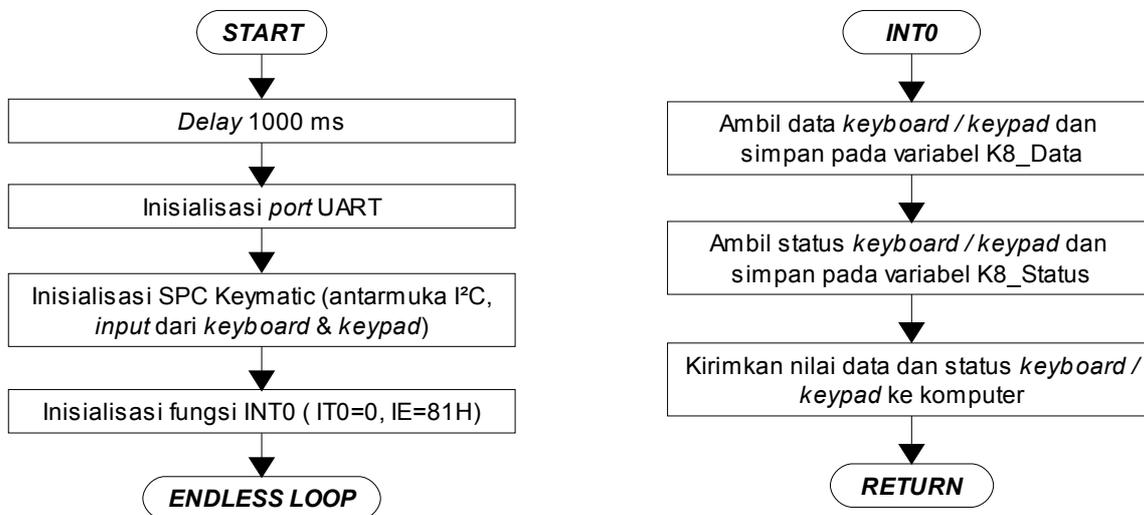
Program utama (komunikasi secara PARALEL) akan diproses sebagai berikut:

1. Proses yang pertama kali dilakukan adalah deklarasi variabel, inisialisasi pin RD sebagai *output* dengan nilai logika awal '1', dan inisialisasi *port* komunikasi serial UART pada *baud rate* 9600 bps, 8 bit data, tanpa bit *parity*, 1 bit stop, dan tanpa *flow control*.
2. Setelah proses inisialisasi selesai, berikutnya program akan memilih sumber *interrupt eksternal* 0 dan mengatur agar interupsi aktif pada saat pin INT0 (P3.2) mendapat logika '0'.
3. Kemudian program akan mengaktifkan sumber *interrupt eksternal* 0 sekaligus *global interrupt*. Setelah itu program akan mengeksekusi proses perulangan tanpa akhir.
4. Jika terjadi *interrupt eksternal* 0 (pin INT0 berlogika '0') maka program akan mengerjakan subrutin yang terdapat pada *Interrupt Service Routine (ISR) External* 0. Proses dalam *ISR External* 0 tersebut yaitu:
 - a) Pin RD akan diberi logika '0' sebagai sinyal untuk membaca data dari SPC Keymatic.
 - b) Kemudian *delay* selama kurang lebih 300 mikro detik.
 - c) Setelah itu, program akan membaca data dari SPC Keymatic dan menyimpan hasil pembacaan ke variabel K8_Data.
 - d) Lalu program memberi logika '1' pada pin RD dan berhenti lagi selama kurang lebih 200 mikro detik dengan asumsi logika pada pin INT telah kembali ke logika '1'. Kondisi logika '1' pada INT menunjukkan bahwa *buffer* data pada SPC Keymatic telah dibaca.
 - e) Proses berikutnya adalah mengirimkan kata "Tombol keypad = " dan variabel K8_Data dalam bentuk desimal dan dalam bentuk heksadesimal untuk ditampilkan pada program Terminal[®] di sisi komputer.
 - f) Setelah proses pengiriman data selesai maka program akan keluar dari *ISR External* 0 dan kembali ke baris program sebelum terjadi *interrupt*.

