



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. – 23 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 Logitech LS1.....	11
3 Диагностирование работы манипулятора	18
4 Снятие временных диаграмм на разъеме манипулятора (USB).....	21
5 Программа на Ассемблере для работы с указателем	26
Список использованных источников.....	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)

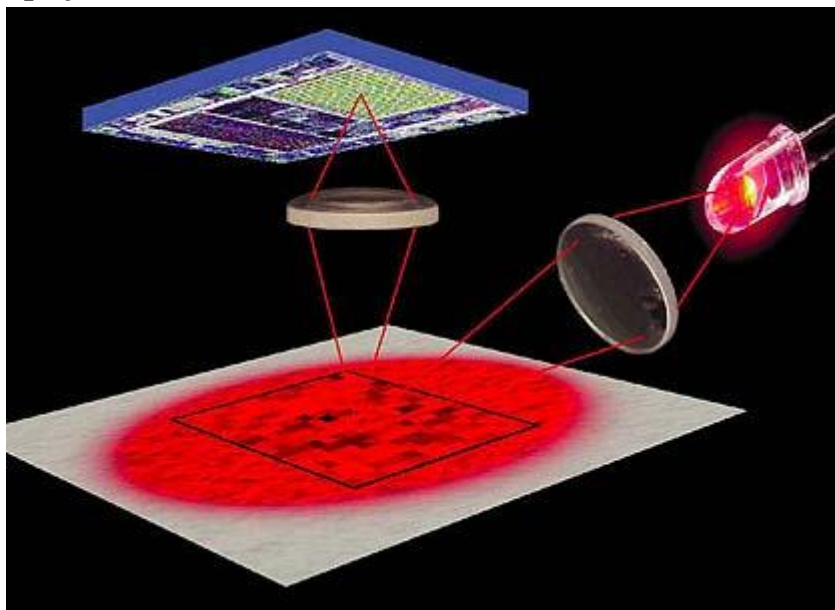


Рисунок 1.1 – Как «видят» компьютерные оптические мыши

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.

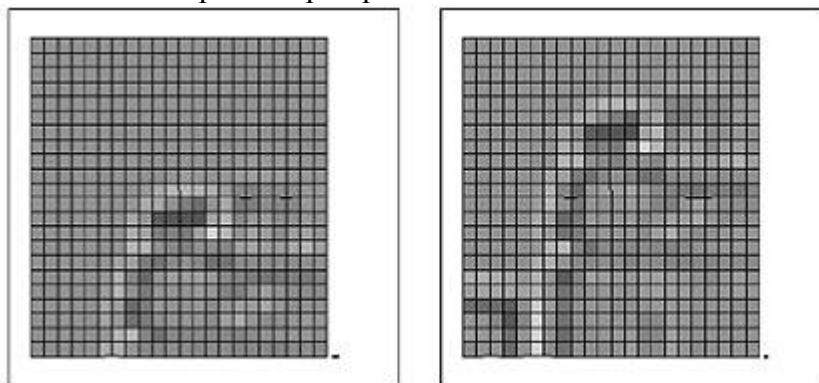
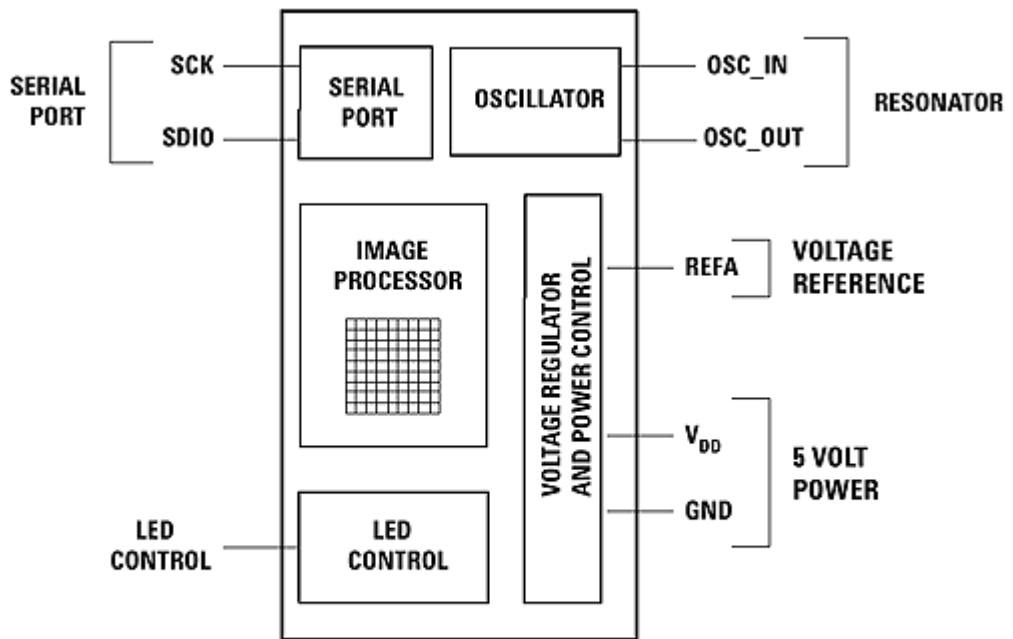


Рисунок 1.2 – 2 последовательных кадра

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 мышшь. Доходило и до того, что мышшь 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.

