

Antarmuka PC Keyboard dengan DST-AVR

Tata kerja keyboard PC

Setiap kali salah satu tombol keyboard ditekan atau dilepas, keyboard akan mengirim kode ke host (host adalah komputer kalau keyboard dihubungkan ke PC, atau berupa mikrokontroler kalau keyboard dihubungkan ke peralatan berbasis mikrokontroler). Kode tersebut dinamakan sebagai scan code.

Scan code tombol 'S' adalah **1B** (angka heksadesimal setara dengan angka biner 00011011). Ketika tombol 'S' ditekan keyboard akan mengirim **1B**, jika tombol 'S' ditekan terus maka keyboard akan mengirimkan **1B** berikutnya terus menerus, sampai ada tombol lain yang ditekan atau tombol 'S' tadi dilepas.

Keyboard juga mengirim kode saat ada satu tombol yang dilepas, kodenya adalah **F0** (angka heksadesimal setara dengan angka biner 11110000), jadi kalau tombol 'S' tadi dilepas keyboard akan mengirim **F0** dan **1B**.

Kode-kode tersebut dikirim keyboard secara seri, artinya dikirimkan satu bit demi satu bit. Misalnya **1B** dikirimkan dengan cara : mula-mula dikirim '1', sesaat kemudian '1' lagi dan menyusul '0' sampai akhirnya terkirim sebanyak 8 bit yang berbentuk 00011011 (dikirim mulai dari bit yang paling kanan kemudian bergeser satu per satu sampai yang paling kiri).

Masing-masing tombol punya scan code sendiri, termasuk tombol 'shift', tombol 'ctrl' dan lain lain, jadi jika tombol 'ctrl' ditekan bersama dengan 'S', maka scan code yang dikirim adalah 14 (scan code untuk 'ctrl') dan **1B** (scan code untuk 'S'). Terserah host untuk menginterpretasi scan code itu sebagai apa, berarti merupakan tugas program dalam mikrokontroler untuk mengenal scan code.

Scan code disusun sebagai kode 8 bit bisa dipakai untuk membedakan 256 macam kode, sedangkan keyboard PC hanya punya 101 tombol, jadi sesungguhnya kode 8 bit tadi cukup untuk semua tombol. Tapi tombol di keyboard PC dikelompokkan menjadi 2 bagian, bagian utama dan bagian tambahan, bagian utama cukup dinyatakan dengan scan code 1 byte saja, sedangkan bagian tambahan diwakili dengan beberapa byte scan code yang selalu diawali dengan **E0**. Misalnya tombol 'ctrl' kiri diwakili dengan **14** sedangkan tombol 'ctrl' kanan diwakili dengan **E0 14**.

Gambar 1 menggambarkan scan code masing-masing tombol keyboard PC. Terlihat pada gambar tersebut, scan code tidak berupa kode ASCII yang biasa dipakai mewakili huruf, dan ditentukan secara acak, juga. Sehingga setelah diterima host, scan code harus dirubah menjadi kode ASCII dengan memakai cara 'pencarian tabel'.

Gambar 1 **Keyboard PC dan Scan Code**

Komunikasi antar keyboard dan host adalah komunikasi dua arah, keyboard mengirim scan code ke host, host bisa mengirim perintah untuk mengatur kerja dari key board. Kode perintah untuk keyboard tidak sebanyak scan code, berikut ini daftar kode perintah untuk keyboard (dalam heksadesimal) selengkapnya :

ED perintah untuk me-nyala/padam-kan lampu indikator di keyboard, setelah menerima perintah **ED** dari host, keyboard akan menjawab dengan **FA** sebagai tanda perintah itu telah dikenali (**ACK** – acknowlwdge) dan menunggu 1 byte perintah lagi dari host untuk menentukan lampu indikator mana yang perlu di-nyala/padam-kan.

1 byte perintah susulan tersebut akan diartikan sebagai berikut : bit 0 dipakai untuk mengatur lampu indikator **Scroll Lock**, bit 1 untuk **Num Lock** dan bit 2 untuk **Caps Lock**, bit-bit lainnya diabaikan.

EE dipakai host untuk memeriksa apakah keyboard masih aktif. Setelah menerima perintah **EE** dari host, keyboard akan menjawab dengan **EE** pula, menandakan dirinya masih aktif.

