

Описание протокола PS/2 для мыши и клавиатуры.

Николай Ковач

6-8 минут

В предыдущей статье был описан интерфейс PS/2 на физическом уровне передачи байт в обе стороны. А какие же байты передаются там? Каков протокол? Попробуем ответить на эти вопросы.

Возможно самый простой способ изучения интерфейса – посмотреть фрагмент кода драйвера для порта PS/2, а именно микросхемы i8042. В Windows DDK есть пример этого драйвера `\src\input\pnpi8042`. Из этого примера становится ясно многое хотя и не все. Попробую дать некоторые разъяснения.

На каждую принятую от компьютера команду, или проще сказать на каждый принятый байт, клавиатура/мышь должны обязательно ответить одним из следующих байтов:

- **ACKNOWLEDGE** = **0xFA** – подтверждение об успешном приеме;
- **RESEND** = **0xFE** - команда принята с ошибкой (вероятно ошибка CRC);
- **FAILURE** = **0xFC** – произошла ошибка (не знаю, что это такое, может внутренняя ошибка устройства?).

Если компьютер примет от клавиатуры или мыши не **0xFA**, а **0xFE** или не дай бог **0xFC**, то скорее всего будет пытаться переповторить посылку команды или последнего посланного байта.

Для клавиатуры компьютер может послать следующие команды:

- **SET_KEYBOARD_INDICATORS** = **0xED** – зажечь или потушить светодиоды CAPS/NUM /SCROLL. Если клавиатура принимает эту команду, то больше она не пошлет ничего, до тех пор, пока компьютер не пришлет следующий байт-параметр. Этот параметр определяет битовую маску – один бит – это один светодиод.

Битовая маска для светодиодов клавиатуры определена вот так:

```
#define KEYBOARD_KANA_LOCK_ON    8 // Japanese keyboard
```

```
#define KEYBOARD_CAPS_LOCK_ON    4
```

```
#define KEYBOARD_NUM_LOCK_ON     2
```

```
#define KEYBOARD_SCROLL_LOCK_ON  1
```

- **SELECT_SCAN_CODE_SET** = **0xF0** – установить текущую таблицу кодов клавиш. Следом будет байт-параметр, номер выбираемой таблицы;

- **READ_KEYBOARD_ID = 0xF2** – не знаю, что это такое. Драйвер из WinDDK похоже не использует эту команду.
- **SET_KEYBOARD_TYPEMATIC = 0xF3** – это тоже двухбайтовая команда. После этой команды следует параметр определяющий частоту повтора кодов при нажатой клавише и интервал времени между нажатием и началом повторов. Параметр байт typematic выглядит следующим образом:

o	d6	d5	t4	t3	t2	t1	to
----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------

Бит 7 - всегда ноль.

Биты **d5** and **d6** определяют задержку (Delay) от времени нажатия, когда посылается первый код, до момента когда начинаются повторы кодов, если удерживать клавишу. Задержку можно вычислить по формуле $(1 + \text{typematic}[6:5]) * 250$ миллисекунд.

Биты **t4**, **t3**, **t2**, **t1**, **to** определяют частоту повторов кодов при удерживании клавиши нажатой. Период повторов можно вычислить по формуле:

$\text{Period} = (8 + \text{typematic}[2:0]) * (2^{\text{typematic}[4:3]}) * 0.00417$ секунд.

- **KEYBOARD_RESET = 0xFF** – получая эту команду клавиатура отвечает, как обычно, **0xFA**, а затем, сбрасывается и посылает в ответ байт **KEYBOARD_COMPLETE_SUCCESS = 0xAA**.

Еще важное дополнение.

После подачи напряжения питания клавиатура посылает компьютеру код **KEYBOARD_COMPLETE_SUCCESS = 0xAA** и немедленно готова к работе. Она сразу, без дополнительного программирования, готова посылать коды нажатых клавиш к компьютеру.