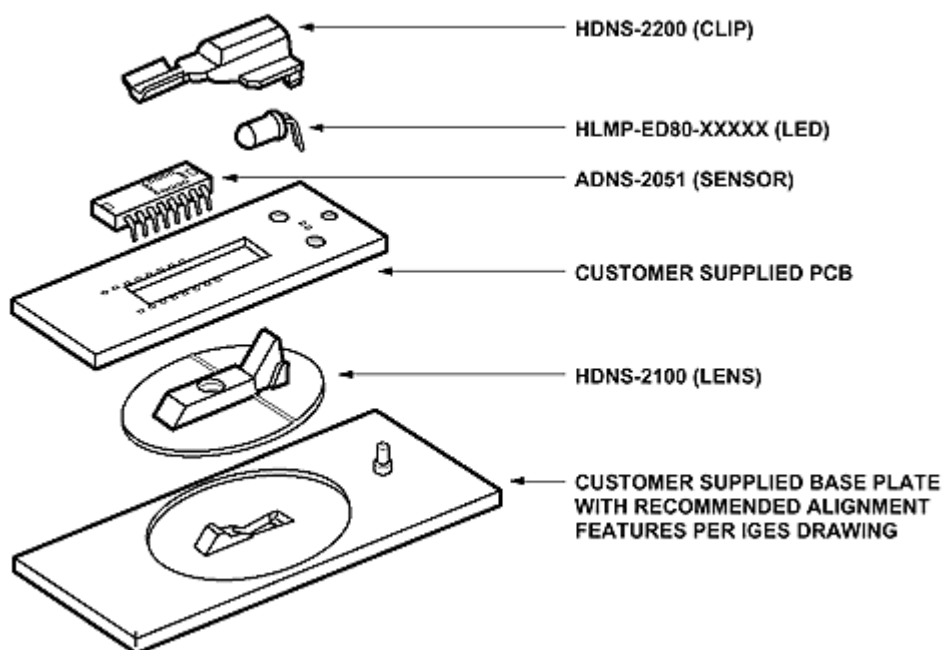
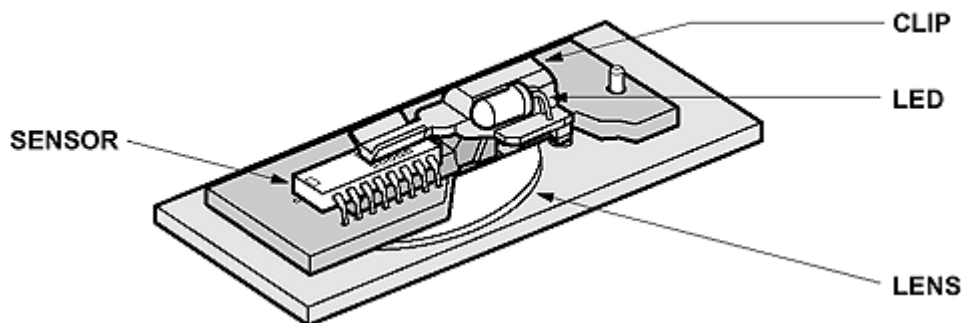
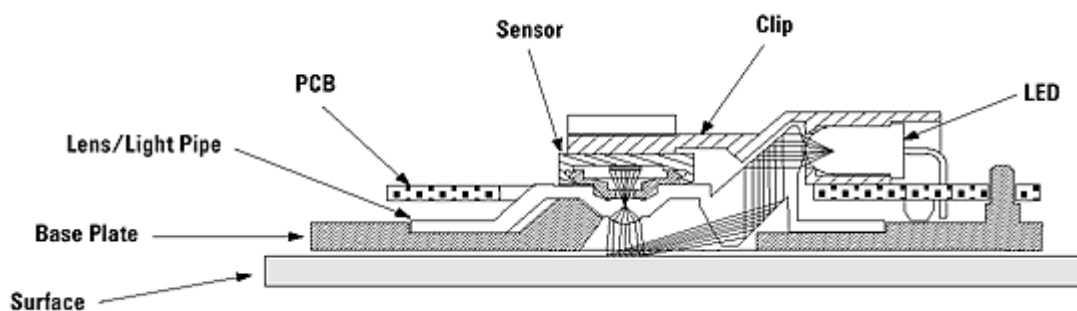


Рисунок 1.4 – Сборка элементов манипулятора на плате

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 Logitech LS1

2.1 Обзор платы

Данный манипулятор является лазерным. Лазерные манипуляторы обладают следующими преимуществами по сравнению с оптическими светодиодами:

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

Внешний вид платы манипулятора представлен на рисунке 2.1.1.