F0 ada keyboard yang dilengkapi 3 set scan code, perintah ini dipakai untuk memilih scan code yang ingin dipakai. Setelah menerima perintah **F0** dari host, keyboard akan menjawab dengan **FA** sebagai tanda perintah itu telah dikenali (**ACK** – acknowlwdge) dan host menjawab 1 byte lagi (nilainya 1 2 atau 3) untuk memilih set scan code. Jika byte yang dikirimkan nilainya 0, keyboard akan menjawab dengan nomor set scan code yang saat itu dipakai.

F3 dipakai untuk mengatur kecepatan tanggapan keyboard (Typematic Repeat Rate), setelah menerima perintah **F3** dari host, keyboard akan menjawab dengan **FA**

sebagai tanda perintah itu telah dikenali (**ACK** – acknowledge) dan host menjawab 1 byte nilai kecepatan tanggapan keyboard yang dikehendaki.

F4 dipakai untuk me-aktif-kan kembali keyboard, setelah menerima perintah ini keyboard akan menjawab dengan **FA** (**ACK** – acknowledge).

F5 dipakai untuk me-nonaktif-kan keyboard, setelah menerima perintah ini keyboard akan menjawab dengan **FA** (**ACK** – acknowledge).

FE dipakai meminta keyboard mengirim ulang scan code terakhir yang dikirim.

FF Perintah untuk me-reset keyboard

Selain perintah dari host, keyboard juga mempunyai kode-kode lain selain scan code yang dikirimkan ke host, sebagai berikut :

FA berarti **ACK** (acknowledge), yaitu jawaban dari keyboard bahwa perintah dari host sudah dikenali dengan baik.

AA berarti keyboard selesai memeriksa diri dan siap bekerja setelah diberi catu daya

EE lihat perintah **EE** di atas

FE artinya minta host mengulang perintah terakhir yang dikirim

FF / **00** berarti terjadi kesalahan di keyboard

Rangkaian penghubung

Keyboard PC dibangun dengan mikrokontroler MCS48, yang merupakan saudara tua MCS51 tapi jauh lebih sederhana. Untuk keperluan membentuk rangkaian penghubung tidak perlu diketahui bagaimana cara kerja mikrokontroler dalam keyboard, tapi cukup meninjau rangkaian elektronik bagian penghubung pada gambar 2.

Gambar 2 Bagian penghubung di dalam Keyboard PC

Yang menarik, rangkaian sederhana ini bisa dipakai untuk komunikasi data 2 arah, yakni keyboard mengirimkan scan code ke PC, atau PC mengirimkan perintah-perintah ke keyboard, misalkan perintah untuk menyalakan beberapa lampu yang ada di keyboard.

Kbd Clock dibangkitkan oleh MCS48, merupakan sinyal pendorong **Kbd Data** yang bisa bersumber dari keyboard maupun bersumber dari PC. Level tegangan pada kedua sinyal ini memenuhi standar sinyal TTL biasa, jadi bisa langsung dihubungkan ke mikrokontroler.

Sumber daya untuk keyboard dicatu dari luar, harus diperhatikan kebutuhan arusnya cukup besar bisa sampai sekitar 300 mA.

Sinyal pengiriman data dari Keyboard

Saat tidak ada pengiriman data, sinyal **Kbd Clock** dan **Kbd Data** dalam keadaan '1'. Sinyal pengiriman data dari keyboard dalam gambar 3 dijelaskan sebagai berikut :

- Data mulai dikirimkan dengan me-nol-kan **Kbd Data** sebagai tanda mulai pengiriman (start bit), berapa saat kemudian setelah **Kbd Data** stabil disusul **Kbd Clock** berubah menjadi '0' dan kembali ke '1' lagi, ini berarti selesai mengirimkan data 1 bit.
- Setelah mengirim 'start bit', dikirimkan bit 0, bit 1 dan seterusnya sampai bit 7.
- Menyusul dikirim 'parity bit', yaitu bit kontrol yang berguna bagi host penerima data untuk memastikan data yang diterima tidak ada kesalahan. Jika banyaknya bit '1' yang terdapat di bit 0 sampai bit 7 ganjil, 'parity bit' akan bernilai '1'.
- Sebagai penutup (stop bit) **Kbd Data** dikembalikan ke keadaan normalnya, yaitu '1'.

