



МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
имени Н.Э. БАУМАНА

Учебное пособие

Методические указания
по выполнению домашних заданий
по единому комплексному заданию по блоку дисциплины

«Системотехника ЭВС, комплексы и сети»

МГТУ имени Н.Э. Баумана

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
имени Н.Э. БАУМАНА

Методические указания
по выполнению домашних заданий
по единому комплексному заданию по блоку дисциплины

«Системотехника ЭВС, комплексы и сети»

Москва
МГТУ имени Н.Э. Баумана

2012

УДК 681.3.06(075.8)
ББК 32.973-018
И201

Методические указания по выполнению домашних заданий по единому комплексному заданию по блоку дисциплины «Системотехника ЭВС, комплексы и сети» / Коллектив авторов –
М.: МГТУ им. Н.Э. Баумана, 2012. – 29 с.: ил.

В методических указаниях рассмотрены основные этапы, их последовательность и содержание по выполнению домашних заданий курсовой работы по единому комплексному заданию по блоку дисциплины «Системотехника ЭВС, комплексы и сети».

Ил. 39. Табл. 5. Библиогр. 7 назв.

УДК 681.3.06(075.8)

© МГТУ им. Н.Э. Баумана, 2012

СОДЕРЖАНИЕ

Введение.....	5
1 Принцип работы оптических манипуляторов.....	7
2 Обзор оптического манипулятора на примере мыши GENIUS.....	11
3 Снятие временных диаграмм на разъеме манипулятора (PS/2).....	15
4 «Особенности» передачи по PS/2.....	22
5 Программа на Ассемблере для работы с указателем	24
Список использованных источников.....	27
Приложение 1.....	28
Приложение 2.....	29

Введение

История "мыши" восходит к сороковым годам и связана с развитием радиолокационной техники. Системы автоматического слежения за целями в те времена отсутствовали, и операторы первых радиолокационных станций (РЛС) определяли их координаты, прикладывая к экрану специальную линейку, а результат сообщали голосом. Позднее с отметкой цели стали совмещать специальный маркер (по теперешней терминологии — "курсор"), формируемый на экране локатора электронным методом. Положение маркера можно было автоматически передать, например, в систему наведения зенитного орудия.

Для управления маркером было придумано специальное устройство, которое у англичан получило название "target tracking ball" (шар для сопровождения целей), а у нас — "**шаровой манипулятор**", хотя чаще неофициально использовалось немецкое название "кнюппель". Устройство представляло собой шар размером с бильярдный (нередко это и был самый настоящий бильярдный шар), вмонтированный в стол оператора и снабженный датчиками вращения. Оператор управлял маркером, вращая шар ладонью. Это оказалось настолько удобным, что "кнюппель" до сих пор остается принадлежностью многих пультов операторов РЛС.

В поисках удобного способа управления курсором на экране компьютера конструкторы обратились к тому же шаровому манипулятору. Революционным было решение перевернуть его шаром вниз и поместить в небольшую коробочку. Для такого манипулятора не требуется специального стола, его можно перемещать по любой плоской поверхности.

Первоначально манипулятор предназначался для компьютерных игр, почему и получил "детское" название "**мышь**". Между прочим, когда подобным манипулятором был оснащен отечественный компьютер ЕС-1841, ему придумали название "колобок". Но распространения оно не получило, возможно потому, что дети редко играли с этим компьютером и не знакомились с его документацией. В последнее время "мышь" снова перевернули шаром вверх — таким манипулятором часто снабжают малогабаритные компьютеры типа "Notebook".

В процессе «эволюции» компьютерной мыши наибольшие изменения претерпели датчики перемещения.

Изначальная конструкция датчика перемещения мыши, изобретённой Дугласом Энгельбартом в Стенфордском исследовательском институте в 1963 году, состояла из двух перпендикулярных колес, выступающих из корпуса устройства. При перемещении колеса мыши крутились каждое в своем измерении. Такая конструкция имела много недостатков и довольно скоро была заменена на мышь с шаровым приводом.

В **шаровом приводе** движение мыши передается на выступающий из корпуса обрешиненный стальной шарик (его вес и резиновое покрытие обеспечивают хорошее сцепление с рабочей поверхностью). Два прижатых к шару ролика снимают его движения по каждому из измерений и передают их на датчики, преобразующие эти движения в электрические сигналы.

Основной недостаток шарового привода — загрязнение шарика и снимающих роликов, приводящее к заеданию мыши и необходимости в периодической её чистке (отчасти эта проблема сглаживалась путём металлизации роликов). Несмотря на недостатки, шаровой привод долгое время доминировал, успешно конкурируя с альтернативными схемами датчиков. В настоящее время шаровые мыши почти полностью вытеснены оптическими мышами.

Оптические датчики призваны непосредственно отслеживать перемещение рабочей поверхности относительно мыши. Исключение механической составляющей обеспечивало более высокую надёжность и позволяло увеличить разрешающую способность детектора.

Первое поколение оптических датчиков было представлено различными схемами оптопарных датчиков с непрямой оптической связью — светоизлучающих и воспринимающих отражение от рабочей поверхности светочувствительных диодов. Такие датчики имели одно общее свойство — они требовали наличия на рабочей поверхности (мышинном коврик) специальной штриховки (перпендикулярными или ромбовидными линиями). На некоторых ковриках эти штриховки выполнялись красками, невидимыми при обычном свете (такие коврики даже могли иметь рисунок). Недостатками таких датчиков обычно называют: необходимость использования специального коврика и невозможность его замены другим. Кроме всего прочего, коврики разных оптических мышей часто не были взаимозаменяемыми и не выпускались отдельно; необходимость определённой ориентации мыши относительно коврика, в противном случае мышь работала неправильно; чувствительность мыши к загрязнению коврика (ведь он соприкасается с рукой пользователя) — датчик неуверенно воспринимал штриховку на загрязнённых местах коврика; высокую стоимость устройства.

Второе поколение оптических мышей имеет более сложное устройство. В нижней части мыши установлен специальный светодиод, который подсвечивает поверхность, по которой перемещается мышь. Миниатюрная камера «фотографирует» поверхность более тысячи раз в секунду, передавая эти данные процессору, который и делает выводы об изменении координат. Оптические мыши второго поколения имеют огромное преимущество перед первым: они не требуют специального коврика и работают практически на любых поверхностях, кроме зеркальных. Они также не нуждаются в чистке.

Предполагалось, что такие мыши будут работать на произвольной поверхности, однако вскоре выяснилось, что многие продаваемые модели (в особенности первые широко продаваемые устройства) не так уж и безразличны к рисункам на коврике. На некоторых участках рисунка графический процессор способен сильно ошибаться, что приводит к хаотичным движениям указателя, не отвечающих реальному перемещению. Для склонных к таким сбоям мышей необходимо подобрать коврик с иным рисунком или вовсе с однотонным покрытием.

Датчики второго поколения постепенно совершенствуются, и в настоящее время мыши, склонные к сбоям, встречаются гораздо реже. Кроме совершенствования датчиков, некоторые модели оборудуются **двумя датчиками перемещения** сразу, что позволяет, анализируя изменения сразу на двух участках поверхности, исключать возможные ошибки. Такие мыши иногда способны работать на стеклянных, оргстеклянных и зеркальных поверхностях (на которых не работают другие мыши).

В последние годы была разработана новая, более совершенная разновидность оптического датчика, использующего для подсветки **полупроводниковый лазер**.

О недостатках таких датчиков пока известно мало, но известно об их преимуществах:

- более высоких надёжности и разрешении
- отсутствию заметного свечения (сенсору достаточно слабой подсветки лазером видимого или, возможно, инфракрасного диапазона)
- низком энергопотреблении.

1 Структурная работа манипулятора (на примере оптического манипулятора)

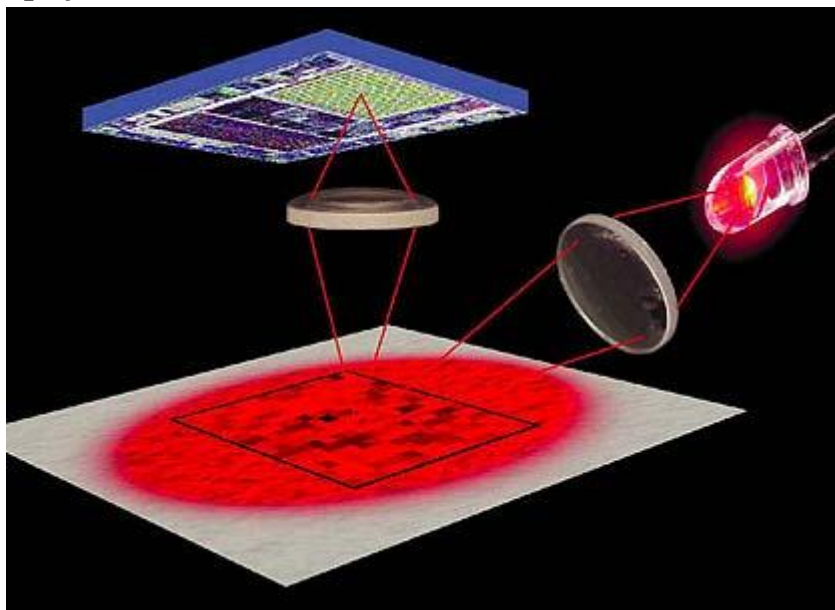


Рисунок 1.1 – Как «видят» компьютерные оптические мыши

Итак, «зрение» оптическая компьютерная мышь получает благодаря следующему процессу. С помощью светодиода, и системы фокусирующих его свет линз, под мышью подсвечивается участок поверхности. Отраженный от этой поверхности свет, в свою очередь, собирается другой линзой и попадает на приемный сенсор микросхемы — процессора обработки изображений. Этот чип, в свою очередь, делает снимки поверхности под мышью с высокой частотой (кГц). Причем микросхема (назовем ее оптический сенсор) не только делает снимки, но сама же их и обрабатывает, так как содержит две ключевых части: систему получения изображения Image Acquisition System (IAS) и интегрированный DSP процессор обработки снимков.