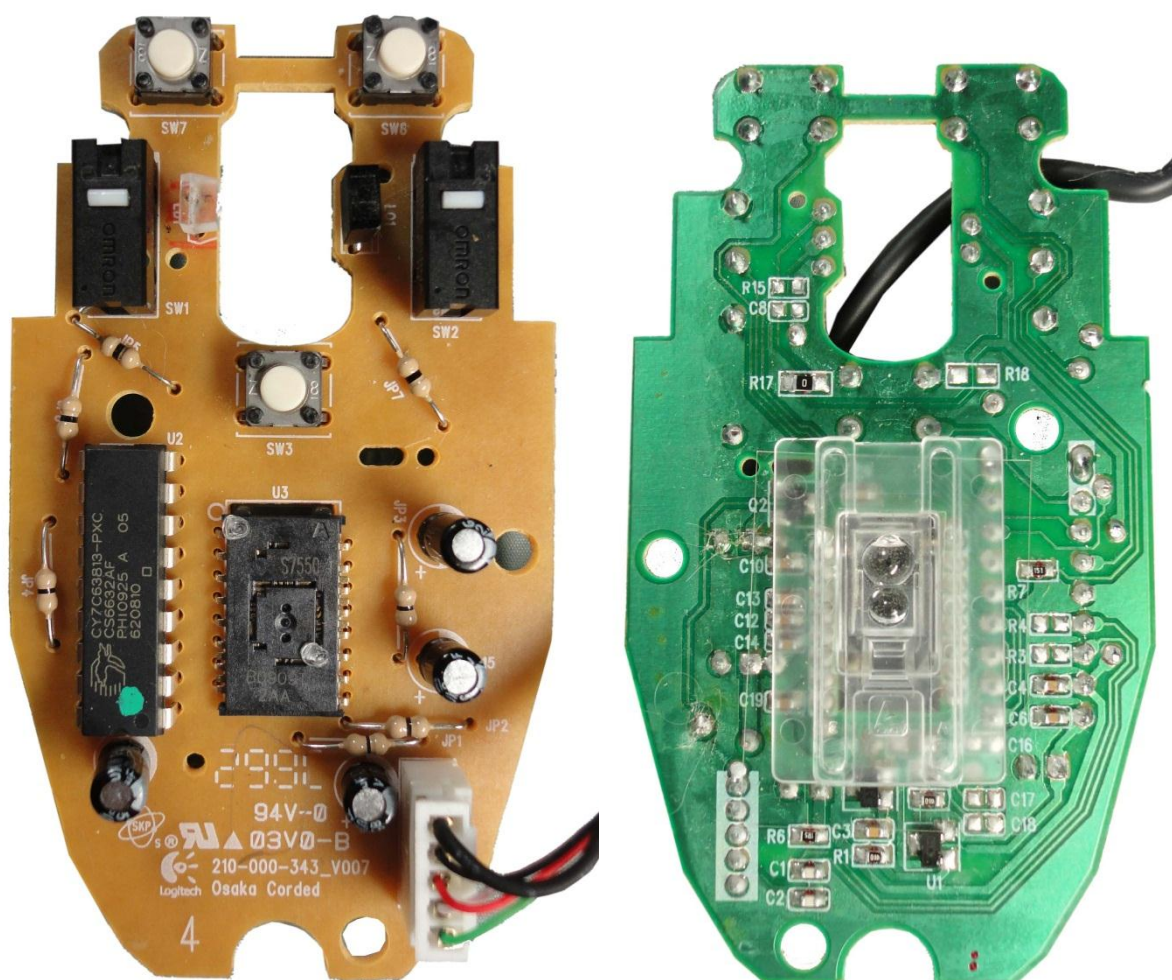


Рисунок 2.1.1 – Внешний вид платы манипулятора Logitech LS1

Схематическое изображение платы манипулятора Logitech LS1 представлено на рисунке 2.2.1.

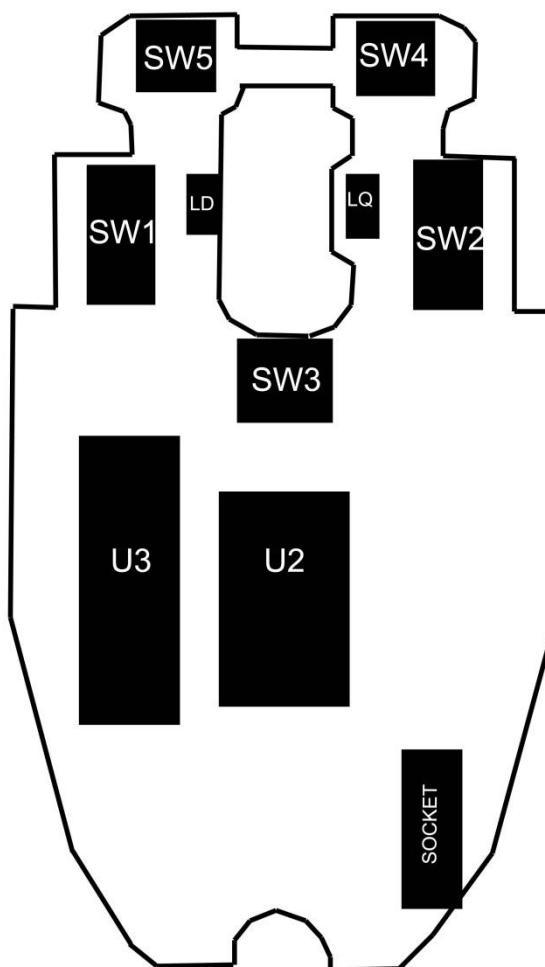


Рисунок 2.2.1 – Схематическое изображение платы манипулятора Logitech LS1

В таблице 2.1.1 представлены описание основных элементов платы.

Таблица 2.1.1 – Основные элементы

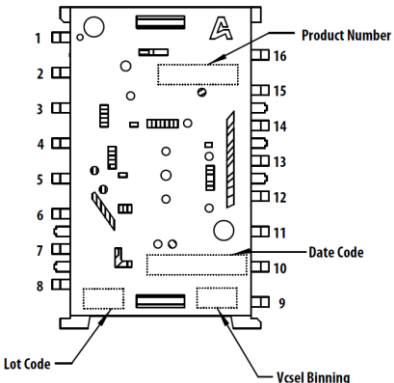
Обозначение	Предназначение
U2 (ADNS-7550)	Оптический датчик
U3 (CY7C63813)	«Низкоскоростной» периферийный контроллер USB (Low-Speed USB Peripheral Controller)
SW1	Кнопка левого клика
SW2	Кнопка правого клика
SW3	Кнопка для колеса прокрутки
SW4	Левая «боковая» кнопка колеса прокрутки
SW5	Правая «боковая» кнопка колеса прокрутки
LD, LQ	Светодиод и фотоприемник(инфракрасный) для регистрации прокручивания колеса прокрутки
SOCKET	Разъем для подключения к USB

2.1.1 Описание микросхем материнской платы

ADNS-7550

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

Таблица 2.1.1.2 – Разлиновка микросхемы ADNS-7550

Внешний вид	Наименование вывода	Назначение вывода
	1 VCSEL+VE	Positive Terminal of VCSEL
	2 LASER_NEN	LASER Enable (Active LOW)
	3 NCS	Chip select (active low input)
	4 MISO	Serial data output (Master In/Slave Out)
	5 SCLK	Serial clock input
	6 MOSI	Serial data input (Master Out/Slave In)
	7 MOTION	Motion Detect (active low output)
	8 XYLASER	XYLASER
	9 VDD5	5V Input
	10 NC	No Connection
	11 GND	Ground
	12 RefB	3V output
	13 RefA	1.8V regulator output
	14 DGND	Digital Ground
	15 VDDIO	IO Voltage input (1.65~3.6V)
	16 VCSEL-VE	Negative Terminal of VCSEL

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	832-865
Количество регистров данных движения (motion data registers)	12