Gambar 3 Sinyal komunikasi data seri dari Keyboard

Menerima data 1 bit dari Keyboard

Frekuensi `kbd clock` sekitar 20 sampai 30 KHz. Jumlah bit yang dikirim lewat `kbd Data` adalah 11, jadi `kbd clock` hanya mengirim 11 pulsa clock untuk mendorong 11 bit data tersebut satu per satu. Perubahan `KBD Clock` dari '1' ke '0' merupakan bagian terpenting dari bahasan di atas, pada saat itulah data di `KBD Data` boleh diambil.

Merubah ScanCode menjadi kode ASCII

Kalau tidak ada kesalahan, data yang diperoleh dari sub-rutin `AmbilByte` bisa berupa:

- scan code kalau nilai data tersebut antara `01` (Hex) sampai `83` (Hex).
- `F0` sebagai awalan dari scan code menandakan ada tombol dilepas
- `E0` sebagai awalan dari scan code tombol tambahan
- `FA, AA, EE, FE, FF, 00` yaitu kode-kode yang dipakai untuk menjawab perintah dari host

Kalau kode tersebut berupa scan code, kode tersebut harus diterjemahkan dulu dengan Potongan Program 4. Perubahan scan code menjadi kode ASCII yang dilakukan sub-rutin `JadikanASCII`, sub-rutin ini mengandalkan data yang sudah ditabelkan dalam `TabelScanCode`.

Header file ini bertujuan untuk menjembatani komunikasi antara mikrokontroler AVR khususnya tipe AT90S8515 dan AT90S8535 pada modul DST – 51 dengan keyboard AT 6 pin yang banyak dijumpai di pasaran.

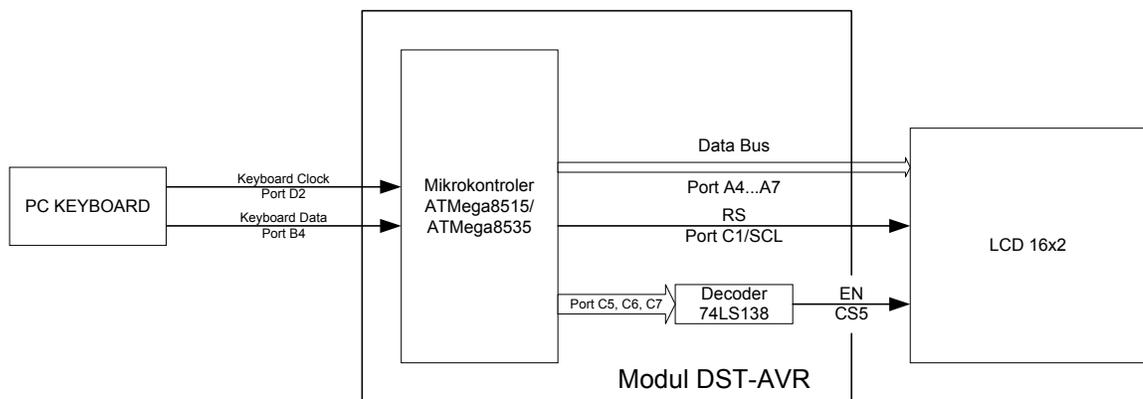
Adapun Fungsi yang terdapat di dalam header file ini adalah :

- **`ambil_data()`**
Fungsi ini digunakan untuk mengambil dan mengirimkan data scan code 1 byte dari keyboard ke mikrokontroler, fungsi ini menghasilkan output berupa data scan code yang bertipe **`unsigned char`**.
- **`Scancode_to_ASCII(unsigned char masukan)`**

Fungsi ini digunakan untuk mengkonversi/mengubah data scan code 1 byte menjadi data ASCII 1 byte, **variabel masukan** diisi dengan data scancode yang akan dikonversi kemudian fungsi akan menghasilkan output yang telah dikonversi berupa data ASCII yang bertipe **unsigned char**.