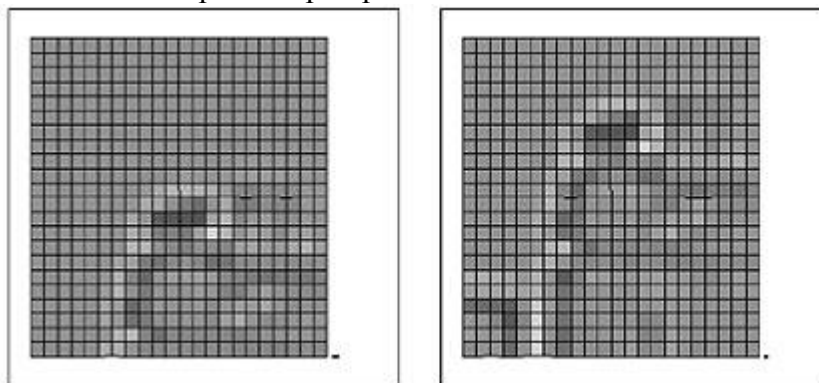
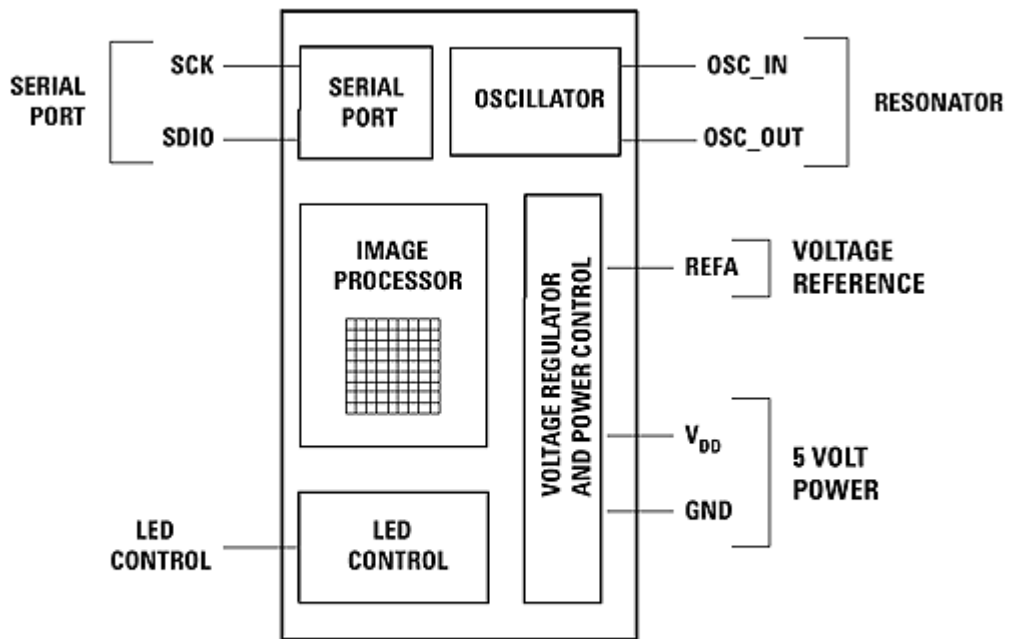


Рисунок 1.2 – 2 последовательных кадра

На основании анализа череды последовательных снимков (представляющих собой квадратную матрицу из пикселей разной яркости), интегрированный DSP процессор высчитывает результирующие показатели, свидетельствующие о направлении перемещения мыши вдоль осей X и Y, и передает результаты своей работы вовне по последовательному порту.

Если мы посмотрим на блок-схему одного из оптических сенсоров, то увидим, что микросхема состоит из нескольких блоков, а именно:



Block diagram of ADNS-2610 optical mouse sensor.

Рисунок 1.3 – Блок схема оптического сенсора мыши

- основной блок, это, конечно же, Image Processor — процессор обработки изображений (DSP) со встроенным приемником светового сигнала (IAS);
- Voltage Regulator And Power Control — блок регулировки напряжения и контроля энергопотребления (в этот блок подается питание и к нему же подсоединен дополнительный внешний фильтр напряжения);
- Oscillator — на этот блок чипа подается внешний сигнал с задающего кварцевого генератора, частота входящего сигнала порядка пары десятков МГц;
- Led Control — это блок управления светодиодом, с помощью которого подсвечивается поверхность по мышью;
- Serial Port — блок передающий данные о направлении перемещения мыши вовне микросхемы.

Нужно уточнить, что информацию о перемещении мыши микросхема оптического сенсора передает через Serial Port не напрямую в компьютер. Данные поступают к еще одной микросхеме-контроллеру, установленной в мышь. Эта вторая «главная» микросхема в устройстве отвечает за реакцию на нажатие кнопок мыши, вращение колеса прокрутки и т.д. Данный чип, в том числе, уже непосредственно передает в ПК информацию о направлении перемещения мыши, конвертируя данные, поступающие с оптического сенсора, в передаваемые по интерфейсам PS/2 или USB сигналы. А уже компьютер, используя драйвер мыши, на основании поступившей по этим интерфейсам информации, перемещает курсор-указатель по экрану монитора.

Именно по причине наличия этой «второй» микросхемы-контроллера, точнее благодаря разным типам таких микросхем, довольно заметно отличались между собой уже первые модели оптических мышей. Если о дорогих устройствах от Microsoft и Logitech слишком плохо отозваться я не могу (хотя и они не были вовсе «безгрешны»), то масса

появившихся вслед за ними недорогих манипуляторов вела себя не вполне адекватно. При движении этих мышей по обычным коврикам курсоры на экране совершали странные кульбиты, скакали чуть ли не на пол Рабочего стола, а иногда... иногда они даже отправлялись в самостоятельное путешествие по экрану, когда пользователь совершенно не трогал мышь. Доходило и до того, что мышь могла запросто выводить компьютер из режима ожидания, ошибочно регистрируя перемещение, когда манипулятор на самом деле никто не трогал.

В ведущие себя совершенно по-разному мыши часто устанавливались совершенно одинаковые микросхемы оптических сенсоров (благо, моделей этих чипов было не так уж много, как мы увидим далее). Однако вот, благодаря несовершенным чипам контроллеров, устанавливаемых в оптические мыши, мы имели возможность сильно поругать первые поколения оптических грызунов.