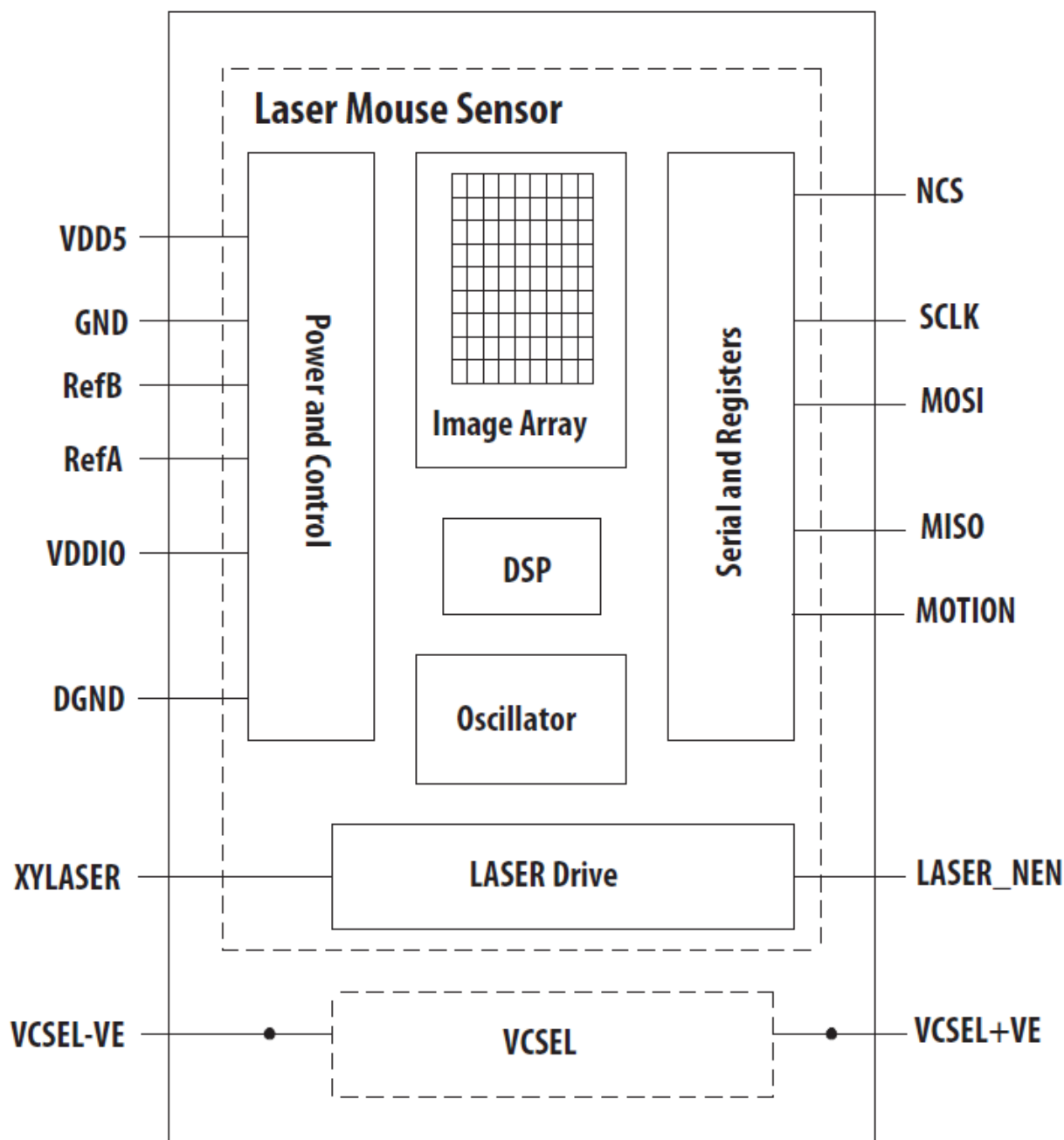


Рисунок 2.1.1.2 – Блок схема датчика

Работа датчика основана на технологии LaserStream, которая измеряет изменения в положении переданных последовательно изображений (кадров) поверхности и математически определяет направление и величину перемещения.

Он содержит систему считывания изображения (Image Acquisition System, IAS), цифровой сигнальный процессор (Digital Signal Processor, DSP) и четырех выводной последовательный порт. На систему IAS 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. Внешний микроконтроллер считывает данные Δx и Δy с последовательного порта датчика. Микроконтроллер затем переводит данные в PS2, USB, или RF сигналы перед отправкой их на компьютер или игровую консоль.

CY7C63813-PXC

Микросхема представляет собой периферийный контроллер USB. Цоколевка микросхемы представлена в таблице 2.1.1.1.

Таблица 2.1.1.1 – Разлиновка микросхемы CY7C63813-PXC

Внешний вид	Наименование вывода	Назначение вывода
	P1.3/SSEL	GPIO Port 1 bit 3 – Configured individually. Alternate function is SSEL signal of the SPI bus TTL voltage thresholds
	P1.4/SCLK	GPIO Port 1 bit 4 – Configured individually. Alternate function is SCLK signal of the SPI bus TTL voltage thresholds
	P1.5/SMOSI	GPIO Port 1 bit 5 – Configured individually. Alternate function is SMOSI signal of the SPI bus TTL voltage thresholds
	P1.6/SMISO	GPIO Port 1 bit 6 – Configured individually. Alternate function is SMISO signal of the SPI bus TTL voltage thresholds
	P1.7	GPIO Port 1 bit 7 – Configured individually
	P0.7	GPIO port 0 bit 7 – Configured individually. Not in 16 pin PDIP or SOIC package
	P0.6/TIO1	GPIO port 0 bit 6 – Configured individually. Alternate function Timer capture inputs or Timer output TIO1
	P0.5/TIO0	GPIO port 0 bit 5 – Configured individually. Alternate function Timer capture inputs or Timer output TIO0
	P0.4/INT2	GPIO port 0 bit 4 – Configured individually. Optional rising edge interrupt INT2
	P0.3/INT1	GPIO port 0 bit 3 – Configured individually. Optional rising edge interrupt INT1
	P0.2/INT0	GPIO port 0 bit 2 – Configured individually. Optional rising edge interrupt INT0
	P0.1 /CLKOUT	GPIO Port 0 bit 1– Configured individually. On CY7C639xx, optional clock out when external crystal oscillator is disabled or crystal output drive when external crystal oscillator is enabled. On CY7C638xx and CY7C63310, oscillator output when configured as Clock out.
	P0.0/CLKIN	GPIO Port 0 bit 0 – Configured individually. On CY7C639xx, optional Clock In when external crystal oscillator is disabled or crystal input when external crystal oscillator is enabled. On CY7C638xx and CY7C63310, oscillator input when configured as Clock In
	VSS	
	P1.0/SDATA/D+	GPIO Port 1 bit 0 / PS2 IO Data / USB D+
	P1.1/SCLK/ D-	GPIO Port 1 bit 1 / PS2 IO Clock / USB D-
	VDD	Power
P1.2/VREG	GPIO Port 1 bit 2 – Configured individually. 3.3V if regulator is enabled (add reference)	

Параметры микросхемы представлены в таблице 1.7

Таблица 1.7 – Параметры микросхемы CY7C63813-PXC

Параметр	Значение
Применение	Манипуляторы (мыши), Беспроводные ключи (Wireless Dongle)
Применение для разъемов ввода/вывода	USB, PS/2, SPI, GPIO
Архитектура памяти	Flash
Память (КВ)	8
Диапазон рабочих напряжений, В	4..5,5
Диапазон рабочих температур, °С	0..70

