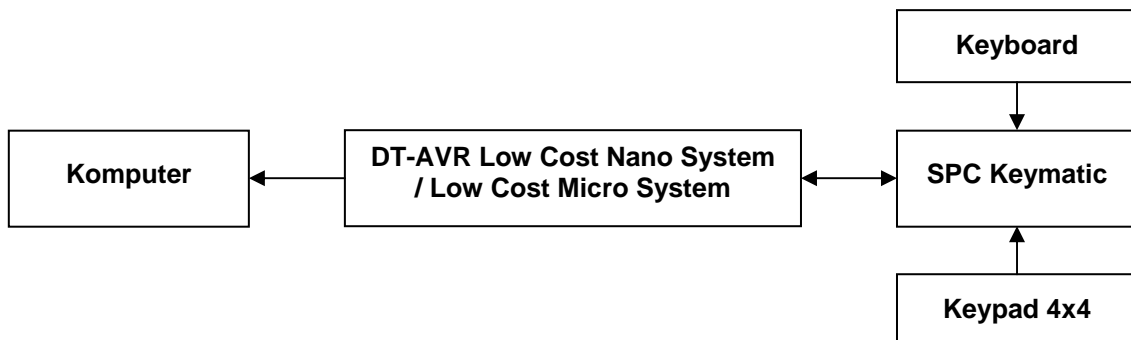


SPC Keymatic dapat menerima input tombol baik dari keyboard PS/2 maupun keypad matrix berukuran 3x4 atau 4x4. Pada aplikasi kali ini SPC Keymatic akan dihubungkan dengan modul DT-AVR Low Cost Series dan menggunakan bahasa pemrograman BASIC melalui *compiler* BASCOM-AVR©. Dalam artikel ini akan dibahas penggunaan SPC Keymatic dan DT-AVR Low Cost Series dengan komunikasi secara serial I²C maupun paralel. Aplikasi ini dapat menjadi contoh sederhana penggunaan DT-AVR Low Cost Series dalam menerima input multi tombol dari keyboard maupun keypad dengan bantuan SPC Keymatic. Serta dapat dikembangkan menjadi suatu divais input bagi komputer.

Komponen yang diperlukan:

- 1 DT-AVR Low Cost Nano System / Low Cost Micro System
- 1 SPC Keymatic
- 1 Keyboard PS/2 mini DIN 6-pin
- 1 Keypad 4x4

Adapun blok diagram sistem secara keseluruhan adalah sebagai berikut:



Gambar 1
Blok Diagram AN79

Hubungan antara modul-modul tersebut adalah sebagai berikut:

DT-AVR Low Cost Nano System/ Low Cost Micro System	SPC Keymatic
+5VDC	+5V (J5)
GND	GND (J5)
PB.6*	SCL (J6 pin 15)
PB.7*	SDA (J6 pin 16)
PB.2*	INT (J6 pin 11)

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

Tabel 1
Hubungan DT-AVR Low Cost Nano System / Low Cost Micro System dengan SPC Keymatic secara Serial I²C

Gunakan kabel serial DT-AVR Low Cost Nano System / Low Cost Micro System untuk menghubungkan modul dengan komputer. Sambungkanlah pula keyboard PS/2 mini DIN 6-pin dan keypad 4x4 ke SPC Keymatic sesuai dengan petunjuk dalam manual SPC Keymatic. Aturlah *jumper* J2 SPC Keymatic untuk memberi resistor *pull up* pada jalur komunikasi I²C (SCL dan SDA). Aturlah pula *jumper* J3 SPC Keymatic agar *board* SPC memiliki alamat terprogram 00b.

Setelah semua rangkaian dan sumber tegangan terhubung dengan benar, programlah I2C_KEYBOARD.BIN atau I2C_KEYBOARD.HEX ke dalam DT-AVR Low Cost Nano System (berbasis AT90S2313) dengan menggunakan DT-HiQ AVR In System Programmer atau divais ISP programmer lain yang memiliki konektor *header* 10-pin standar Atmel.