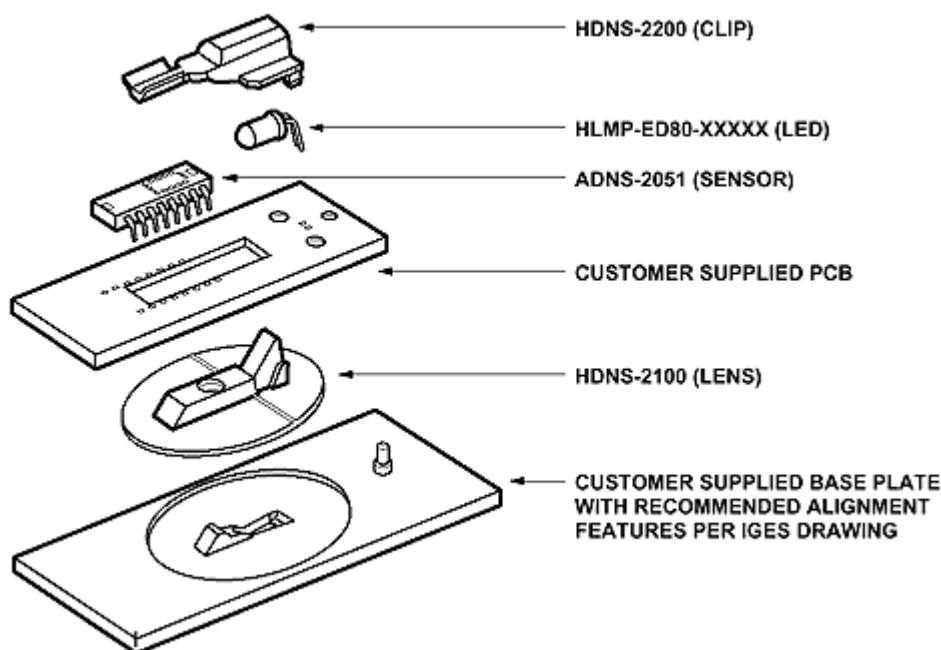
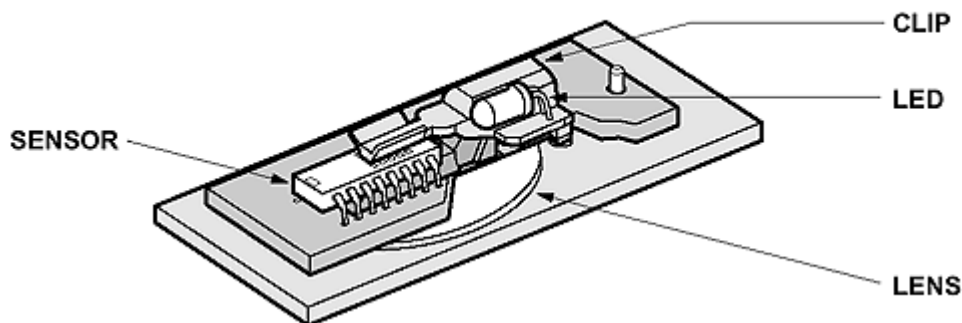
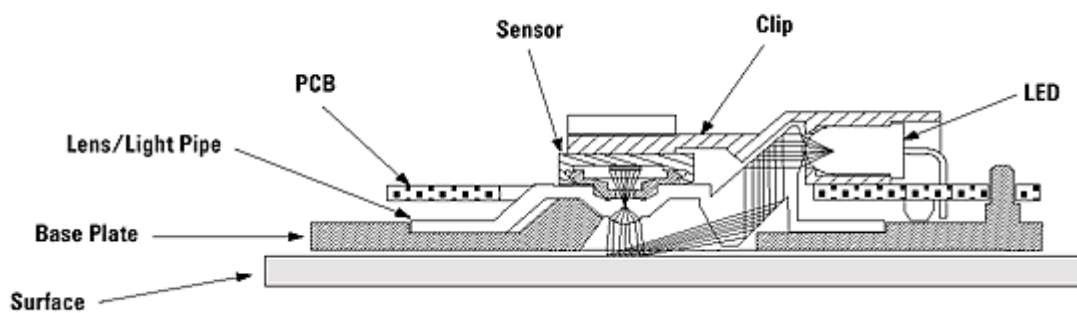


Рисунок 1.4 – Сборка элементов манипулятора на плате

В целом система оптического слежения мышей, помимо микросхемы-сенсора, включает еще несколько базовых элементов. Конструкция включает держатель (Clip) в который устанавливаются светодиод (LED) и непосредственно сама микросхема сенсора (Sensor). Эта система элементов крепится на печатную плату (PCB), между которой и нижней поверхностью мыши (Base Plate) закрепляется пластиковый элемент (Lens), содержащий две линзы (о назначении которых было написано выше).



В собранном виде оптический элемент слежения выглядит как показано выше. Схема работы оптики этой системы представлена ниже.



Оптимальное расстояние от элемента Lens до отражающей поверхности под мышью должно попадать в диапазон от 2.3 до 2.5 мм. Это рекомендации производителя сенсоров. Вот вам и первая причина, почему оптические мыши плохо себя чувствуют «ползая» по оргстеклу на столе, всевозможным «полупрозрачным» коврикам и т. п. И не стоит клеить на оптические мыши «толстые» ножки, когда отваливаются или стираются старые.

2 Обзор оптического манипулятора на примере мыши Genius

2.1 Обзор платы

Данный манипулятор является светодиодным.

Внешний вид платы манипулятора представлен на рисунке 2.1.1.

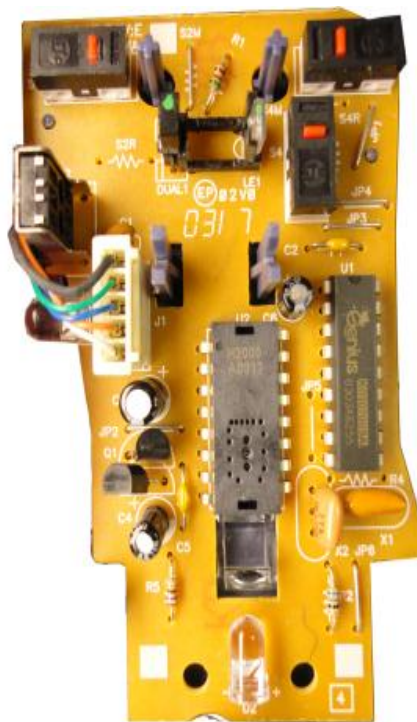


Рисунок 2.1.1 – Внешний вид платы манипулятора GENIUS

Схематическое изображение платы манипулятора Genius представлено на рисунке 2.2.1.

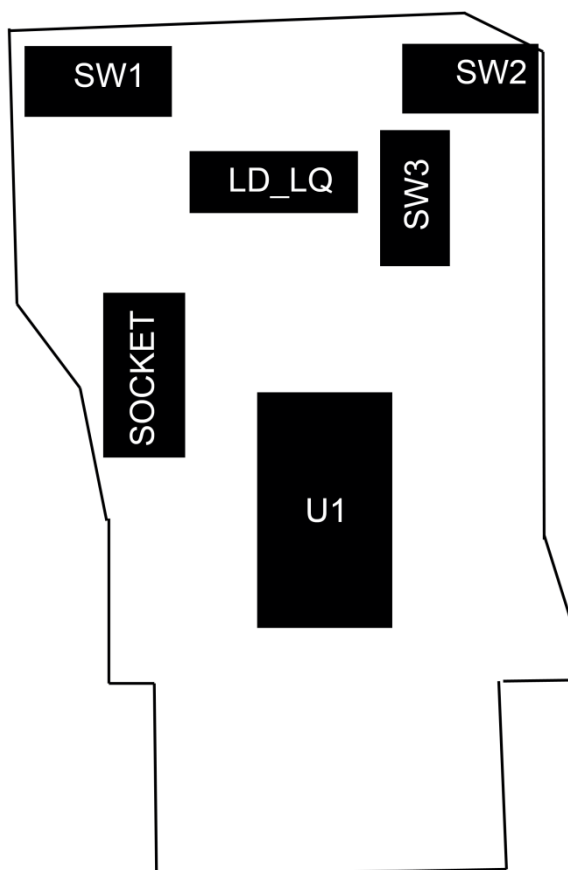


Рисунок 2.2.1 – Схематическое изображение платы манипулятора GENIUS

В таблице 2.1.1 представлены описание основных элементов платы.

Таблица 2.1.1 – Основные элементы

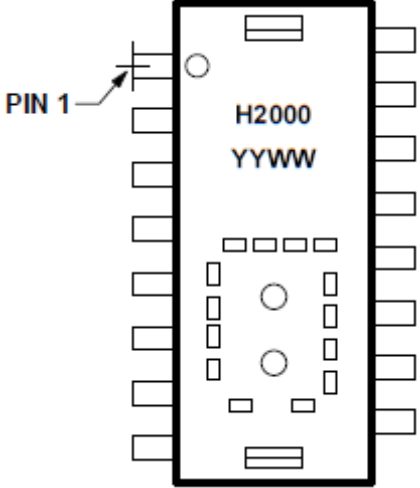
Обозначение	Предназначение
U1 (HDNS-2000)	Оптический датчик с встроенным микроконтроллером
SW1	Кнопка левого клика
SW2	Кнопка правого клика
SW3	Кнопка для колеса прокрутки
LD_LQ	Светодиод и фотоприемник(инфракрасный) для регистрирования прокручивания колеса прокрутки
SOCKET	Разъем для подключения к USB

2.1.1 Описание оптического датчика

HDNS-2000

Микросхема представляет собой оптический датчик. Цоколевка микросхемы представлена в таблице 2.1.1.2.