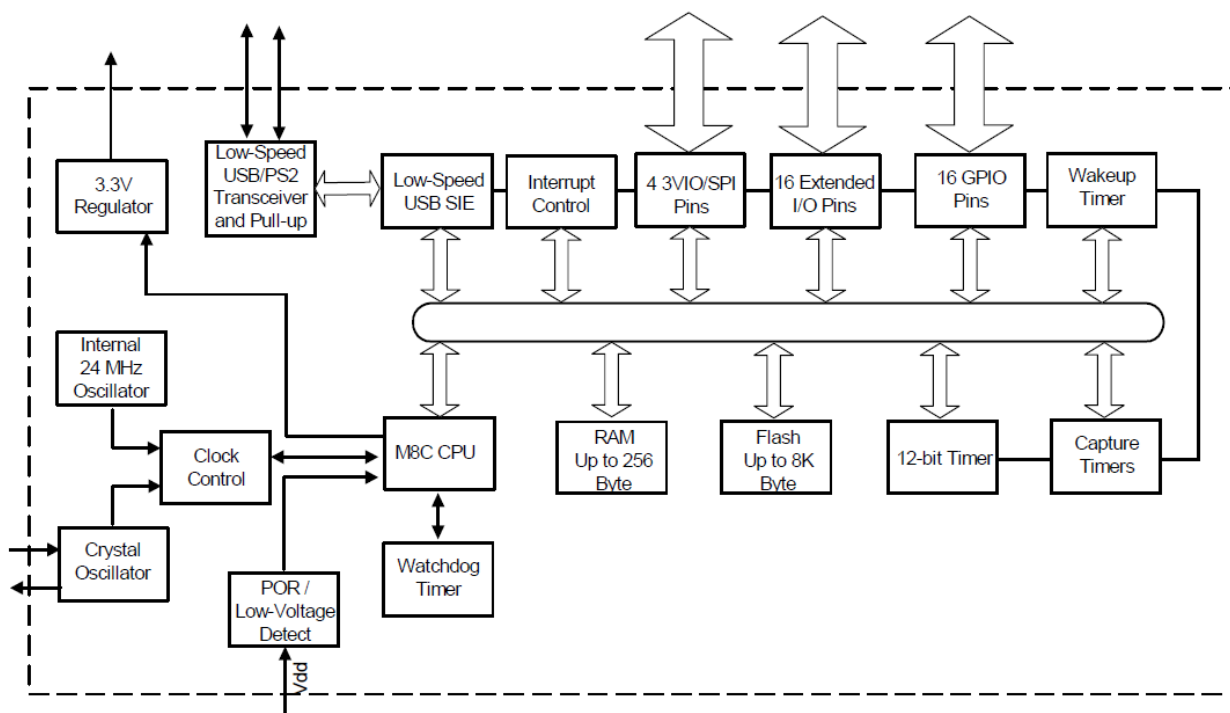


Рисунок 2.1.1.2 – Блок схема микросхемы CY7C63813-PXC

Контроллер, функции которого состоят в обеспечении работы кнопок, колёсика прокрутки, чтения информации о перемещении из датчика и работы по интерфейсу PS/2, USB или COM.

Контроллер является 8-битным флеш-программируемым микроконтроллером с интегрированным низко скоростным USB интерфейсом. Контроллер имеет 18 основных I/O (GPIO) выводов, поддерживающих USB, PS/2 и др. Выводы I/O сгруппированы в 5 портов (порты 0 - 4). Каждый вывод Порта0 и Порта1 могут быть настроен индивидуально в то время как выходы Портов 2, 3 и 4 могут быть настроены только в группе. Каждый GPIO порт поддерживает высоко импедансное состояние, с открытым стоком, КМОП/ТТЛ входы и КМОП выходы с 5 выводами, которые поддерживают программируемый драйвер с величиной тока стока до 50мА. Выводы GPIO Порта1 поддерживают уровень напряжения в 3,3В. В добавок, каждый вывод I/O может быть использован для формирования GPIO сигнала прерывания с исключением GPIO Порта 0. GPIO Порта 0 имеет 3 установленных вывода, которые имеют независимые сигналы прерывания (P0.2 - P0.4). Контроллер характеризуется внешним генератором. При наличии USB-связи внешний генератор может быть установлен на точную настройку временных требований

к USB (24 MHz \pm 1.5%). По желанию внешний кристалл на 12- 24MHz может быть использован для обеспечения высоко точных характеристик в USB-операциях. Тактовый генератор обеспечивает тактирование сигнала с частотой 12- 24MHz, который посылается внутрь микроконтроллера.

Контроллер может выделять до 8 Кбайт на флеш-памяти для пользовательского кода и до 256 байт RAM памяти для пространства стека и пользовательских переменных.

В добавок Контроллер имеет низковольтную логику сброса, Watchdog timer, контроллер сигналов прерывания, 16-битный Free-Running Timer и Capture Timer (захвата). Низковольтная логика сброса (LVR) определяет когда питание подаётся на устройство, сбрасывается логика на известное состояние и начинается выполнение инструкций с адреса 0x0000 на флеш-памяти.

Микроконтроллер поддерживает 23 маскируемых прерываний в контроллере сигналов прерывания. В систему прерывания входят шина прерывания USB, LVR/POR, программируемый таймер интервалов, 1.024-мс выход из Free Running Timer, 3 конечные точки USB, 2 таймера захвата, 5 GPIO Портов, 3 GPIO выводов, 2 SPI, 16-битный free running timer wrap, внутренний wake-up таймер и шина активного прерывания. Wake-up таймер вызывает периодические прерывания при включении. Конечный точки USB посылают прерывания после завершения USB-транзакции на шине. Таймер захвата производит прерывание всякий раз, когда новое значение таймера сохраняется благодаря выбранному GPIO событию.

free-running 16-bit timer обеспечивает 2 источника прерывания: программируемый таймер интервалов с разрешением в 1 мс и 1.024 мс выходами. Таймер может быть использован для измерения продолжительности события под контролем прошивки посредством чтения времени начала и окончания события, затем вычисления разницы между 2 значениями. 2 8-битных таймера захвата сохраняют программируемый 8-битный уровень free-running timer, когда GPIO подаются на 2 выхода захвата (P0.0, P0.1). 2 8-битных захватов могут быть соединены в один 16-битный.

Контроллер включает в себя также интегрированный интерфейс-двигатель USB (USB serial interface engine, SIE), который позволяет чипу легко связываться с другим устройством. Аппаратные средства поддерживают адресное USB-устройство с 3 конечными точками.

