

--- ФИРМА НИППЕЛЬ ПРЕДСТАВЛЯЕТ ---

ОПИСАНИЕ КОНТРОЛЛЕРА МАНИПУЛЯТОРА МЫШЬ

"NIPPEL MOUSE CARD"

для ЭВМ Агат 9

Руководство пользователя и программиста

Автор: Голоз А.А.

© Ниппель 1993

ВНИМАНИЕ

Перед установкой контроллера манипулятора мышь в ЭВМ внимательно ознакомьтесь с инструкцией по эксплуатации. Помните, что неправильная установка может привести к выходу из строя не только собственно контроллера, но и Вашей ЭВМ Агат.

ОСНОВНЫЕ ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

Контроллер "Nippel Mouse Card" предназначен для совместной работы с устройством ввода координатным - мышью, "Марсианка" УВК-01 или УВК-07, далее УВК, производства МОКБ "Марс" и поставляется жестко с ним связанным (кабель припаян к контроллеру). Небольшие габариты контроллера позволили разместить его непосредственно в упаковочной коробке УВК и играть роль адаптирующего разъема.

Несмотря на малые размеры (75x45мм) платы контроллера, она содержит 9 интегральных микросхем и обладает рядом преимуществ перед используемым ранее в качестве контроллера мыши, контроллером принтера.

На плате установлены два семиразрядных счетчика, способных производить 64 отсчета в каждом направлении без обращения к контроллеру процессора, что позволило разгрузить последний и избежать потерь импульсов из манипулятора.

Кроме того, контроллер надежно идентифицируется именно как контроллер мыши, что не вызывает конфликтов, как в случае использования в машине двух контроллеров принтера, и способен работать в первом разъеме Агата-9.

Данный контроллер является стандартом фирмы Ниппель на управление УВК мышью и все разрабатываемые фирмой программы будут придерживаться именно этого стандарта.

ИНСТРУКЦИЯ ПО ЭКСПЛУАТАЦИИ

1. Меры предосторожности.

Контроллер мыши "Nippel Mouse Card" не слишком прихотлив в работе и использовании. Установленные в контроллере микросхемы не подвержены выходу из строя от статических зарядов, поэтому пользователю важно соблюдать осторожность, только чтобы не повредить плату или элементы механически.

Для этого старайтесь брать плату только за торцы, не бросайте контроллер где попало, лучше всего, вынув его из машины, уложите сразу в упаковочную коробку вместе с УВК мышь.

2. Установка контроллера.

Установку и изъятие контроллера из ЭВМ производите только при отключенной питании. Знайте, что наибольшая вероятность отказа контроллера или ЭВМ возникает именно при изъятии или установке плат в необесточенную машину.

Откройте крышку Вашей ЭВМ Агат 9 и осмотрите, какие разъемы у нее не заняты. Согласно принятой терминологии, разъемы у Агата 9 нумеруются слева направо (если смотреть с фронтальной стороны корпуса), начиная с первого и так далее до шестого.

В принципе "Nippel Mouse Card" может быть установлена Вами в любой свободный разъем машины, но наиболее удобным для Вас, видимо, будет установка ее в первый (самый левый) разъем ЭВМ, который у Вас скорее всего свободен, так как другие, разработанные производителем Агата 9 модули не способны функционировать в первом разьеме, и он как правило пустует.

Установка "Nippel Mouse Card" в машину требует предельной внимательности, так как контроллер не имеет ключа, не позволяющего установить его неправильно. Поэтому усвойте сразу контроллер мыши нужно сначала просунуть через щель в задней стенке, после чего Вам уже будет

тяжело или, точнее сказать неудобно, установить его неверно, тем не менее отметим, что провод должен выходить к задней стенке. Есть общее правило установки плат; элементы, находящиеся на плате, должны располагаться справа (вид с фронтальной стороны). В случае с "Nippel Mouse Card" это не совсем верно, так как на ней установлены элементы с обеих сторон платы, но на основной (правой) стороне элементов значительно больше.

При установке платы удерживайте ее двумя указательными пальцами за боковые торцы и, надавливая двумя большими пальцами на верхний торец платы, немного покачивая вдоль разъема, углубите ее до упора. Еще раз проверьте, правильно ли Вы установили контроллер - детали справа, провод к УВК сзади и, только убедившись в правильности, можете подать питание на ЭВМ.

3. Проверка контроллера.

Для проверки "Nippel Mouse Card" лучше всего воспользоваться графическим редактором "MouseGraf" версии 4.0 или выше. Если же у Вас его нет, то можно воспользоваться простой программкой находящейся на магнитной диске, входящей в комплект с контроллером. На нем Вы найдете два объектных файла типов 'В' и 'К', называющихся соответственно BTESTMOUSE и KTESTMOUSE. Это совершенно одинаковые программы, представленные в два типа только во избежании осложнений с запуском в конкретной используемой Вами системе программирования.