Таблица 2.1.1.2 – Разлиновка микросхемы HDNS-2000

Внешний вид	Pin	Наименование вывода	Назначение вывода
	1	PS2_C	PS/2 Interface Clock
	2	MODE/XA	Select PS/2 mode
	3	RX/XB	Right Button input
	4	MB/YB	Middle Button input
	5	LB/YA	Left Button input
	6	XY_LED	LED control output
	7	V _{DD3}	3.3 VDC input
	8	REFB	Internal reference
	9	OSC1	Oscillator input
	10	GND	Ground
	11	OSC2	Oscillator output
	12	GND	Ground
	13	V _{DD5}	5 VDC input
	14	V _{DD5}	5 VDC input
	15	NRESET	NRESET
	16	PS2_D	PS/2 Interface Data

Параметры микросхемы представлены в таблице 1.7

Таблица 1.7 – Параметры микросхемы ADNS-7550

Параметр	Значение
Диапазон рабочего напряжения, В	4.0-5.25
Скорость распознавания движения, кадр/сек	30
Внутренний генератор	Отсутствует, необходимо подключение
Длина волны излучения, нм	639
Количество регистров данных движения (motion data registers)	12

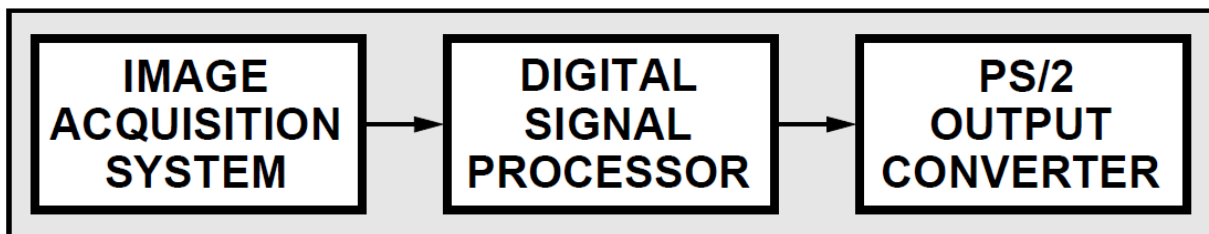


Рисунок 2.1.1.2 – Схема структурная датчика

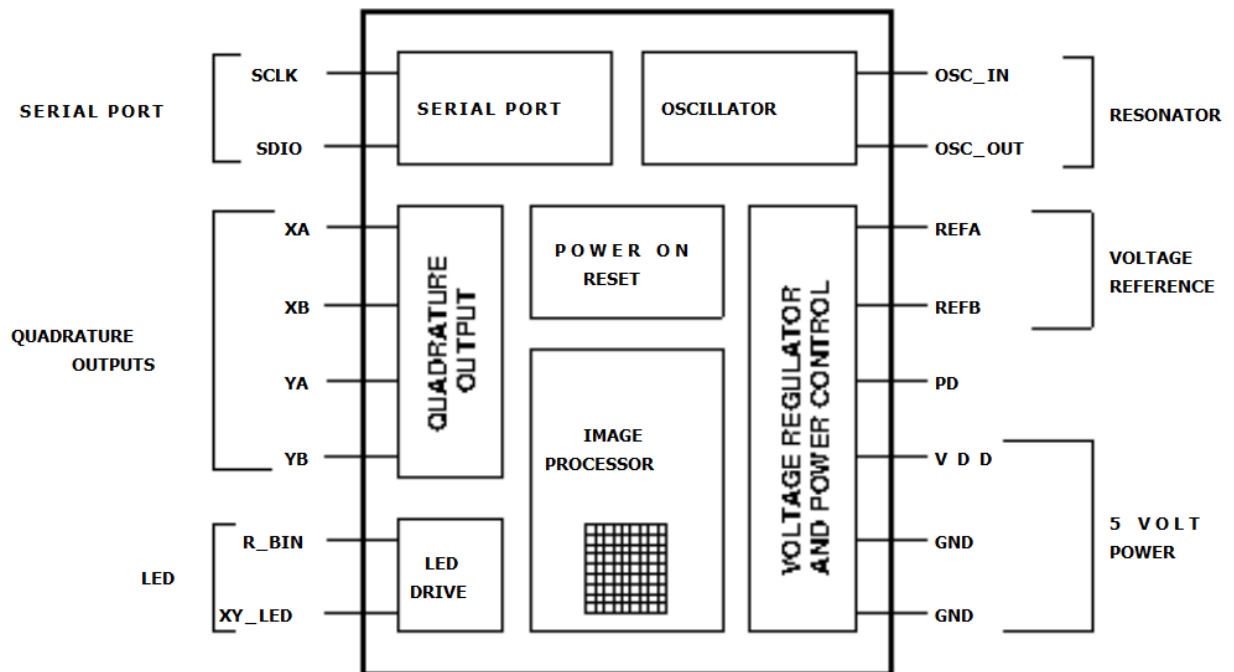


Рисунок 2.1.1.3 – Блок схема датчика

Работа датчика основана измерении изменения в положении переданных последовательно изображений (кадров) поверхности и математически определяет направление и величину перемещения.

Он содержит систему считывания изображения (Image Processor, IP), цифровой сигнальный процессор (Digital Signal Processor, DSP) и четырех выводной последовательный порт. На систему IP поступают микроскопические изображения поверхности посредством линзы и системы освещения. Эти изображения обрабатываются цифровым сигнальным процессором DSP для определения направления и расстояния движения. DSP вычисляет Δx и Δy - относительное смещение координат. Встроенный микроконтроллер считывает данные Δx и Δy и переводит данные в PS/2, USB, или RF сигналы перед отправкой их на компьютер или игровую консоль.

3 Снятие временных диаграмм на разъеме манипулятора (PS/2)

PS/2 — разъем, применяемый для подключения клавиатуры и мыши. Впервые появился в 1987 году на компьютерах IBM PS/2 и впоследствии получил признание других производителей и широкое распространение в персональных компьютерах и серверах рабочих групп. Скорость передачи данных — от 80 до 300 Кб/с и зависит от производительности подключенного устройства и программного драйвера.

PS/2-мышь использует интерфейс двунаправленный последовательный протокол для передачи движения и кнопки состояния данных на вспомогательный контроллер устройств компьютера (часть контроллера клавиатуры). Контроллер, в свою очередь, может отправить ряд команд мыши, чтобы установить отчет о скорости, разрешения, сбросить мышь, отключите мышь и т.д. Хост обеспечивает мышь с 5V ~ 100 мА питания.

Стандартный PS/2 мышь Интерфейс поддерживает следующие входы: X (вправо/влево) движение, Y (вверх/вниз) движения, левая кнопка, средняя кнопка, и правой кнопкой мыши. Мышь периодически читает эти входы и обновления различных счетчиков и флаги с учетом движения и состояния кнопки.

На рисунке 3.1 изображен разъем (Socket) PS/2. В таблице 3.1 приведена распиновка разъема.

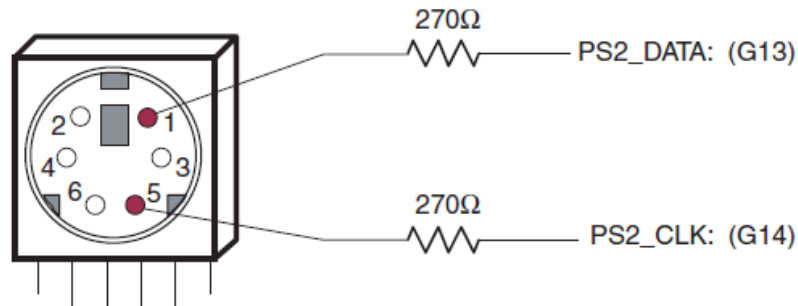


Рисунок 3.1 – Разъем PS/2

Таблица 3.1 – Обозначения контактов в разъеме PS/2

Номер контакта	Обозначение
1	Данные (PS2_DATA)
2	Не используется (резервировано)
3	Земля (GND)
4	Питание 5В (+5V)
5	CLK (PS2_CLK)
6	Не используется (резервировано)

Мышь генерирует синхроимпульсы и данные сигнала при перемещении, в противном случае эти сигналы остаются. Каждый раз, когда мышь перемещается, мыши отправляет три 11-разрядных слов хосту. Каждый из 11-битных слов содержит “0” стартовый бит, затем 8 бит данных, затем бит четности, и прекращается с “1” стоп-бит. Каждая передача данных содержит 33 общего бит, где биты 0, 11 и 22 “0” биты начала, и биты 10, 21, и 32 стоп-биты (рисунок 3.2, таблица 3.2).