USB D+ и D- выводы могут быть альтернативно использованы как PS/2 SCLK и SDATA сигналы, т.л. устройство может быть спроектировано для отклика на подобные USB или PS/2 режимы операций. PS/2 операции поддерживаются внутренними подстроечными резисторами на SCLK и SDATA и прерывания сигналов начала PS/2 действий.

Контроллер поддерживает в программируемую систему, используя D+ и D- выводы как серийный программируемый режим интерфейса. Программируемый протокол не является USB.

3 Диагностирование работы манипулятора

Диагностирование манипулятора проводилось при помощи программы Troubleshooter V7.02 через загрузочную дискету. Экран загрузки программы представлен на рисунке 3.1. Результаты диагностирования представлены на рисунках 3.2-3.8.



Рисунок 3.1 – Программа Troubleshooter V7.02



Рисунок 3.2 – Главное меню программы Troubleshooter V7.02

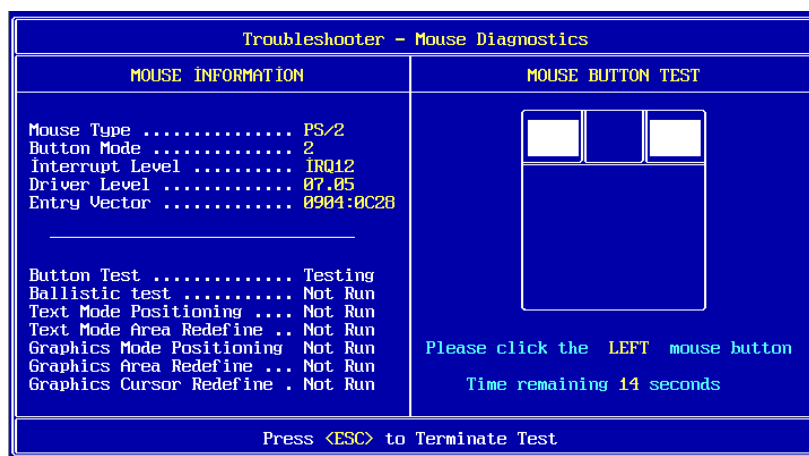


Рисунок 3.3 – Тестирование кнопок мыши

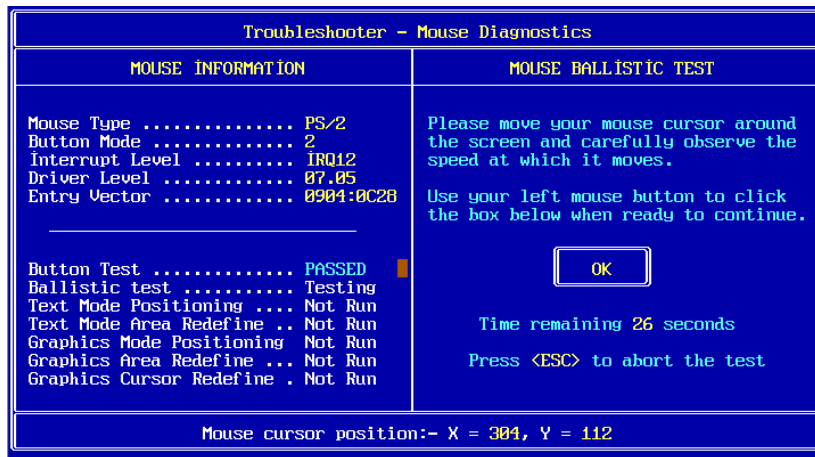