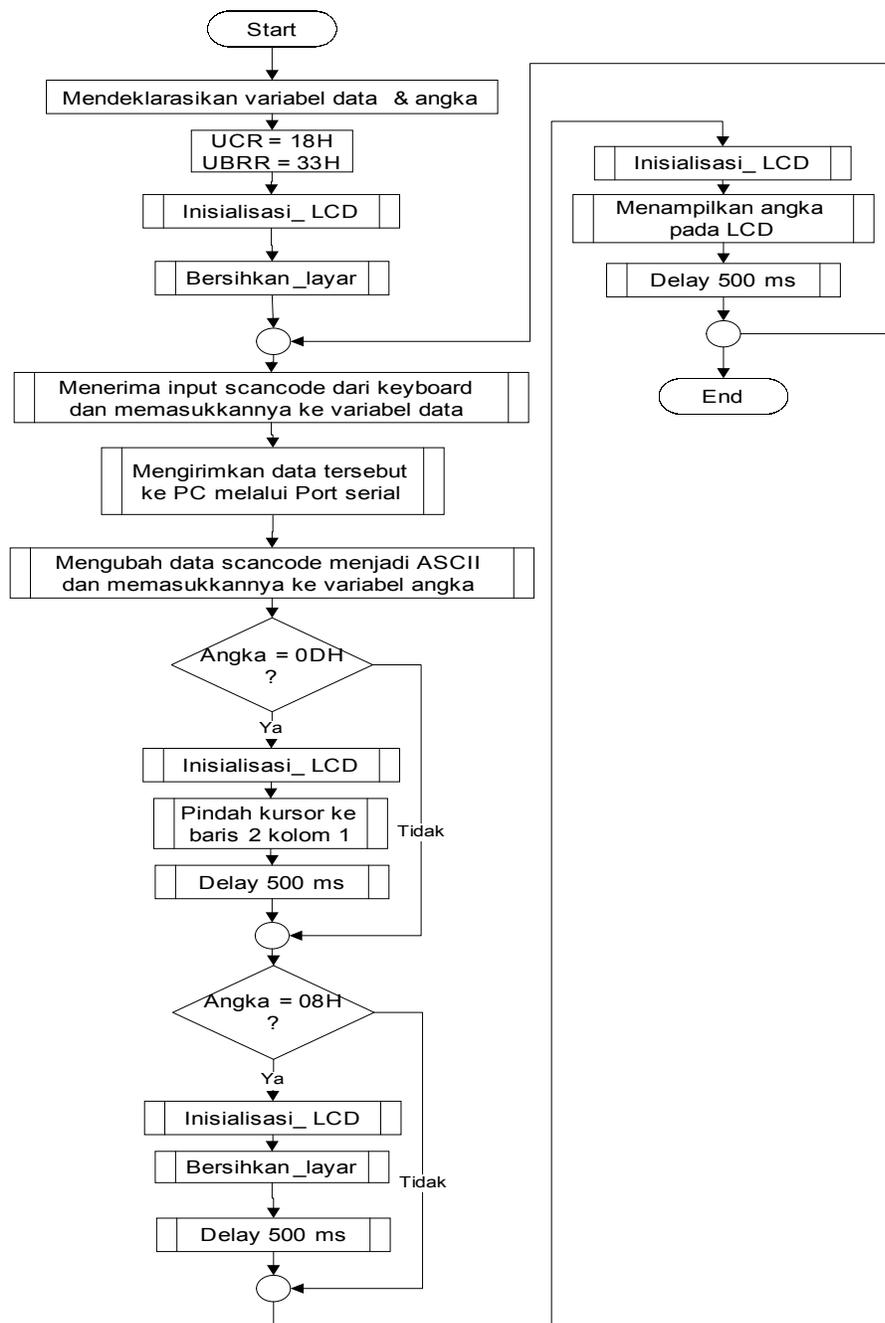
Aplikasi :

Program berikut ini bertujuan untuk menampilkan display dari tombol yang ditekan pada keyboard ke LCD LMB162A, serta mengidentifikasi tombol Enter pada keyboard (jika ditekan maka kursor pada LCD akan berganti baris), demikian pula dengan tombol BackSpace (penekanan aka bertujuan menghapus keseluruhan display, serta meletakkan kursor pada posisi awal, yaitu di pojok kiri atas)



Gambar 4
Block Diagram

Flowchart Program :



**Gambar 5
Flowchart**



Gambar 6
Modul DST-AVR

Aplikasi dan program ini dapat didownload dari AN1001 application note dari www.delta-electronic.com bagian Delta Kits, **Davin Tirta, Delta Electronic**