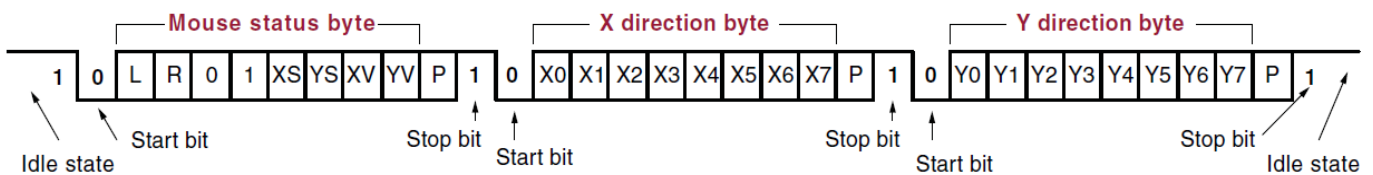
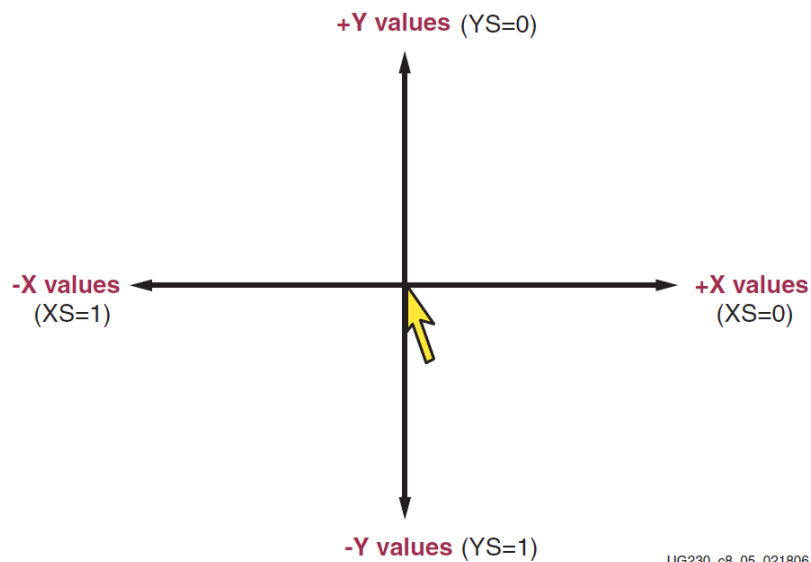


Рисунок 3.2 – Передача пакета PS/2 мыши к хосту

Таблица 3.2 – Стандартный набор передаваемого пакета от мышки к хосту (контроллеру PS/2 на компьютере)

	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
Байт 1	Y-переполнение	X-переполнение	Y знаковый бит	X знаковый бит	1	Средняя кнопка	Правая кнопка	Левая кнопка
Байт 2	X-движение							
Байт 3	Y-движение							

PS/2-мышь работает в относительной системе координат (рис. 3.3), в котором перемещая мышь вправо создает положительное значение в поле X, и двигаясь влево создает отрицательное значение. Кроме того, при перемещении мыши вверх создает положительное значение в поле Y, и перемещая его вниз представляет собой отрицательную величину. XS и YS биты в байт состояния определяют знак каждого значения, где “1” указывает на отрицательное значение.

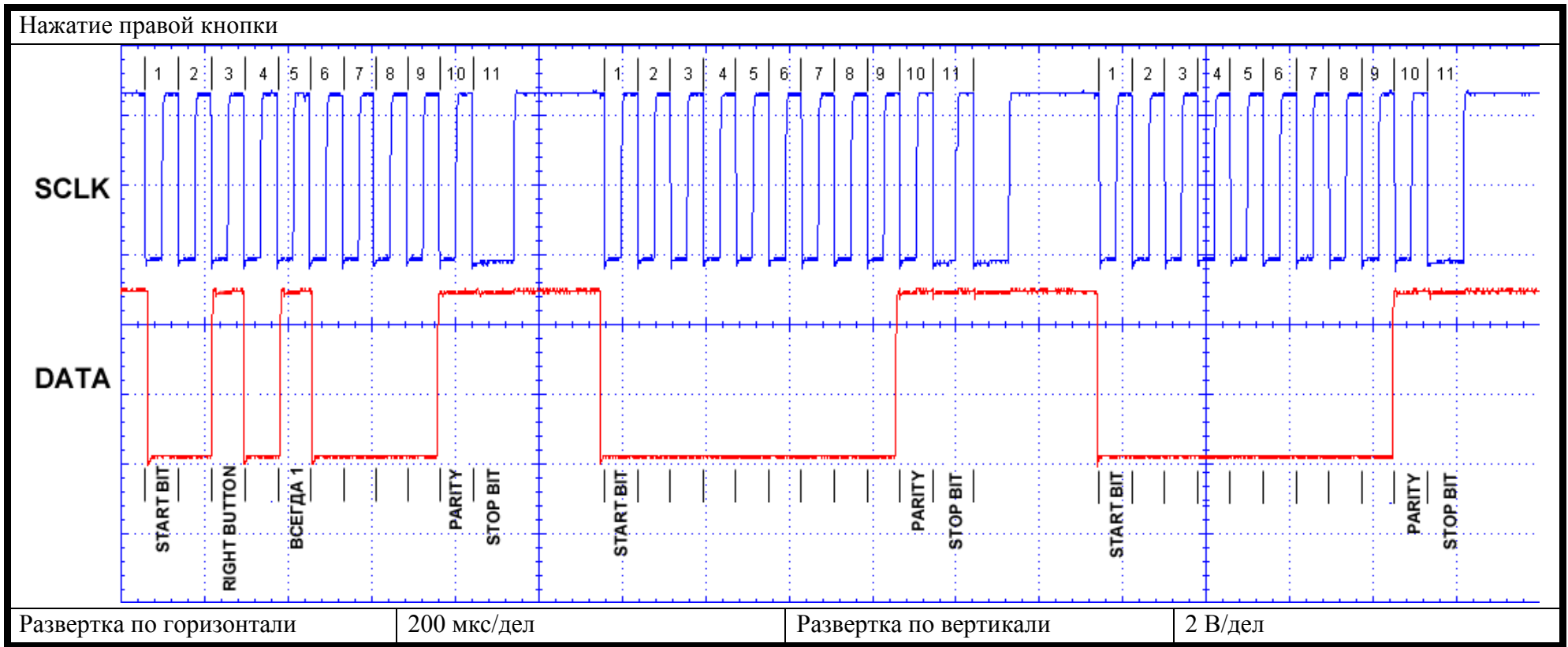


Величины X и Y значения представляют собой скорость движения мыши. Чем больше значение, тем быстрее движется мышь. XV и YV биты в байт состояния указать, когда X и Y значения превышают максимальные значения, условия переполнения. '1' указывает, когда происходит переполнение. Если мышь движется непрерывно, 33-битная передача повторять каждые 50 мс или около того.

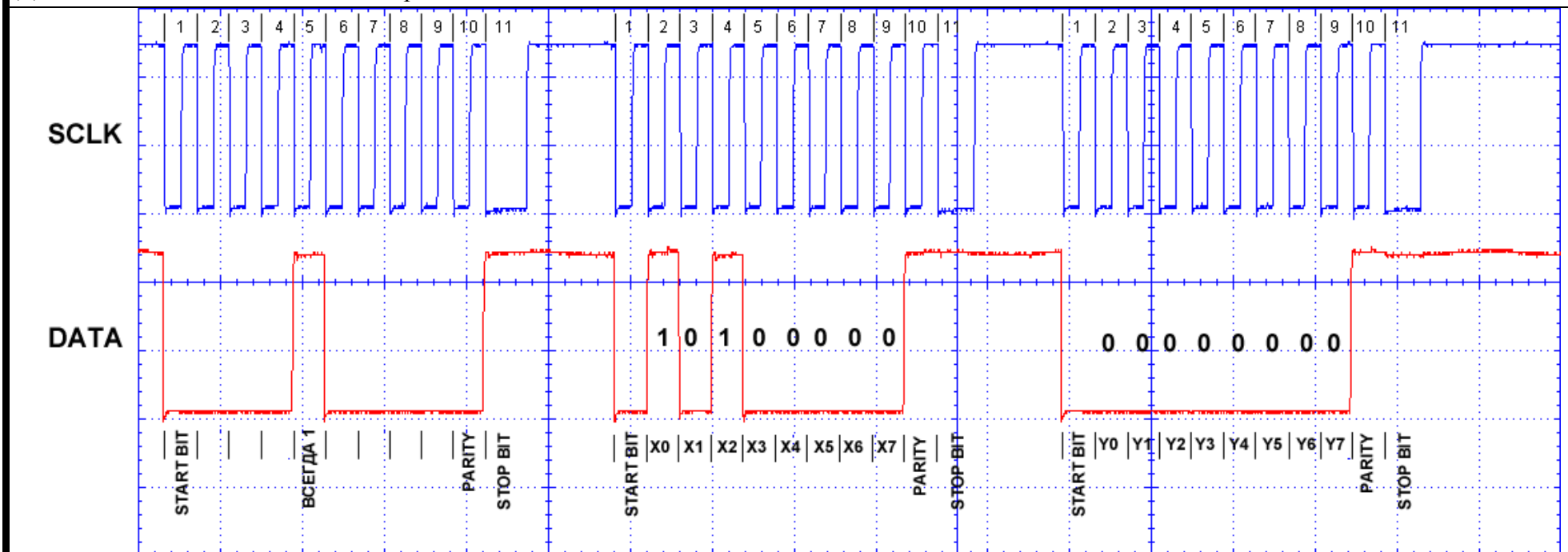
L, R и M поля в байте состояния указывают на нажатие левой, правой или средней кнопки. “1” означает, что кнопка мыши была нажата.