DT-AVR Low Cost Nano System/ Low Cost Micro System	SPC Keymatic
+5VDC	+5V (J5)
GND	GND (J5)
PB.2*	Data bit 3 (J4 pin 6)
PB.3*	Data bit 2 (J4 pin 5)
PB.4*	Data bit 1 (J4 pin 4)
PB.5*	Data bit 0 (J4 pin 3)
PB.6*	INT (J4 pin 2)
PB.7*	RD (J4 pin 1)

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

Tabel 2

Hubungan DT-AVR Low Cost Nano System / Low Cost Micro System dengan SPC Keymatic secara Paralel

Gunakan kabel serial DT-AVR Low Cost Nano System / Low Cost Micro System untuk menghubungkan modul dengan komputer. Pada sistem komunikasi secara paralel ini, SPC Keymatic hanya dapat menerima *input* dari keypad (dalam hal ini keypad 4x4).

Setelah semua rangkaian dan sumber tegangan terhubung dengan benar, programlah PARALLEL_KEYPAD.BIN atau PARALLEL_KEYPAD.HEX ke dalam DT-AVR Low Cost Nano System dengan menggunakan DT-HiQ AVR In System Programmer atau divais ISP programmer lain yang memiliki konektor *header* 10-pin standar Atmel.

Catatan:

Dalam aplikasi ini menggunakan DT-AVR Low Cost Nano System (dengan mikrokontroler AT90S2313) sehingga I2C_KEYBOARD.BIN / I2C_KEYBOARD.HEX maupun PARALLEL_KEYPAD.BIN / PARALLEL_KEYPAD.HEX yang disertakan dalam **AN79.ZIP** hanya berlaku untuk AT90S2313.

Apabila menggunakan mikrokontroler tipe lain, bukalah I2C_KEYBOARD.BAS atau PARALLEL_KEYPAD.BAS menggunakan BASCOM-AVR© dan ubahlah baris pertama dalam program tersebut:

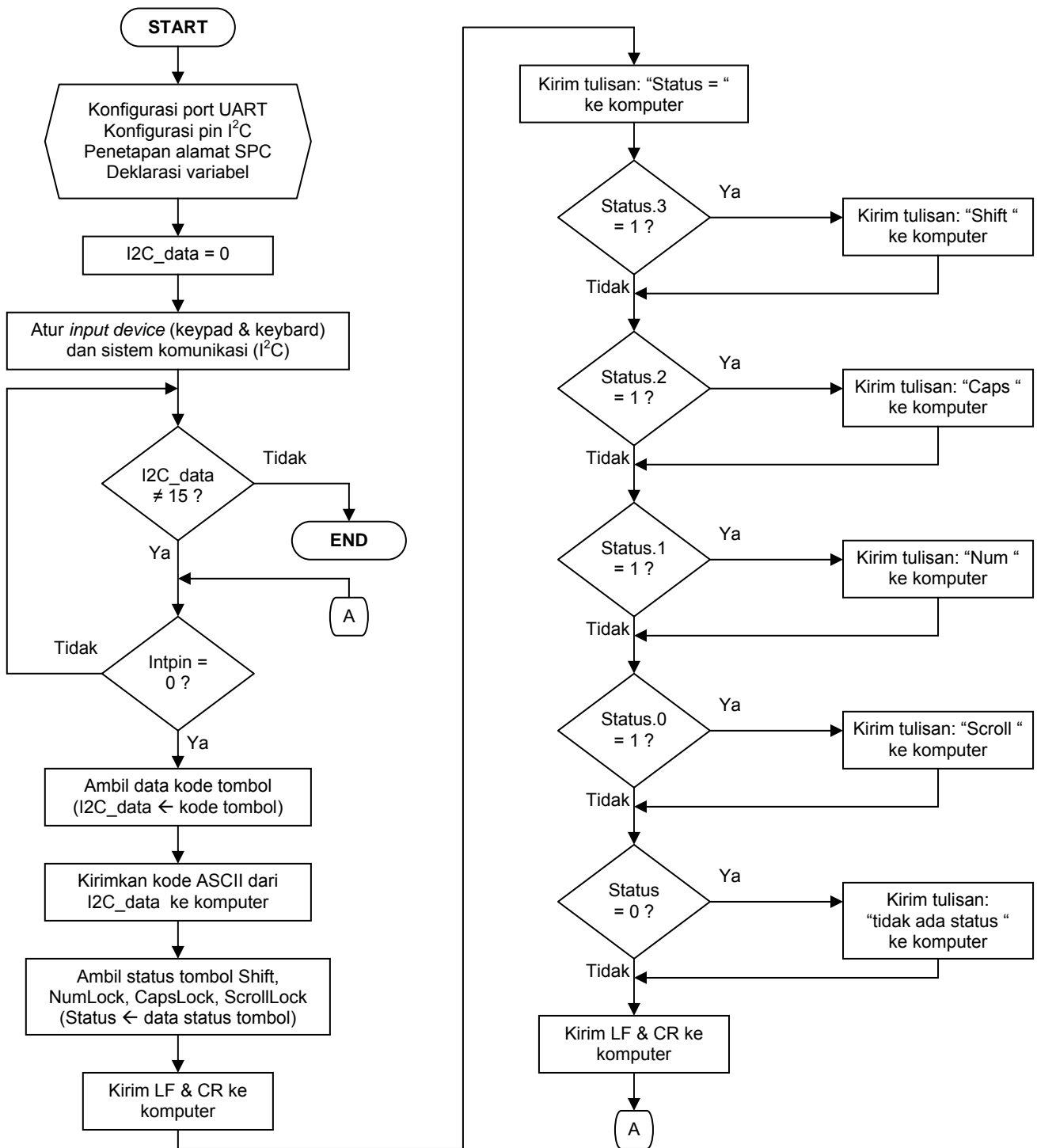
\$regfile = "2313def.dat" '→ untuk AT90S2313

