

Контроллер XT-клавиатуры на 51-ом контроллере.

Режим работы контроллера определяется двумя младшими битами кода передаваемого в контроллер по команде: 0x55 0x08 <mode>.

- 0 - Spectrum KBD
- 1 - RD Code KBD
- 2 - Mode CP/M
- 3 - Direct RD

Остальные биты не влияют на работу контроллера, хотя судя по предварительному описанию старший бит должен был переключать режим RUS/LAT.

mode=0 режим Spectrum KBD

В этом режиме контроллер эмулирует подключение к порту клавиатуры обычной матрицы 5*8 клавиш.

Каждое нажатие клавиши преобразуется в одно или два замыкания в узлах этой матрицы. При сканировании порта клавиатуры на выход выдается код соответствующий запрошенной адресной линии матрицы. Дополнительно к этому коду добавляется код считанный с "родного" порта клавиатуры, что позволяет считать линию входа магнитофонного интерфейса и сигналы с джойстиков или механической клавиатуры. Добавка идет командой OR.

Соответствие клавиш IBM-клавиатуры и матрицы Spectrums задается таблицей:

```
; Scan-code IBM(1) -> code Spectrum
; D7 - Symbol Shift
; D6..D4 - Number bit Adress (A8=000..A15=111)
; D3 - Caps Shift
; D2..D0 - Number bit Data (D0=001..D4=101)
```

```
L_4B6:
    db 39h ;01 ESC           CapSh + 1
;
    db 31h ;02 1/!
    db 32h ;03 2/@
    db 33h ;04 3/#
    db 34h ;05 4/$
    db 35h ;06 5/%
;
    db 45h ;07 6/^
    db 44h ;08 7/&
    db 43h ;09 8/*
    db 42h ;0A 9/(
    db 41h ;0B 0/)
;
    db 0E4h ;0C -/_         SymSh+Kl_J
    db 0E2h ;0D =/+         SymSh+Kl_L
    db 49h ;0E BS           CapSh+Kl_0
    db 3Bh ;0F TAB         CapSh+Kl_3
;
    db 21h ;10 Q
    db 22h ;11 W
    db 23h ;12 E
    db 24h ;13 R
    db 25h ;14 T
;
```

```

db 55h ;15 Y
db 54h ;16 U
db 53h ;17 I
db 52h ;18 O
db 51h ;19 P
;
db 0D5h ;1A [/{      SymSh+Kl_Y
db 0D4h ;1B ]/}      SymSh+Kl_U
db 61h ;1C ENTER
db 88h ;1D Ctrl      CapSh+SymSh
;
db 11h ;1E A
db 12h ;1F S
db 13h ;20 D
db 14h ;21 F
db 15h ;22 G
;
db 65h ;23 H
db 64h ;24 J
db 63h ;25 K
db 62h ;26 L
;
db 0D2h ;27 ;/:      SymSh+Kl_O
db 0D1h ;28 '/'"      SymSh+Kl_P
db 91h ;29 `/~      CapSh+Kl_A
;
db 08h ;2A Left Shift CapSh
db 92h ;2B \/|      CapSh+Kl_S
;
db 02h ;2C Z
db 03h ;2D X
db 04h ;2E C
db 05h ;2F V
;
db 75h ;30 B
db 74h ;31 N
db 73h ;32 M
;
db 0F4h ;33 ,/<      SymSh+Kl_N
db 0F3h ;34 ./>      SymSh+Kl_M
db 85h ;35 //?      SymSh+Kl_V
db 80h ;36 Rght Shift SymSh
db 0F5h ;37 [*]      SymSh+Kl_B
db 3Ch ;38 Alt      CapSh+Kl_4
;
db 71h ;39 SPACE
;
db 3Ah ;3A CapsLock  CapSh+Kl_2
;
db 0B1h ;3B F1      SymSh+Kl_1
db 0B2h ;3C F2      SymSh+Kl_2
db 0B3h ;3D F3      SymSh+Kl_3
db 0B4h ;3E F4      SymSh+Kl_4
db 0B5h ;3F F5      SymSh+Kl_5
db 0C5h ;40 F6      SymSh+Kl_6
db 0C4h ;41 F7      SymSh+Kl_7
db 0C3h ;42 F8      SymSh+Kl_8
db 0C2h ;43 F9      SymSh+Kl_9
db 0C1h ;44 F10     SymSh+Kl_0
;

