MSX — Программируемый периферийный интерфейс (PPI)

В MSX PPI (программируемый периферийный интерфейс) — это микросхема, используемая для переключения слотов, управления клавиатурой (включая звук «щелчка») и кассетным магнитофоном.

PPI, используемый системами MSX, основан на Intel 8255, хотя в большинстве систем присутствует клон или функциональность интегрирована в микросхему MSX Engine.

Клавиатура

Описание клавиатуры приведено на странице: Русификация MSX.

Пример опроса клавиатуры без использования подпрограмм ПЗУ

Пример опроса клавиатуры без использования подпрограмм MSX — ПЗУ, автор **KoD/SDM**

Для опроса клавиатуры используются два порта — A9 и AA:

Регистр	Порт	Описание
A9	B	Колонка матрицы клавиатуры (Только чтение) Чтение из этого порта вернёт состояние выбранной строки клавиатуры (выбранной через порт AA) Биты 0–7 соответствует одной из 8 клавиш в каждой строке матрицы клавиатуры **Если клавиша нажата, то соответствующий бит устанавливается в 0**
AA	C	Строка матрицы клавиатуры, LED, Cassette (Чтение/Запись) Биты 0-3 соответствуют строке матрицы клавиатуры

Полное описание регистров/портов приведено ниже.

Пример интернациональной раскладки матрицы клавиатуры:

	Порт #A9							
Порт #AA	**бит 7**	**бит 6**	**бит 5**	**бит 4**	**бит 3**	**бит 2**	**бит 1**	**бит 0**
строка 0	7	6	5	4	3	2	1	0
строка 1	;	]	[	\	=	-	9	8
строка 2	b	a	DEAD	/	.	,	`	'
строка 3	j	i	h	g	f	e	d	c
строка 4	r	q	p	o	n	m	l	k
строка 5	z	y	x	w	v	u	t	s
строка 6	F3	F2	F1	CODE	CAPS	GRAPH	CTRL	SHIFT
строка 7	RET	SELECT	BS	STOP	TAB	ESC	F5	F4
строка 8	→	↓	↑	←	DEL	INS	HOME	SPACE

Это пример раскладки без «шифта» и не расширенной клавиатуры:

```
;-------------------------------------------------------
NAMBAS              EQU     #F922         ; адрес  PNT
KEY_MATRIX_LAYOUT   EQU     #0200         ; адрес алрес таблицы матрицы КРАТНЫЙ 256!!!

    org     #100

    di
```

```
    ld hl,(NAMBAS)
    ld a,l
    out (#99),a
    ld a,h
    out (#99),a
    nop
    in a,(#98)
    ld (save_byte+1),a
    ld hl,key_matrix_layout
    ld de,KEY_MATRIX_LAYOUT
    ld bc,8*9
    ldir
main_loop:
    call KEY_SCAN
    and a
    jr z,main_loop            ; клавиши не нажимались
    ld h,KEY_MATRIX_LAYOUT/256    ; H=старший байт таблмцы
    ld a,(hl)
    cp #1B                ; ESC?
    jr z,exit_2_dos
    push af
    ld hl,(NAMBAS)
    ld a,l
    out (#99),a
    ld a,h
    or #40
    out (#99),a
    pop af
    out (#98),a            ; печатаем символ
    jr main_loop
exit_2_dos:
    ld hl,(NAMBAS)
    ld a,l
    out (#99),a
    ld a,h
    or #40
    out (#99),a
save_byte:
    ld a,0
    out (#98),a            ; восстанавливаем что было на экране
    ei
    ret
;-------------------------------------------------------
; На выходе- L=индекс клавиши; A=флаг (1-нажатие 0-нет)
KEY_SCAN:
    ld l,0            ; индекс нажатой клавиши
    ld e,0            ; строка матрицы клавиатуры
key_scan_loop2:
    ld b,8            ; одни ряд - 8 клавиш
    in a,(#AA)
        and #F0                 ; очищаем 4 младших бита (биты 0-3 номер строки)
    or e                    ; устанавливаем строку матрицы клавиатуры
        out (#AA),a
        in a,(#A9)              ; читаем состояние строки матрицы клавиатуры
key_scan_loop1:
    rrca
    jr nc,key_pressed         ; переход если клавиша нажата
    inc l                 ; увеличиваем индекс
    djnz key_scan_loop1
    inc e                 ; следующая строка матрицы клавиатуры
    ld a,e
    cp #09
    jr nz,key_scan_loop2     ; переход если не все строки обработаны
    xor a                 ; все строки обработаны, клавиши не нажимались
```

```
    ret
key_pressed:
    ld a,1              ; клавиши нажимались
    ret
;-------------------------------------------------------------------------------
key_matrix_layout:
;строка 0
    db    "01234567"
;строка 1
    db    "89-=\[];"
;строка 2
    db    "'`,./ ab"
;строка 3
    db    "cdefghij"
;строка 4
    db    "klmnopqr"
;строка 5
    db    "stuvwxyz"
;строка 6
    db    "         "
;строка 7
    db    "  ",#1B,"     "     ; #1B-ESC
;строка 8
    db    "         "
;-------------------------------------------------------------------------------
```

Более подробно см. здесь: MSX Assembly Page: Keyboard matrices

Регистры

Здесь приведено соответствие регистров PPI и портов ввода/вывода:

Регистр	Порт	Бит	Режим (ввод/выход)	Имя	Назначение
A	A8	0	ввод/выход	CS0 L	Выбор слота для адресов в диапазоне 0000-3FFF
		1		CS0 H	
		2		CS1 L	Выбор слота для адресов в диапазоне 4000-7FFF
		3		CS1 H	
		4		CS2 L	Выбор слота для адресов в диапазоне 8000-BFFF
		5		CS2 H	
		6		CS3 L	Выбор слота для адресов в диапазоне C000-FFFF
		7		CS3 H	
B	A9	0	ввод		Состояние строки матрицы клавиатуры, указанной через порт AA битами KB1–KB3 **Если клавиша нажата, то соответствующий бит устанавливается в 0**
		1			
		2			
		3			
		4			
		5			
		6			
		7			

Регистр	Порт	Бит	Режим (ввод/выход)	Имя	Назначение
C	AA	0	выход	KB0	Выбор строки матрицы клавиатуры для сканирования
		1		KB1	
		2		KB2	
		3		KB3	
		4		CASON	CAS CTRL (сброс для включения = 1) — Включение мотора магнитофона
		5		CASW	Включен режим записи на магнитофон
		6		CAPS	CAPS LAMP (сброс для включения = 1) Вкл./выкл. режим ввода заглавных букв (индикатор на клавише CAPS)
		7		SOUND	Вывод звука (1, затем 0, чтобы издать звук «щелчка» клавиатуры)
Управление	AB	0	вход		статус бита для изменения
		1			Номер бита, который необходимо изменить в порту C PPI
		2			
		3			
		4			Не используется
		5			
		6			
		7			Всегда должен быть сброшен

Ссылки

PPI

Архитектура и устройства микрокомпьютеров стандарта MSX-2 — Порты программируемого периферийного интерфейса (PPI ports)

http://sysadminmosaic.ru/msx/ppi

2022-09-01 21:14