Рисунок 3.3 – Использование мыши относительной системы координат для отслеживания движения

Таблица 3.3 – Временные диаграммы на разъеме PS/2 (сигналы SCLK и DATA)



Движение мышки на 5 пикселей вправо



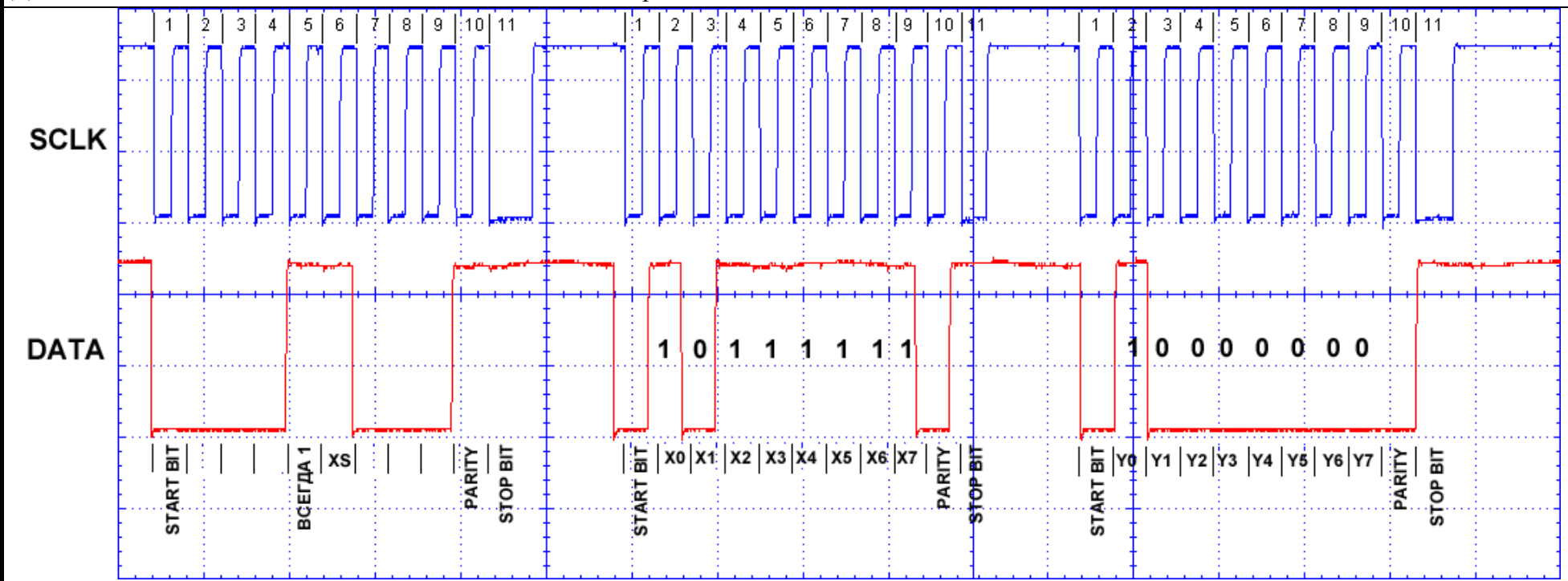
Развертка по горизонтали

200 мкс/дел

Развертка по вертикали

2 В/дел

Движение мышки на 253 пикселя влево и на 1 пиксел вверх



Развертка по горизонтали

200 мкс/дел

Развертка по вертикали

2 В/дел

Движения замерялись с помощью программы Screen Loupe Magnification Utility (рисунок 3.4).

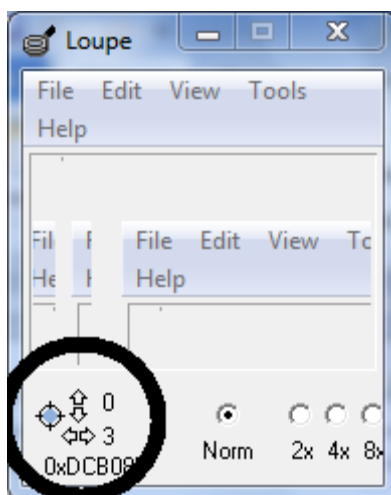


Рисунок 3.4 – Координаты мыши, отображаемые в программе Loupe

Далее приводится связь между компьютером и PS/2-мышью во время процесса загрузки. Инициализация мыши довольно типична, и при «эмуляции» работы мыши нужно реализовать следующую последовательность команд:

	Сторона	Команда	Комментарии
		Power-on Reset:	
1	Mouse:	AA	;самотестирование прошло
2	Mouse:	00	;id мыши
3	Host:	FF	;команда reset
4	Mouse:	FA	; подтверждение
5	Mouse:	AA	;самотестирование прошло
6	Mouse:	00	;id мыши
7	Host:	FF	;команда reset
8	Mouse:	FA	; подтверждение
9	Mouse:	AA	;самотестирование прошло
10	Mouse:	00	;id мыши
11	Host:	FF	;команда reset
12	Mouse:	FA	; подтверждение
13	Mouse:	AA	;самотестирование прошло
14	Mouse:	00	;id мыши
15	Host:	F3	;установка частоты дискретизации:
16	Mouse:	FA	; подтверждение :Режим прокрутки мыши
17	Host:	C8	; decimal 200 :
18	Mouse:	FA	; подтверждение :
19	Host:	F3	;установка частоты дискретизации:
20	Mouse:	FA	; подтверждение :
21	Host:	64	; decimal 100 :
22	Mouse:	FA	; подтверждение :
23	Host:	F3	;установка частоты дискретизации:
24	Mouse:	FA	; подтверждение :
25	Host:	50	; decimal 80 :
26	Mouse:	FA	; подтверждение :
27	Host:	F2	;считывание типа устройства :
28	Mouse:	FA	; подтверждение :
29	Mouse:	00	; id мыши :
30	Host:	F3	; установка частоты дискретизации:
31	Mouse:	FA	; подтверждение

32	Host:	0A	; decimal 10
33	Mouse:	FA	; подтверждение
34	Host:	F2	; считывание типа устройства
35	Mouse:	FA	; подтверждение
36	Mouse:	00	; id мыши
37	Host:	E8	; установка разрешения
38	Mouse:	FA	; подтверждение
39	Host:	03	; 8точек/мм
40	Mouse:	FA	; подтверждение
41	Host:	E6	; установить масштаб 1:1
42	Mouse:	FA	; подтверждение
43	Host:	F3	; установка частоты дискретизации
44	Mouse:	FA	; подтверждение
45	Host:	28	; decimal 40
46	Mouse:	FA	; подтверждение
47	Host:	F4	; включить
48	Mouse:	FA	; подтверждение
49		Initialization	; инициализация завершена.
		complete...	
		Если нажата (зажата) левая кнопка ...	
52	Mouse:	09 1 1 00001001	; бит0-нажатие левой кнопки, бит3-всегда 1
53	Mouse:	00 1 1	нет перемещения по X
54	Mouse:	00 1 1	нет перемещения по Y
		... и после отжатия кнопки	
57	Mouse:	08 1 1 00001000	
58	Mouse:	00 1 1	
59	Mouse:	00 1 1	

Особенности передачи по PS/2 расписаны в следующей главе.

4 «Особенности» передачи по PS/2

Есть два основных процесса: передача и получение. У простых UARTs есть отдельные провода для их последовательных данных, но устройства PS/2 совместно используют единственный провод сигнала данных. Таким образом, устройства не должны передавать, когда получатель занят, или получить, когда передатчик занят (рисунок 4.1).

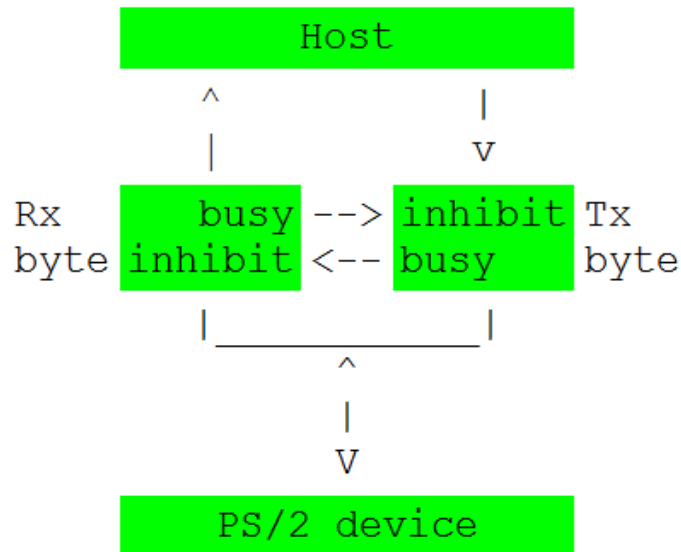


Рисунок 4.1 – Передача данных по PS/2, реализация 1

Диаграмма выше кажется хорошим на первый взгляд, работающий над байтом за основание байта. Во время передачи байта другой процесс запрещен, но для одного все еще возможно начаться между байтами пакета. Поэтому нужна более сложная модель (рисунок 4.2).