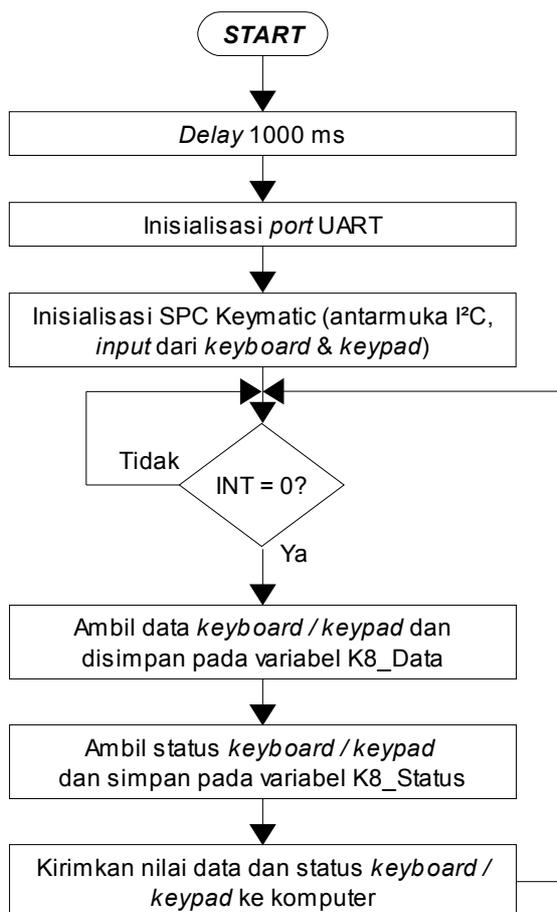
Pada program dengan metode *pooling*, proses nomor 4a hingga 4e terletak di dalam baris program utama. Program akan memeriksa kondisi logika pin INT0, ketika pin INT0 berlogika 0 maka program akan melakukan proses 4a - 4e.

Listing program untuk komunikasi secara PARALEL terdapat pada **AN136.ZIP** (direktori "Program_PARALEL").

Flowchart program untuk komunikasi secara I²C adalah sebagai berikut:



Gambar 4
Flowchart Program Untuk Pengambilan Data Keypad Dan Keyboard Dengan Metode Interrupt



Gambar 5
Flowchart Program Untuk Pengambilan Data Keypad dan Keyboard Dengan Metode Pooling

Program utama dengan komunikasi secara I²C akan diproses sebagai berikut:

1. Proses yang pertama kali dilakukan adalah menentukan definisi *pin* I/O SDA & SCL untuk komunikasi serial I²C, serta definisi *pin* *input* "INT" untuk menerima sinyal interupsi dari SPC Keymatic (akan berlogika '0' jika ada tombol yang ditekan). Lalu program melakukan deklarasi konstanta untuk alamat SPC Keymatic yaitu E8H, dan program akan berhenti selama 1000 mili detik.
2. Proses selanjutnya adalah inisialisasi komunikasi serial UART pada *baud rate* 9600 bps, 8 bit data, tanpa bit *parity*, 1 bit *stop* dan tanpa *flow control*.
3. Setelah itu program mengirimkan nilai K8_Mode sebesar 03H ke SPC Keymatic yang mempunyai arti *input* SPC Keymatic berasal dari *keypad* & *keyboard* dan antarmuka yang digunakan adalah I²C. Pengiriman nilai K8_Mode dilakukan dengan menggunakan prosedur K8_SetMode.
4. Setelah pengiriman nilai K8_Mode selesai, program akan mengatur mode *interrupt eksternal 0* sebagai *low level interrupt*. Kemudian mengaktifkan *global interrupt* dan *interrupt eksternal 0* secara bersamaan yaitu mengisi register IE dengan nilai 81H.
5. Setelah itu program akan mengeksekusi proses perulangan tanpa akhir. Jika terjadi interupsi pada *interrupt eksternal 0* (*pin* INT0 berlogika '0') maka program akan mengeksekusi *ISR External 0*, yaitu:
 - ✓ Program akan mengambil data dari SPC Keymatic dengan menggunakan prosedur K8_GetData dan hasil pembacaan disimpan pada variabel K8_Data.
 - ✓ Proses yang sama dilakukan untuk mengambil status *keyboard* dengan menggunakan prosedur K8_GetStatus dan nilainya disimpan pada variabel K8_Status.
 - ✓ Proses berikutnya adalah program membandingkan nilai K8_Data. Ada 3 kategori pengelompokan nilai K8_Data yang digunakan dalam contoh ini, masing-masing akan menampilkan hasil yang berbeda pada program Terminal[®] di sisi komputer. Tiga kategori tersebut yaitu:
 - i. Jika nilai K8_Data kurang dari atau sama dengan 0FH maka program akan menampilkan kata "Data keypad = " dan menampilkan variabel K8_Data dalam bentuk desimal dan heksadesimal.
 - ii. Jika nilai K8_Data antara 20H sampai 7EH maka program akan menampilkan kata "Data keyboard = " dan menampilkan variabel K8_Data dalam bentuk desimal, heksadesimal, dan ASCII (sehingga tampil karakter yang ditekan pada *keyboard*). Kemudian program juga menampilkan kata "Status = " dan status *keyboard* sesuai dengan nilai K8_Status, yaitu: "Shift", "Caps", "Num", "Scroll", atau "tidak ada status".
 - iii. Jika nilai K8_Data selain nilai di atas maka program akan menampilkan kata "Data keyboard = " dan menampilkan variabel K8_Data dalam bentuk desimal dan heksadesimal ke komputer. Kemudian program juga menampilkan kata "Status = " dan status *keyboard* sesuai dengan nilai K8_Status.