Подготовьте мышь к работе, как это сделать, можно узнать из описания, входящего в ее комплект, и запустите программу TestMouse. На экране появится изображение курсора. Двигая мышью по гладкой поверхности, Вы должны наблюдать синхронные перемещения курсора по экрану. Нажатие правой кнопки мыши должно приводить к закрашиванию экрана белым цветом, левой кнопки - черным.

Чтобы выйти из программы TestMouse, нужно нажать на клавиатуре УПР К.

РУКОВОДСТВО ДЛЯ ПРОГРАММИСТА

В руководстве для программиста мы расскажем о внутренней архитектуре контроллера "Nippel Mouse Card", способах доступа и управления им, а также приведем рекомендации и примеры программного обеспечения и поддержки некоторых функций контроллера.

1. Внутренняя структура модуля.

Основой контроллера "Nippel Mouse Card" являются два семиразрядных реверсивных двоичных счетчика, по одному для каждой из координат X и Y. Приходящие от УВК сигналы направления воздействуют на входы направления счета и тактирования счетчиков, которые производят отсчет импульсов. Область счета простирается от -64 до 63. При разрешающей способности УВК 0,5мм и частоте опроса контроллера 50Гц (частота встроенного в Agat-9 прерывания NMI) допускается скорость перемещения манипулятора в одном направлении 1,5м/с, что, правда, не допускается самим манипулятором (см. технические хар-ки УВК).

Такие большие размерности счетчиков применены в связи с планируемым увеличением точности УВК как минимум вдвое, а также допустимым из-за зрительного эффекта снижением частоты опроса до 25Гц.

Для опроса счетчиков и кнопок выделены восемь адресов в области Device Select, т.е. области \$C0X0-C0XF. Чтобы согласовать возможное размещение на одной плате нескольких устройств, таких как, контроллеры принтера и новой клавиатуры, модуль часов и предлагаемого контроллера мыши, были выбраны непересекающиеся адреса обращения ко всем этим устройствам. Как известно, контроллер принтера использует адреса \$C0X0-\$C0X3, часы \$C0X6-\$C0X7, для мыши же зарезервировали адреса \$C0X8-\$C0XC.

В таблице 1 представлено распределение адресов контроллера "Nippel Mouse Card" и указано их назначение.

Таблица 1

Адрес	Назначение
\$C0X8	чт: четыре младших бита смещения по X
\$C0X8	зп: сброс счетчиков в значение \$22
\$C0X9	чт: три старших бита смещения по X и правая кнопка
\$C0XA	чт: четыре младших бита смещения по Y
\$C0XB	чт: три старших бита смещения по Y и левая кнопка
\$C0XC	чт: то же, что и чтение из \$C0X8
\$C0XC	зп: сброс счетчиков в значение \$00

Примечание: чт: означает чтение, зп: - запись

Из считанных по соответствующему адресу восьми битов только младшие четыре имеют валидное значение, старшие же в связи с тем, что они не подключены к шине, могут принимать произвольное значение, поэтому для надежности у каждого считанного байта нужно обнулять старшие биты командой AND #\$F.

2. Основные приемы и процедуры

Наличие двух разных сигналов сброса счетчиков используется для надежной идентификации контроллера при поиске. Появление по одним и тем же адресам двух разных константных значений (0 и 2) позволяет с весьма высокой точностью утверждать, что это может быть только контроллер "Nippel Mouse Card".

В связи с тем, что, как было сказано выше, на одном модуле могут быть размещены многие устройства, процедуру поиска конкретного устройства лучше всего оформить отдельно и вызывать вместе с другими подобными процедурами в блоке. Каждая из процедур сформирует свой признак наличия/отсутствия конкретного устройства и его номер разъема.