\$regfile = "8535def.dat" '→ untuk AT90S8535

\$regfile = "m8535.dat" '→ untuk ATmega8535 (DT-AVR Low Cost Micro System)

agar sesuai dengan tipe mikrokontroler yang akan digunakan. Lalu *compile* ulang program sehingga menghasilkan *file* dengan ekstensi .bin atau .hex yang sesuai dan dapat diprogram ke dalam modul mikrokontroler.

Flowchart program untuk komunikasi secara Serial I²C adalah sebagai berikut:



Gambar 2
Flowchart program untuk komunikasi secara Serial I²C

Program utama untuk komunikasi secara Serial I²C akan diproses sebagai berikut:

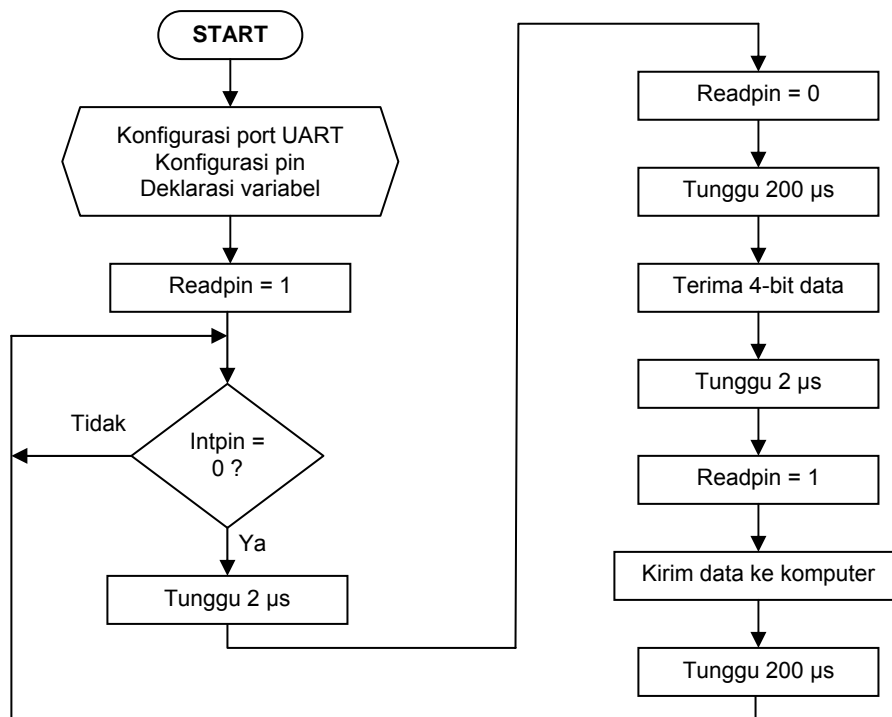
1. Proses yang pertama dilakukan adalah menentukan definisi pin I/O (SDA dan SCL) untuk komunikasi serial I²C dan konfigurasi port UART (*baudrate* 9600 bps, 8-bit data, 1 *stop bit*, tanpa *parity bit*).
2. Penetapan alamat tulis SPC Keymatic yaitu E8h (alamat terprogram 00b).
3. Kemudian deklarasi variabel I2C_data, Readaddr, Status, Key_mode.
I2C_data adalah variabel I/O yang digunakan dalam prosedur *low level* (misalnya: I2c_wbyte).
Readaddr adalah variabel berisi alamat baca SPC Keymatic untuk proses baca.