Рисунок 3.4 – Тестирование нажатия кнопки

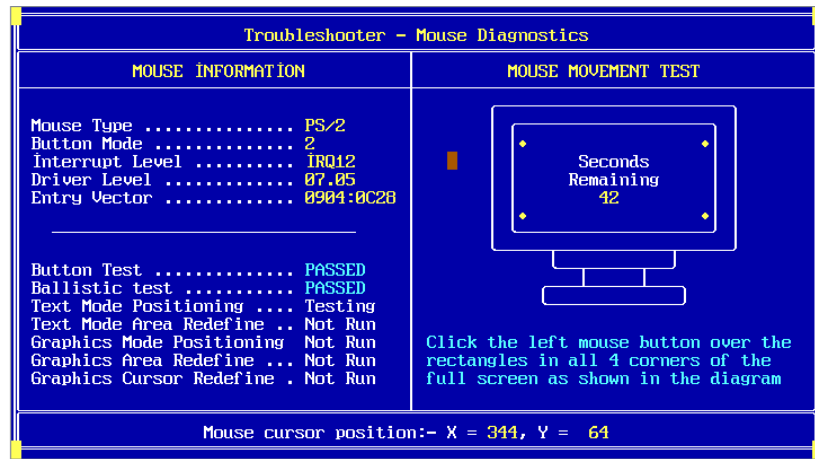


Рисунок 3.5 – Тестирование движения курсора

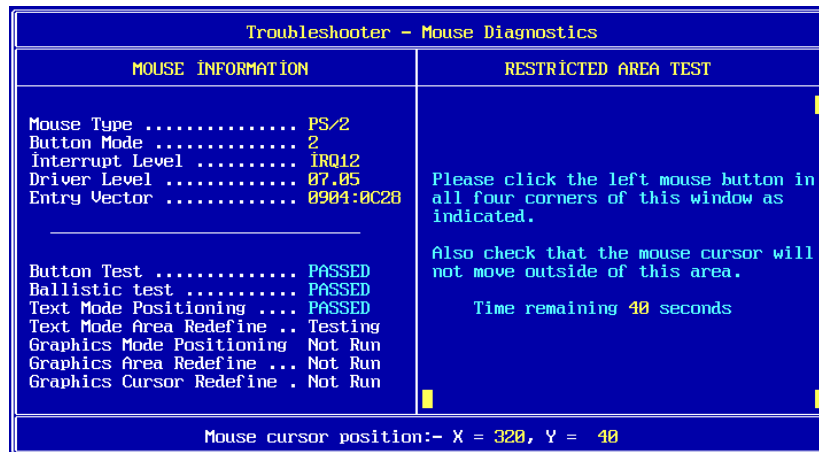


Рисунок 3.6 – Тестирование ограниченной области

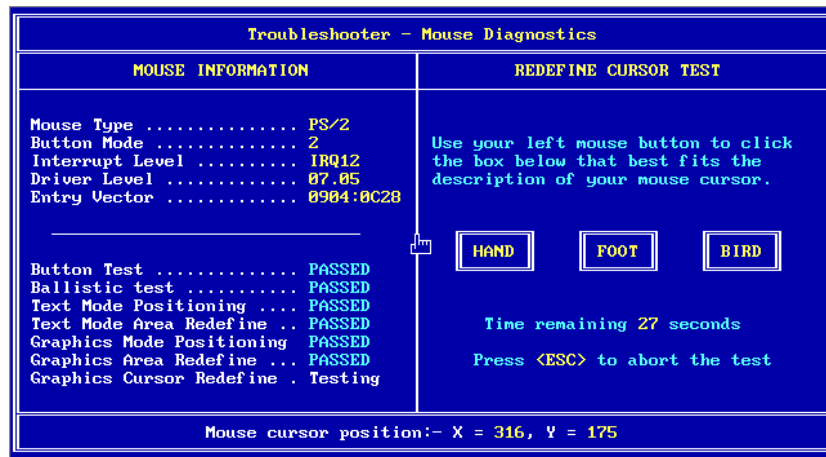


Рисунок 3.7 – Тестирование курсора (другое изображение курсора)

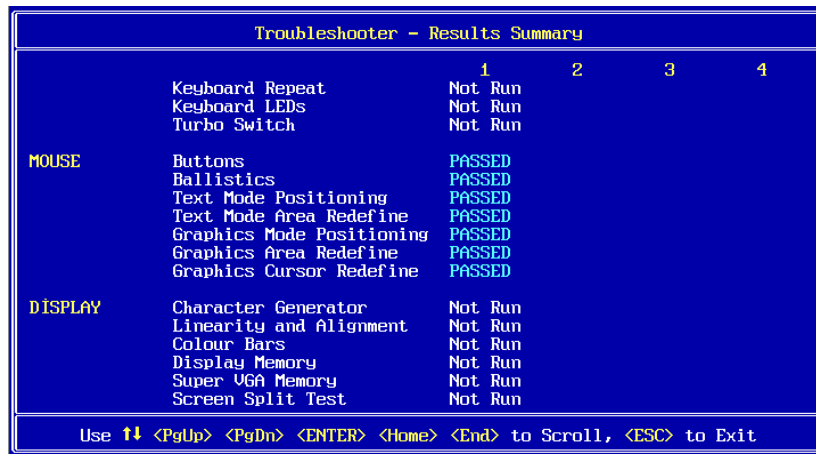


Рисунок 3.8 – Результаты тестирования

4 Снятие временных диаграмм на разъеме манипулятора (USB)

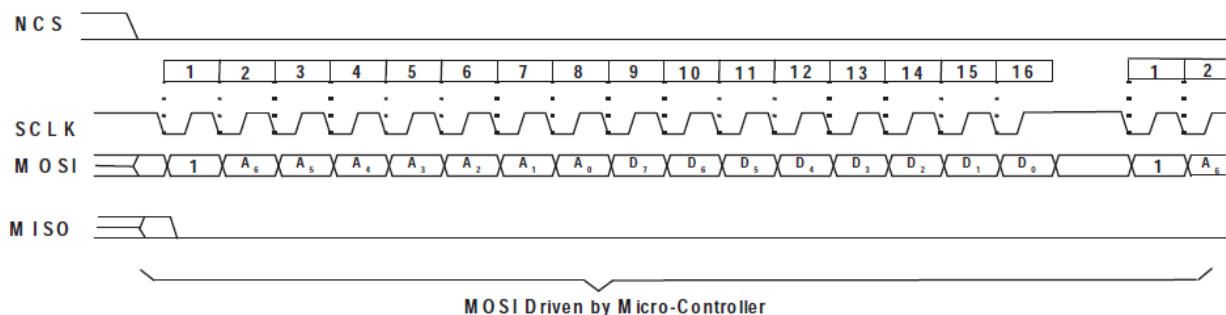


Рисунок 4.1 – Временные диаграммы с датчика мыши (операция записи)

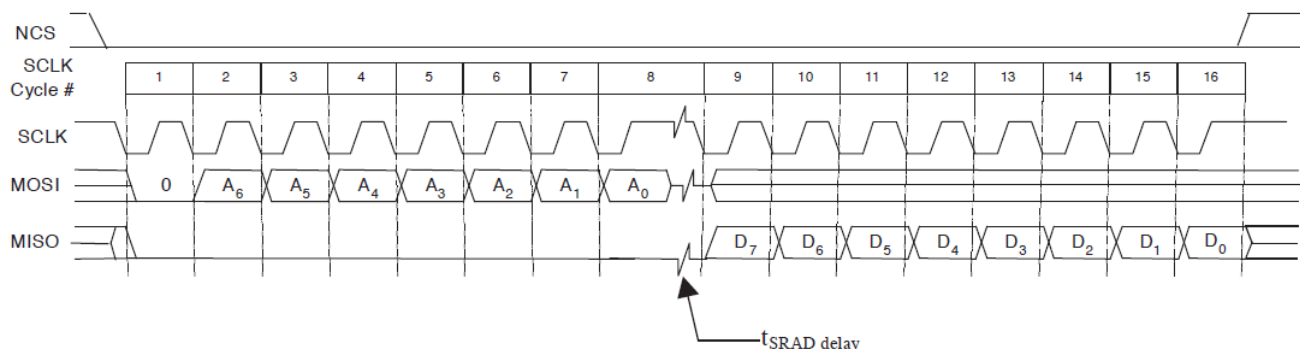


Рисунок 4.2 – Временные диаграммы с датчика мыши (операция чтения)

При операции чтения, данные собираются от ADNS-7550 для микроконтроллера. Эти данные всегда инициируется микроконтроллером и состоит из двух байтов. Первый байт содержит адрес, который посылает микроконтроллеру MOSI. Второй байт содержит данные и приводится в движение ADNS-7550 по MISO (Рисунок 4.1).