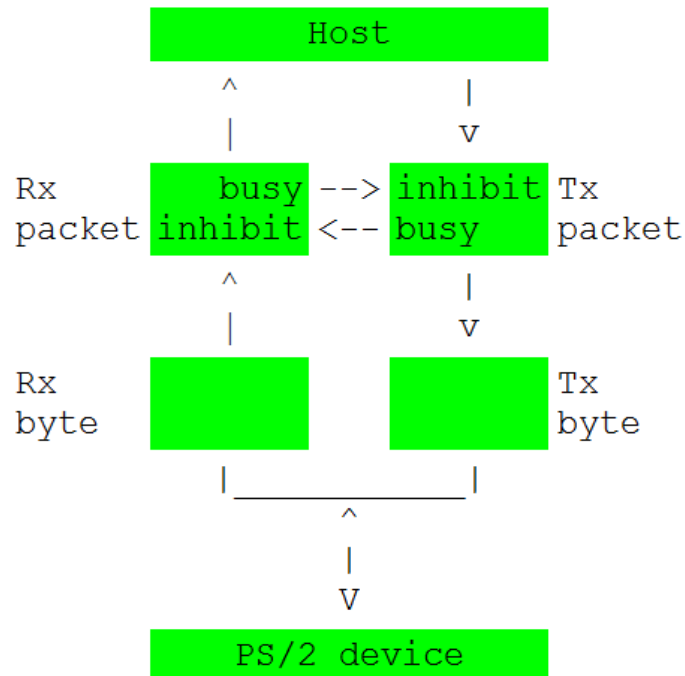


Рисунок 4.2 – Передача данных по PS/2, реализация 2

Пакет процессы тормозят друг друга на протяжении всего пакета. В этой связи возникает еще одна проблема: размер пакета может варьироваться. В случае с PS/2 мышью,

входящие пакеты имеют 3 или 4 байта. Исходящие пакеты для мышей и клавиатур может быть гораздо более сложным. Поэтому более сложные проблемы берет на себя оборудование. Так как комплекс пакета нужен только в программное обеспечение контролируемой окружающей среды, то имеет смысл иметь программное обеспечение контролирующее включение и отключение Tx/Rx процессов. Например, процессор может отключить прием перед отправкой пакета, а затем снова включить прием.

В дальнейшем уточнении, можно разделить PS/2 передатчика процесса между клавиатурой и мышью, так как процессор может справиться с опрашиванием обеих устройств одновременно. При обычном использовании, байты часто отправляется мышке или клавиатуре поэтому делиться передатчика не должна повлиять на производительность системы (рисунок 4.3).

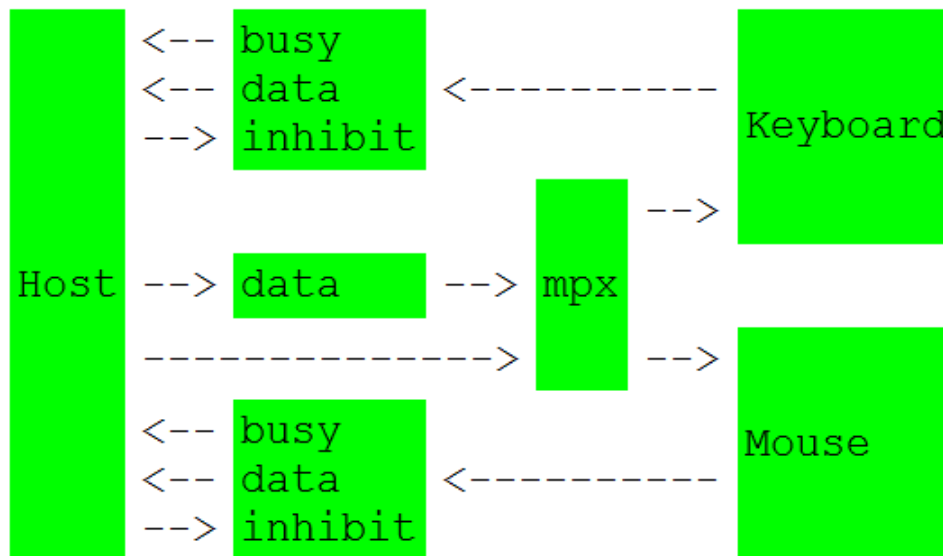


Рисунок 4.3 – Передача данных по PS/2, реализация 3

5 Программа на Ассемблере для работы с указателем

Для работы в MS-DOS используется прерывание 33h, позволяющее устанавливать драйвер мыши, узнавать координаты мыши, устанавливать обработчик и т.д. Рассмотрим некоторые операции, необходимые для работы с мышью.

Инициализация мыши. Прерывание 33h (далее везде подразумевается это прерывание), значение регистра ax=0000h.

Показать/спрятать курсор. ax=0001h/0002h.

Определить состояние мыши. Номер функции 0003h. Но данная функция используется редко, т.к. необходимо зацикливать программу и вызывать данную функцию. Чаще используется функция **Установить обработчик событий**. Подробная работа этой функции: Во-первых, полный набор параметров выглядит следующим образом:

- AX = 000Ch
- ES:DX = адрес обработчика
- CX = условие вызова
 - бит 0 - любое перемещение мыши
 - бит 1 - нажатие левой кнопки
 - бит 2 - отпускание левой кнопки
 - бит 3 - нажатие правой кнопки
 - бит 4 - отпускание правой кнопки
 - бит 5 - нажатие средней кнопки
 - бит 6 - отпускание средней кнопки
- CX = 0000h - отменить обработчик

Обработчик следует оформлять в виде дальней процедуры (то есть завершаться командой RETF). При вызове обработчика регистр CX содержит X-координату, а DX - Y-координату. Следует отметить, что координаты совпадают с координатами соответствующей точки на экране практически во всех графических режимах, кроме 04, 05, 0Dh, 13h. В них X-координату мыши нужно разделить на 2, для того чтобы получить номер столбца соответствующей точки на экране. В текстовых режимах обе координаты надо разделить на 8, чтобы получить номер строки и столбца соответственно.

В качестве программы демонстрации работы манипулятора приведена программа вывода точка в месте щелчка мышки, листинг которой предоставлена в таблице 5.1.

Компиляция проводилась в NASM (Netwide Assembler) (Рисунок 5.1).

Работа программы представлена на рисунке 5.2

Таблица 5.1 – Листинг программы вывода точки в месте щелчка мышки

```

org 100h
start:
    mov ax,12h ;инициализация гр. режима
    int 10h
    mov ax,0 ;ин. мыши
    int 33h
    mov ax,1 ; показать курсор
    int 33h

    mov ax,000ch ; Установка обработки щелчка
    mov cx,0002h
    mov dx, mouse_click
    int 33h

    mov ah,0 ; Ожидание ввода символа
    int 16h
    mov ax,000ch ; Отменить обработчик
    mov cx,0000h
    int 33h
    mov ax,3 ; возврат в текстовый режим
    int 10h
    ret

mouse_click:
    mov ax,2 ; спрятать курсор
    int 33h
    mov ah,0ch ; вывести точку
    mov al,0ah
    int 10h
    mov ax,1 ; показать курсор
    int 33h
    retf ;Обработчик событий должен быть оформлен как дальняя
процедура