```

```

db    0 ;45 NumLock
db    0 ;46 ScrollLock
;
db  3Ch ;47 [7]          CapSh+Kl_4
db  4Ch ;48 [8] [Up]    CapSh+Kl_7
db  3Dh ;49 [9]          CapSh+Kl_5
db  0E4h ;4A [-]         SymSh+Kl_J
db  3Dh ;4B [4]          CapSh+Kl_5
db  35h ;4C [5]          5
db  4Bh ;4D [6]          CapSh+Kl_8
db  0E3h ;4E [+]         SymSh+Kl_K
db  4Ah ;4F [1]          CapSh+Kl_9
db  4Dh ;50 [2]          CapSh+Kl_6
db  4Bh ;51 [3]          CapSh+Kl_8
db  84h ;52 [Insert]    SymSh+Kl_C
db  49h ;53 [Delete]    CapSh+Kl_0
db  0E5h ;57 F11        SymSh+Kl_H
db  94h ;58 F12        SymSh+Kl_F
;*****

```

mode=1 - RD code KBD

В этом режиме, независимо от состояния адресной линии, контроллер возвращает CP/M код последней нажатой клавиши.

При этом регистр контроллера, хранящий этот код сбрасывается в 0.

```
;*****
```

mode=2 - CP/M KBD

В этом режиме, в зависимости от состояния двух старших битов адреса сканирования, контроллер возвращает:

A15=0,A14=0 - CP/M код последней нажатой клавиши.

Регистр после чтения сбрасывается.

A15=0,A14=1 - состояние регистра управляющих клавиш;

```

d0 - Shift (1-нажата)
d1 - Ctrl  (1-нажата)
d2 - ALT   (1-нажата)
d3 - всегда 0
d4 - Caps Lock trigger
d5 - Num Lock trigger
d6 - Scroll Lock trigger
d7 - RUS(1)/LAT(0)

```

A15=1,A14=0 - состояние дополнительного регистра

```

d0 - Right Shift
d1..d7 - всегда 0

```

A15=1,A14=1 - не несет полезной информации

```
;*****
```

mode=3 - Direct RD

В этом режиме, независимо от состояния адресных линий, контроллер возвращает скан-код последней нажатой клавиши клавиатуры IBM, причем код XT-клавиатуры.

```
;*****
```

Управление контроллером:

Для управления контроллером при чтении порта клавиатуры выставляется адрес сканирования равный 0x55. При этом контроллер возвращает в качестве ответа код 0xAA. Это является признаком готовности контроллера к приему кода команды в следующем цикле чтения порта.

Код сканирования клавиатуры в цикле передачи команды, в 6 младших битах содержит собственно код команды, а в двух старших битах адресный код команды.

Необходимо отметить, что при приеме незадействован-

ного кода команды контроллер в ответ выдает код = 0xFF.

В частности в игре MINER наличие контроллера определяется выполнением команды с кодом = 0x00. Поскольку этот код не является командой, то контроллер отвечает на 0x55 -> 0xAA, а на 0x00 -> 0xFF. Этот факт и является признаком наличия контроллера.

Коды команд:

1) чтение версии прошивки контроллера

0x55h 0x01 - первый байт версии (обычно он и читается)

0x55h 0x41 - второй

0x55h 0x81 - третий

0x55h 0xC1 - четвертый

Эти команды читают содержимое ПЗУ контроллера начиная с адреса 0x2C. Судя по всему автор прошивки зарезервировал эти адреса для размещения кодов версии.

В XT-прошивке в этих адресах записаны коды 6,0,1,0.

Текст в конце прошивки при этом указывает V1.06.

В AT-прошивке в этих адресах записаны коды 1,0,0,0

Текст в конце прошивки при этом указывает V1.00.

В BIOS ATM проверяется только первый байт.

При этом наличие контроллера опознается сравнением этого байта либо с 10 либо с 16 (проверка производится дважды). То есть первый байт должен быть не более 10.