Status adalah variabel yang menunjukkan status tombol Shift, CapsLock, ScrollLock, dan NumLock.

Key_mode adalah variabel yang berisi data untuk menentukan jenis komunikasi dan *input device* yang dipakai.

4. Pada tahap berikutnya program akan menjalankan rutin Key_setmode_i2c dengan Key_mode = 03h (sistem komunikasi serial I²C, keyboard & keypad sebagai *input device*).
5. Setelah itu program akan melihat apakah I2c_data yang didapat dari input device adalah 15h. Apabila benar maka Program akan segera berakhir tetapi apabila sebaliknya maka program berjalan terus.
6. Kemudian program akan melihat keadaan pin Int, bila berlogika low (terjadi penekanan tombol) maka program melanjutkan ke langkah 7 tetapi bila berlogika high maka program akan kembali ke langkah 5.
7. Setelah itu program akan mengirimkan perintah Key_getdata_i2c untuk mengetahui tombol apa yang ditekan.
8. Segera setelah data diperoleh, program mengirimkan data dalam format ASCII itu ke komputer melalui komunikasi serial. Dan data ini dapat ditampilkan pada layar komputer menggunakan bantuan Terminal© atau HyperTerminal© (dengan *baudrate* 9600 bps, 8-bit data, 1 *stop bit*, tanpa *parity bit*, tanpa *flow control*).
9. Pada tahap berikutnya, program akan mengirimkan perintah Key_getstatus_i2c untuk melihat status tombol Shift, CapsLock, ScrollLock, dan NumLock.
10. Setelah itu program akan mengirimkan status dari masing-masing tombol tersebut ke komputer melalui komunikasi serial. Dan dapat ditampilkan pada layar komputer menggunakan bantuan Terminal© atau HyperTerminal©
11. Kembali ke langkah 6.

Flowchart program untuk komunikasi secara Paralel adalah sebagai berikut:



Gambar 3
Flowchart program untuk komunikasi secara Paralel

Program utama untuk komunikasi secara Paralel akan diproses sebagai berikut:

1. Proses yang pertama dilakukan adalah konfigurasi port UART (*baudrate* 9600 bps, 8-bit data, 1 *stop bit*, tanpa *parity bit*). Dan menentukan definisi pin I/O (Readpin, Intpin, Data1, Data2, Data3, Data4).
2. Kemudian program melakukan deklarasi variabel Data_out untuk menampung 4-bit data dari pin Data1-Data4.
3. Kemudian program menetapkan kondisi Readpin menjadi high (keadaan awal / tidak melakukan proses baca).
4. Program akan menunggu kondisi Intpin hingga menjadi low yang berarti telah terjadi penekanan tombol.
5. Setelah menunda 2 μs, kondisi Readpin diubah menjadi low (akan melakukan proses baca) agar data hasil penekanan tombol dapat dibaca.
6. Setelah menunggu 200 μs, program membaca 4-bit data dari pin Data4-Data1 dan diletakkan pada variabel Data_out.
7. Program menunda 2 μs, lalu mengubah kondisi Readpin menjadi high (proses baca selesai).

8. Setelah itu, data dalam variabel `Data_out` segera dikirimkan ke komputer dengan format ASCII secara serial. Dan data ini dapat ditampilkan pada layar komputer menggunakan bantuan Terminal© atau HyperTerminal© (dengan *baudrate* 9600 bps, 8-bit data, 1 *stop bit*, tanpa *parity bit*, tanpa *flow control*).
9. Kembali ke langkah 4.

Listing program terdapat pada **AN78.ZIP**.

Selamat berinovasi!

BASCOM-AVR is copyright by MCS Electronics.
HyperTerminal is a copyright by Hilgraeve Inc.
Terminal is a copyright by Bray++.