Ниже представлена процедура для определения номера разъема, на котором установлен контроллер "Nippel Mouse Card". По выходу из процедуры, номер разъема, умноженный на \$10, находится в ячейке SLOT, ноль означает отсутствие контроллера.

```

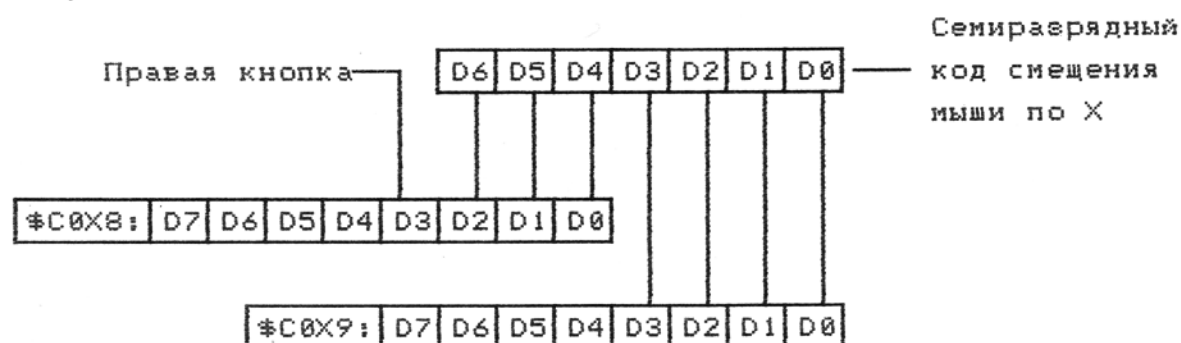
*=====
* Найти контроллер "Nippel Mouse Card" *
*=====
SEARMS   LDA #0           - Контроллер отсутствует
          STA SLOT       - ;
          LDX ##10       Ищем начиная с первого
SEARMS0  LDA ##FF        Константа на всякий случай
          STA #C08C,X    Сбросить в ноль
          LDA #C088,X    Прочитать содержимое
          AND ##F
          BNE SEARMS1    = Это не мыш
          LDA ##FF        Константа на всякий случай
          STA #C088,X    Сбросить в два
          LDA #C088,X    Прочитать содержимое
          AND ##F
          CMP #2
          BNE SEARMS1    = Это не мыш
          STA #C08C,X    Сбросить регистры
          STX SLOT       - Мышь найдена
          RTS           - ;
SEARMS1  TXA             - Перейти к следующему разъему
          CLC
          ADC ##10
          TAX
          CMP ##70
          BCC SEARMS0    = Еще не все разъемы просмотрены - ;
          RTS           - ;

```

Найденный таким образом номер разъема можно использовать в остальных процедурах для доступа к счетчикам контроллера.

Из таблицы 1 видно, что по адресам #C0X8 и #C0XA читаются четыре младших бита соответственно смещений по X и Y, а по адресам #C0X9 и #C0XB по три старших бита смещения, оставшиеся биты индицируют состояние кнопок.

Рисунок 1



Код смещения для координаты Y и левой кнопки формируется аналогичным образом из содержимого ячеек #C0XA и #C0XB.

Основной проблемой, возникающей при формировании кода, является возможное изменение кода второй половинки во время операций доступа. Иными словами, например, между операциями чтения младшего и старшего полубайтов младший может досчитать до нуля и совершить перенос в старший, при этом курсор совершит скачок сразу на 16 позиций. Чтобы разрешить эту проблему, нужно совершать операции чтения дважды и, анализируя результаты обоих прочтений, установить правильное значение. Каким образом это делается, можно будет увидеть из программы слежения за мышью представленной ниже.

Сформированный согласно рисунку 1 семиразрядный код смещения представлен в дополнительном коде, т.е. отрицательные значения формируются вычитанием из нуля -1 - \$7F, -2 - \$7E, -3 - \$7D и т.д. Этот код удобен тем, что для формирования новой координаты нужно просто прибавить смещение в дополнительном коде к текущей координате. По правилам дополнительного кода для того, чтобы сформировать код большей разрядности чем исходный, нужно распространить старший разряд (разряд знака) на все расширяемые разряды. В нашем случае разряд D6 нужно распространить и на разряд D7. Все это производится в приводимой ниже программе.

В программе используются следующие ячейки:

XPOS и YPOS	хранят текущую координату.
RTBUT и LFBUT	состояние правой и левой кнопок соответственно
SLOT	номер разъема контроллера мыши.
MDATA0 и MDATA1	буферы по четыре байта каждый.
DELTA X и DELTA Y	вычисляемые значения ΔX и ΔY
ACCUM, XREG, YREG	для хранения регистров.

Программа обработки прерываний NMI должна быть сначала подключенной к вектору, например, можно разместить ее адрес в ячейках \$FFFA, \$FFFB, после чего можно запустить прерывания, например, командой BIT \$C040.

Когда придет прерывание NMI, процессор передаст управление по адресу, указанному в ячейках \$FFFA, \$FFFB, т.е. в нашем случае на программу NMI.