2) очистка буфера SPECTRUM клавиатуры

0x55h 0x07|0x47|0x87|0xC7

Очищается буфер, в котором хранятся скан-коды нажатых клавиш в режиме 0. Нажатие клавиши заносит ее скан-код в буфер, отжатие должно удалить этот скан-код из буфера. Похоже эта команда введена на всякий случай, поскольку при корректной работе контроллера буфер должен автоматически очищаться при нажатии и отжатии клавиш.

Команда возвращает коды 0AAh 0xFF.

3) Установка режима работы контроллера

0x55h 0x08|0x48|0x88|0xC8 <mode>

В результате выполнения этой команды в переменной контроллера сохраняется код <mode>.

Реально в контроллере используются только 2 мл. бита этого кода, определяющие текущий режим работы контроллера.

Команда возвращает коды 0AAh 0xFF.

4) Чтение регистров CP/M кода клавиатуры

0x55 0x09 - байт предыдущего скан-кода клавиатуры

0x55 0x49 - байт текущего скан-кода клавиатуры

0x55 0x89 - байт регистра управляющих клавиш;

d0 - Shift (1-нажата)

d1 - Ctrl (1-нажата)

d2 - ALT (1-нажата)

d3 - всегда 0

d4 - Caps Lock trigger

d5 - Num Lock trigger

d6 - Scroll Lock trigger

d7 - RUS(1)/LAT(0)

0x55 0xC9 - байт дополнительного регистра

d0 - Right Shift

d1..d7 - всегда 0

По сути эти команды дублируют чтение регистров контроллера, выполняемое при его работе в режиме 2.

5) Установка режима RUS
0x55 0x0A|0x4A|0x8A|0xCA

Эта команда устанавливает в 1 бит d7 регистра управляющих клавиш контроллера.

Команда возвращает коды 0AAh 0xFF.

6) Установка режима LAT
0x55 0x0B|0x4B|0x8B|0xCB

Эта команда сбрасывает в 0 бит d7 регистра управляющих клавиш контроллера.

Команда возвращает коды 0AAh 0xFF.

7) Установка режима ожидания.
0x55 0x0C|0x4C|0x8C|0xCC

Команда устанавливает контроллер в режим выдачи сигнала /WAIT на процессор Z80. То есть она эквивалентна нажатию клавиши PAUSE/BREAK. Для выхода из этого режима надо нажать эту клавишу.

Команда возвращает коды 0AAh 0xFF.

8) Выдача сигнала /RESET
0x55 0x0D|0x4D|0x8D|0xCD

На Z80 выдается сигнал сброса, что перезагружает всю систему. Эквивалентно нажатию Ctrl/Alt/Del

9) Чтение регистров текущего времени

0x55 0x10 - секунды

0x55 0x50 - минуты

0x55 0x90 - часы

0x55 0xD0 - дни

Читается содержимое регистров часов реального времени, реализованных на основе встроенного таймера микроконтроллера. Таймер в обеих прошивках настроен на работу при частоте тактирования контроллера 7 МГц. При отключении питания таймер естественно сбрасывается.

10) Установка регистров текущего времени

0x55 0x11 <секунды>

0x55 0x51 <минуты>

0x55 0x91 <часы>

0x55 0xD1 <дни>

Эти команды записывают передаваемые данные в регистры часов реального времени.

11) Чтение регистров текущей даты

0x55 0x12 - секунды

0x55 0x52 - минуты

0x55 0x92 - часы

0x55 0xD2 - дни

12) Запись регистров текущей даты

0x55 0x13 <секунды>

0x55 0x53 <минуты>

0x55 0x93 <часы>

0x55 0xD3 <дни>

13) Установка в 1 линий порта P1

0x55 0x14|0x54|0x94|0xD4 <cod>

Выполняется команда:

```
orl    P1,<cod>
```

14) Сброс в 0 линий порта P1

0x55 0x15|0x55|0x95|0xD5 <cod>

Выполняется команда

```
cpl    <cod> ;инверсия кода
anl    P1,<cod>
```

Предполагалось, видимо, что эти команды позволят программно управлять последовательным портом контроллера. Но я считаю их использование не целесообразным, тем более, что реально можно задействовать только биты 3 (DTR) и 4 (RTS). С остальными битами при таком способе управления можно получить непредсказуемый результат работы контроллера.

```
; порт P1
; P10 - CD      input
; P11 - CTS     input
; P12 - RI      input
; P13 - DTR     out
; P14 - RTS     out
; P15 - INT_T   out
; P16 - /RES    -out
; P17 - W_ON    -out
```