По умолчанию клавиатура посылает на нажатие один байт-код, а на отжатие клавиши два байта. Первый байт в кодах "отпускания клавиши" – это префикс отжатия **KEYBOARD_BREAK_CODE = 0xF0**.

Например, если нажать и отпустить клавишу «1», то клавиатура пошлет **0x16**, **0xF0**, **0x16**.

Существуют еще так называемые дополнительные extended коды, это еще два префикса **0xE0** и **0xE1**. Они посылаются вместе с кодами «дополнительных» клавиш.

Полный перечень сканкодов клавиатуры можно взять здесь:

Теперь перейдем к рассмотрению протокола PS/2 мыши (обычной 3-х кнопочной с колесом).

Придется немного огорчить читателя тем, что запрограммировать мышь не легче, чем клавиатуру, а даже и труднее.

Здесь есть две проблемы.

Первая проблема состоит в том, что после включения питания мышь (в отличие от клавиатуры) не готова к работе. После включения питания мышь посылает **MOUSE_COMPLETE_SUCCESS = 0xAA** и следом **MOUSE_ID = 0x00**. Требуется запрограммировать мышь, «разрешить передачу». Для этого, как минимум, нужно послать ей специальную команду **ENABLE_MOUSE_TRANSMISSION = 0xF4**. Только после этой команды мышь начнет посылать пакеты с координатами и состоянием кнопок. Вторая проблема состоит в том, что по умолчанию мышь работает «без колеса прокрутки». Для того, чтобы включить «колесо» нужно выполнить совершенно дурацкую и длинную

инициализацию мыши. Кто мог додуматься до этого странного решения я не могу понять.

Однако все по порядку. Наиболее важные команды для мыши вот такие:

- **SET_MOUSE_RESOLUTION** = **0xE8**
- **SET_MOUSE_SAMPLING_RATE** = **0xF3**
- **ENABLE_MOUSE_TRANSMISSION** = **0xF4**
- **SET_MOUSE_SCALING_1TO1** = **0xE6**
- **READ_MOUSE_STATUS** = **0xE9**
- **GET_DEVICE_ID** = **0xF2**
- **MOUSE_RESET** = **0xFF**

Некоторые из них идут в паре с параметром, некоторые – это одиночные команды.

Мышь без колеса прокрутки посылает пакеты по три байта, а мышь с колесом пакеты по четыре байта.

Чтобы «включить колесо» компьютер посылает в мышь последовательность: **0xF3, 200, 0xF3, 100, 0xF3, 80, 0xF2** (на каждый из этих байт мышь отвечает **ACK = 0xFA**).

Последняя команда здесь – это **GET_DEVICE_ID = 0xF2**. После этой последовательности мышь должна прислать ID. Если пришел ноль, значит колеса нет и не будет. Если пришло «3», то значит колесо прокрутки включено. Но и это еще не все программирование. Мышь будет «ездить медленно», если больше ничего не делать. Обычно скорость мыши увеличивают программируя примерно так, посылая команды: **0xE8, 3, 0xE6, 0xF3, 40**, ну и напоследок нужно разрешить передачу послав команду **0xF4**.

Теперь осталось только описать формат пакетов мыши. Координаты передаются как относительное смещение.

Первый байт в пакете передает знак перемещения SY и SX (вверх-вниз и влево-вправо), а также состояние трех кнопок BM (middle - центральная), BR (right - правая), BL (left - левая):

1	0	SY	SX	1	BM	BR	BL
---	---	----	----	---	----	----	----

Второй байт передает смещение по координате X.

X7	X6	X5	X4	X3	X2	X1	X0
----	----	----	----	----	----	----	----

Третий байт передает смещение по координате Y.

Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
----	----	----	----	----	----	----	----

Четвертый байт посылается только для мышей с колесом и только если оно включено.

Определяет вращение колеса.

0	0	0	0	Z3	Z2	Z1	Z0
---	---	---	---	----	----	----	----

Ну вот пожалуй это и все, что нужно знать для программирования простых PS/2 устройств.