После сохранения регистров программа дважды считывает текущее состояние контроллера в промежуточные буферы MDATA0 и MDATA1, как было указано выше, для коррекции возможных ошибок. Следующей вызывается подпрограмма, проверяющая наличие ошибок и корректирующая их. Подпрограмма DELTAXY формирует восьмибитные значения смещения в дополнительном коде. И наконец, подпрограмма CALCPOS вычислит новую координату путем сложения старой с числом в дополнительном коде.

```

*=====*
* Обработка прерываний 50 Гц для слежения за мышью *
*=====*
NMI      STA ACCUM      - Сохранить содержимое регистров
          STX XREG
          STY YREG      - ;
          CLD           Для процессора 6502
          JSR RDDATA    Прочитать данные из контроллера
          JSR NORMDAT   Нормализовать данные буфера MDATA0
          JSR DELTAXY   Установить значение дельта
          JSR CALCPOS   Вычислить новую позицию курсора
          LDA ACCUM     - Восстановить содержимое регистров
          LDX XREG
          LDY YREG      - ;
          RTI

```

```

*-----*
* Прочитать данные из контроллера *
*-----*
RDDATA   LDA SLOT      - Указатель на $C0XB
          ORA ##B
          PHA
          TAX          - ;
          LDY #3       - Считать данные в первый буфер
RDDATA0  LDA $C080,X
          AND ##F      Отбросить старшие четыре бита
          STA MDATA0,Y
          DEX
          DEY
          BPL RDDATA0  - ;
          PLA          - Считать данные во второй буфер
          TAX
          LDY #3
RDDATA1  LDA $C080,X
          AND ##F      Отбросить старшие четыре бита
          STA MDATA1,Y
          DEX
          DEY
          BPL RDDATA1  - ;
          LDX SLOT     - Сбросить счетчики в ноль
          STA $C08C,X  - ;
          RTS

```

```

*-----*
* Произвести нормализацию данных буфера MDATA0 *
*-----*
NORMDAT  LDA MDATA0+1  - Удалить кнопки из старших тетрад
          AND #:111
          STA MDATA0+1  $C0X9
          LDA MDATA0+3
          AND #:111
          STA MDATA0+3  $C0XB - ;
          LDA MDATA1+1  - Сравнить старшие триады X
          AND #:111
          CMP MDATA0+1  $C0X9
          BEQ NORMDAT0  = Содержимое MDATA0..+1 валидно - ;
          LDA MDATA1
          STA MDATA0     $C0X8
          LDA MDATA1+1
          STA MDATA0+1  $C0X9 - ;
NORMDAT0 LDA MDATA1+3  - Сравнить старшие триады Y
          AND #:111
          CMP MDATA0+3  $C0XB
          BEQ NORMDAT1  = Содержимое MDATA..+3 валидно - ;
          LDA MDATA1+2  - Переслать валидно значение MDATA1
          STA MDATA0+2  $C0XA
          LDA MDATA1+3
          STA MDATA0+3  $C0XB - ;
NORMDAT1 RTS

```

```

*-----*
* Вычислить значения дельта X и дельта Y *
*-----*
DELTAXY  LDA MDATA0+1  - Получить значение дельта X
          AND #:111
          ASL A
          ASL A
          ASL A
          ASL A
          ORA MDATA0
          STA DELTAX
          BIT DELTAX
          BVC DELTAXY0  = Положительное смещение
          ORA #:10000000  Распространить на старший разряд
          STA DELTAX    - ;
DELTAXY0 LDA MDATA0+3  - Получить значение дельта Y
          AND #:111
          ASL A
          ASL A
          ASL A
          ASL A
          ORA MDATA0+2
          STA DELTAY
          BIT DELTAY
          BVC DELTAXY1  = положительное смещение
          ORA #:10000000  Распространить на старший разряд
          STA DELTAY    - ;
DELTAXY1 RTS

```

```

*-----*
* Вычислить новую позиция курсора и состояние кнопок *
*-----*
CALCPOS  LDA XPOS          - Новая позиция по X
          CLC
          ADC DELTAX
          STA XPOS          - ;
          LDA YPOS          - Новая позиция по Y
          CLC
          ADC DELTAY
          STA YPOS - ;
          LDA MDATA1+3 - Значение левой кнопки
          AND #:00001000
          ASL A
          ASL A
          ASL A
          ASL A
          STA LFBUT - ;
          LDA MDATA1+1 - Значение правой кнопки
          AND #:00001000
          ASL A
          ASL A
          ASL A
          ASL A
          STA RTBUT - ;
          RTS

```

Таким образом, мы получаем независимо работающую программу, поддерживающую изменение координаты курсора синхронно с перемещением мыши. Если в другой программе, ответственной за собственно перемещение курсора, сначала буферизовать состояние текущей координаты и вывести по ней курсор, а потом в цикле сравнивать буферизованную координату с новой XPOS и YPOS, то несовпадение координат будет означать смещение мыши, и значит нужно вывести курсор на новое место.

Естественно, что представленная выше программа не является идеальной и для многих случаев потребуются ее доработка, например, может быть недостаточным иметь по одному байту для текущей координаты. Нужно только помнить, что при фиксации текущих положений XPOS и YPOS в случае, если они, например, будут иметь размер два байта, нужно запрещать прерывания, так как между чтением первого и второго байтов может прийти прерывание и изменить состояние второго байта настолько, что это приведет к ошибочному результату.