Setelah menampilkan nilai K8_Data dan K8_Status sesuai dengan kategori nilai K8_Data maka program akan kembali ke program utama.

Pada program dengan metode *pooling*, proses nomor 5 terletak di dalam baris program utama. Program akan memeriksa kondisi logika *pin* INT0, ketika *pin* INT0 berlogika 0 maka program akan melakukan proses 5.

Listing program untuk komunikasi secara I²C terdapat pada **AN136.ZIP** (pada direktori "Program_I2C").

Pada penjelasan program di atas (komunikasi secara paralel maupun I²C) digunakan fungsi *interrupt* sehingga pada baris program utama hanya terdapat proses perulangan tanpa akhir. Pada contoh program juga terdapat tambahan baris program untuk proses pemeriksaan kondisi *pin* INT0 dengan sistem *polling* (tidak menggunakan fungsi *interrupt*) yaitu dengan cara selalu memeriksa kondisi *pin* INT0. Jika menggunakan sistem *polling* maka *ISR External 0* tidak digunakan. Agar program dapat menggunakan sistem *polling* maka beberapa instruksi pada baris program utama harus ditutup (dijadikan baris komentar). Berikut ini merupakan contoh pengubahan program utama agar dapat menggunakan sistem *polling*.

◆ **Program utama untuk komunikasi secara paralel dan menggunakan metoda *polling*:**

```
void main (void)
{
    unsigned char i;
    RD=1;
    init232();          //Inisialisasi UART
```

```

//Memeriksa pin INT dari SPC KEYAMATIC dengan menggunakan Interrupt
/*IT0=0;           //Low level interrupt
IE=0x81;          //Enable global int enable external 0 int.
while(1);*/

//Memeriksa pin INT dari SPC KEYAMATIC dengan menggunakan sistem polling
while(1)
{
    while (INT==0)
    {
        RD=0;
        for (i=0; i<150; i++);
        K8_Data=(P1 & 0x0F);    //Ambil Low nibble
        RD=1;
        for (i=0; i<150; i++); //Time Last Buffer to INT high
        printf("Tombol keypad = %u = \"%X\"\n",K8_Data,K8_Data);
    }
}
}

```

◆ **Program utama untuk komunikasi secara I²C dan menggunakan metoda *pooling*:**

```

void main (void)
{
    _wait_ms(1000);
    init232();           //Inisialisasi UART

    K8_SetMode(ADDR_SPC,0x03); //Input Device (0x03) : Keypad & Keyboard
    (I2C)

    //Memeriksa pin INT dari SPC KEYAMATIC dengan menggunakan Interrupt
    /*IT0=0;           //Low level interrupt
    IE=0x81;          //Enable global int., enable external 0 int.
    while(1);*/

    //Memeriksa pin INT dari SPC KEYAMATIC dengan menggunakan sistem polling
    while(1)
    {
        while (INT==0)
        {
            K8_Data=K8_GetData(ADDR_SPC);           //Ambil data
            K8_Status=K8_GetStatus(ADDR_SPC);       //Ambil status
            print_tombol(K8_Data,K8_Status);
        }
    }
}

```

Selamat berinovasi!

Terminal is copyright by Bray++.

DT-51 is trademark of Innovative Electronics.

µC/51 is copyright by Wickenhäuser Elektrotechnik.

I²C is a registered trademark of Philips Semiconductors.

PS/2 is a trademark of International Business Machines Corporation.