```

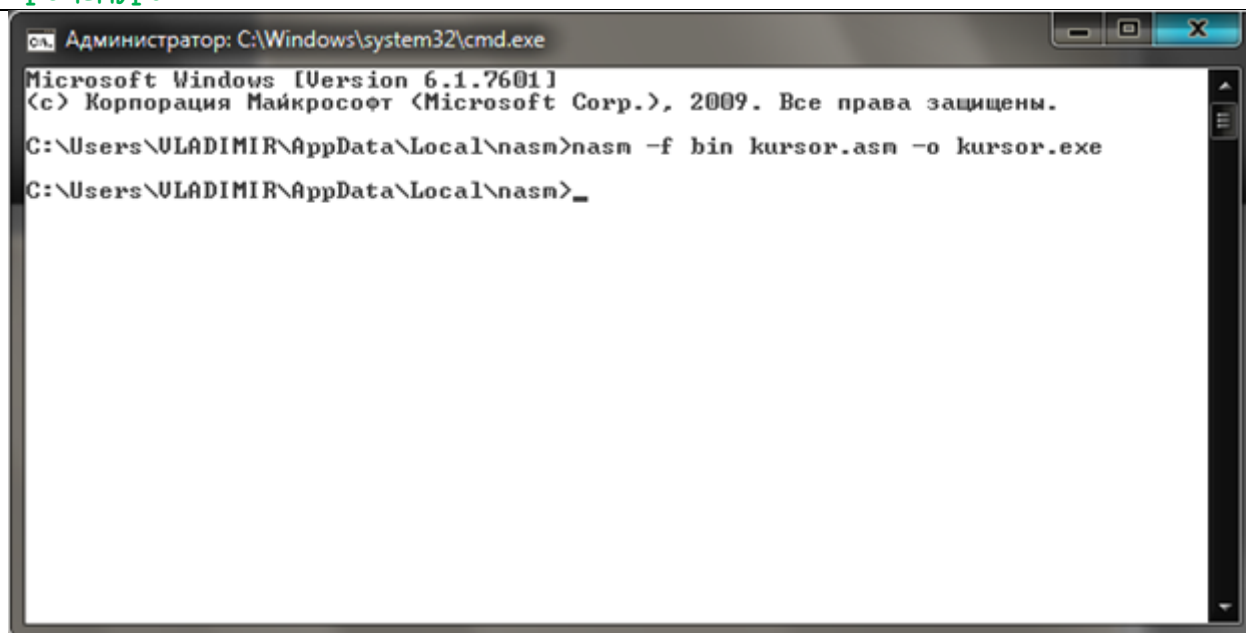


Рисунок 5.1 – Компиляция программы в NASM

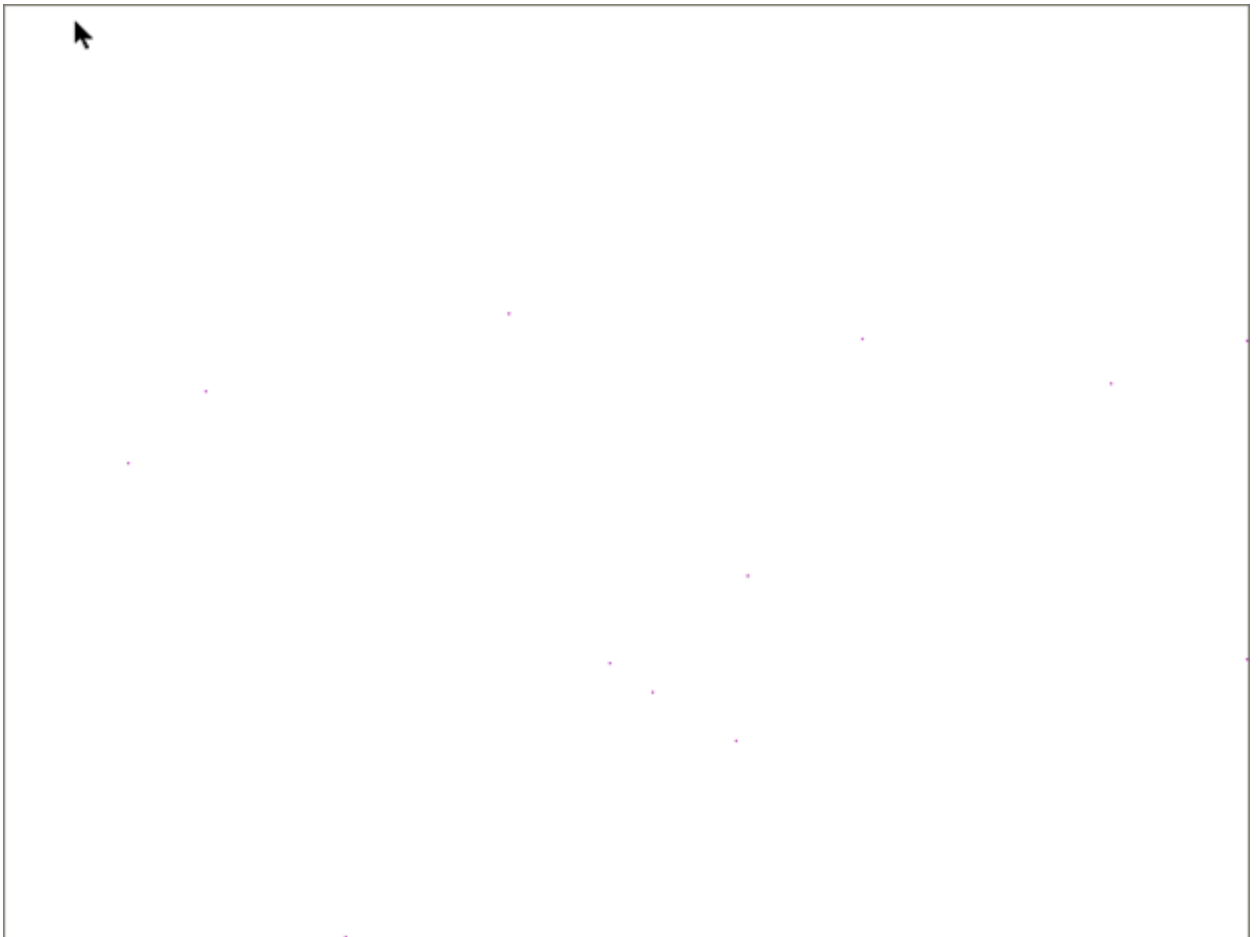
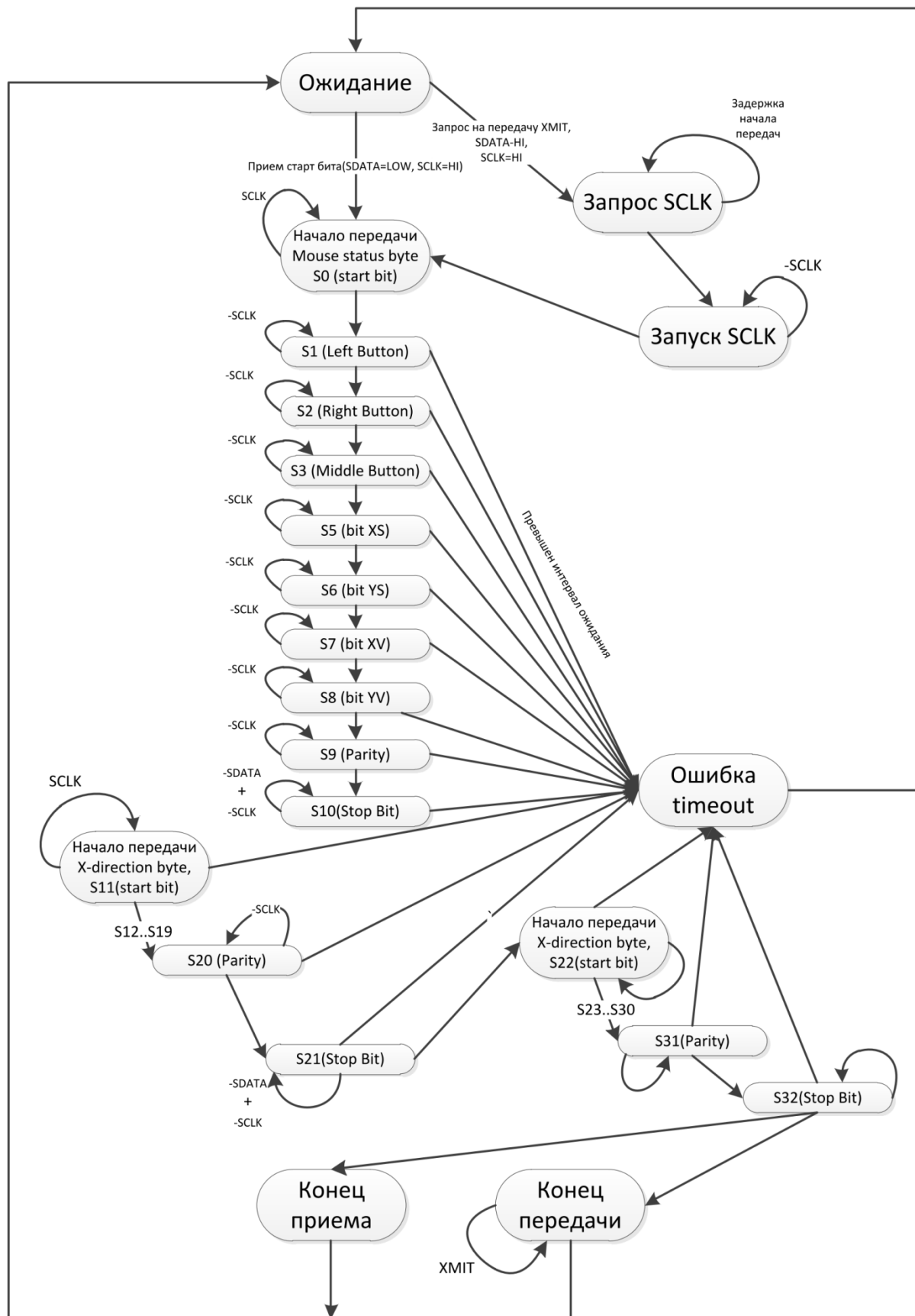


Рисунок 5.2 – Работа программы (для экономии краски приведена в негативе)

Список использованных источников

1. <http://www.ixbt.com/peripheral/mice-sensor.shtml>
2. http://licrym.org/index.php/Оптическая_мышь
3. http://o-d-v.nm.ru/optical_mouse/index.htm
4. http://www.freescale.com/files/microcontrollers/doc/ref_manual/DRM055.pdf
5. <http://www.computer-engineering.org/ps2protocol/>
6. <http://www.computer-engineering.org/ps2mouse/>
7. <http://members.casema.nl/hhaydn/howel/logic/mouse.htm>

Приложение 1 (конечный автомат)



Приложение 2 (алгоритм работы программы из 5 главы)