15) Чтение линий порта P3

0x55 0x16|0x56|0x96|0xD6

Команда должна выполнять чтение состояния линий порта P3 контроллера, но в результате ошибки всегда выдает код 0xFF.

```
; порт P3
; P30 - RX      input
; P31 - TX      -output
; P32 - CLK_K   input
; P33 - /KEYRD  input
; P34 - VE1     input
; P35 - DATA_K input
; P36 - /VWR    -output
; P37 - /VRD    -output
```

Судя по линиям порта, команда годится только для анализа линии VE1.

16) Чтение линий порта P1

0x55 0x17|0x57|0x97|0xD7

Та же самая ошибка приводит к чтению 0xFF.

```
; порт P1
; P10 - CD      input
; P11 - CTS     input
; P12 - RI      input
; P13 - DTR     out
; P14 - RTS     out
; P15 - INT_T   out
; P16 - /RES    -out
; P17 - W_ON    -out
```

Команда годится для чтения линий CD, CTS, RI.

```
;-----
```

```

; Таблица CP/M - кодов клавиатуры
; код CP/M ; Клавиша IBM
L_400: db 1Bh ; ESC
       db 31h ; 1
       db 32h ; 2
       db 33h ; 3
       db 34h ; 4
       db 35h ; 5
       db 36h ; 6
       db 37h ; 7
       db 38h ; 8
       db 39h ; 9
       db 30h ; 0
       db 2Dh ; -/_
       db 3Dh ; =/+
       db 8 ; BS
;
       db 9 ; TAB
       db 51h ; Q
       db 57h ; W
       db 45h ; E
       db 52h ; R
       db 54h ; T
       db 59h ; Y
       db 55h ; U
       db 49h ; I
       db 4Fh ; O
       db 50h ; P
       db 5Bh ; [
       db 5Dh ; ]
       db 0Dh ; Enter
;
       db 41h ; A
       db 53h ; S
       db 44h ; D
       db 46h ; F
       db 47h ; G
       db 48h ; H
       db 4Ah ; J
       db 4Bh ; K
       db 4Ch ; L
       db 3Bh ; ;
       db 27h ; '
       db 60h ; `
       db 5Ch ; \
;
       db 5Ah ; Z
       db 58h ; X
       db 43h ; C
       db 56h ; V
       db 42h ; B
       db 4Eh ; N
       db 4Dh ; M
       db 2Ch ; ,
       db 2Eh ; .
       db 2Fh ; /
       db 0AAh ; \ / |
       db 20h ; SPACE
;
       db 61h ; F1

```

```

db 62h ; F2
db 63h ; F3
db 64h ; F4
db 65h ; F5
db 66h ; F6
db 67h ; F7
db 68h ; F8
db 69h ; F9
db 6Ah ; F10
;-----
; Num keyboard scan code set1 > 47h
db 80h+37h ; [7]
db 80h+38h ; [8]
db 80h+39h ; [9]
db 80h+2Dh ; [-]
db 80h+34h ; [4]
db 80h+35h ; [5]
db 80h+36h ; [6]
db 80h+2Bh ; [+]
db 80h+31h ; [1]
db 80h+32h ; [2]
db 80h+33h ; [3]
db 80h+30h ; [0]
db 80h+2Eh ; [.]
;
db 6Bh ; F11
db 6Ch ; F12
;-----
; Эта таблица для Num Pad в режиме
; Num Lock=0
L_511: db 76h ;47h Home
db 70h ;48h Cur Up
db 74h ;49h Page Up
db 0ADh ;4Ah [-]
db 72h ;4Bh Cur Left
db 0B5h ;4Ch [5]
db 73h ;4Dh Cur Right
db 0ABh ;4Eh [+]
db 77h ;4Fh End
db 71h ;50h Cur Down
db 75h ;51h Page Down
db 78h ;52h Insert
db 79h ;53h Del
;=====

```