Были сняты временные диаграммы с SCLK, MOSI, MISO, NSC. Также временные диаграммы сняты с разъема USB. В таблице 4.1 приведены обозначения контактов. Результаты показаны в таблицах 4.2-4.3

Таблица 4.1 – Обозначения контактов в разъеме USB

Номер контакта	Обозначение	Цвет провода
1	VBUS	Красный
2	D-	Белый
3	D+	Зеленый
4	GND	Черный

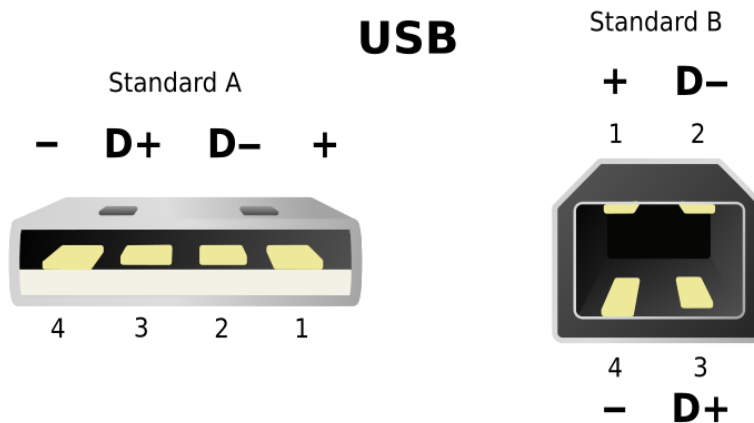


Рисунок 4.2 – Размещение проводников в разъемах USB типа А и В

В USB информация по каналу передается в виде *пакетов* (рис. 4.3). Каждый пакет начинается с *поля синхронизации SYNC*, за которым следует *идентификатор пакета PID*. Поле *check* представляет собой побитовую инверсию PID.

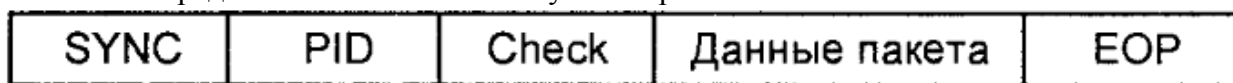


Рисунок 4.3 – Структура пакета

На рисунках 4.4 и 4.5 представлены структуры пакетов при работе манипулятора, а также диаграммы стартовых областей. Т.к. области данных и контрольной проверки трудно измерить на осциллографе, то в рамках данной работы было снята только область старта.

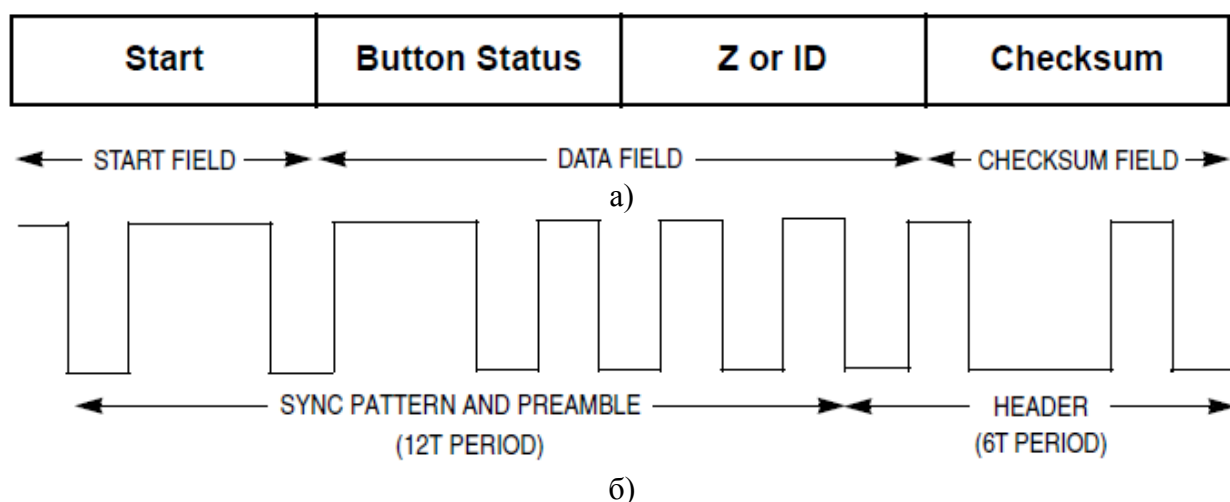


Рисунок 4.4 – а - структура пакета для нажатия кнопок и перемещения по вертикали манипулятора; б - диаграмма области «старт» пакета

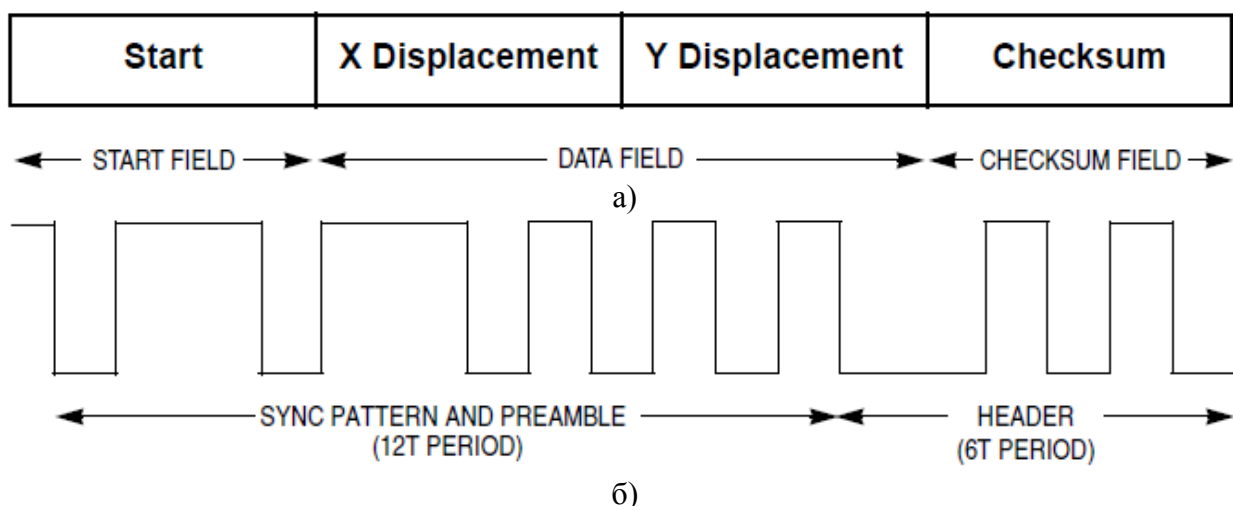
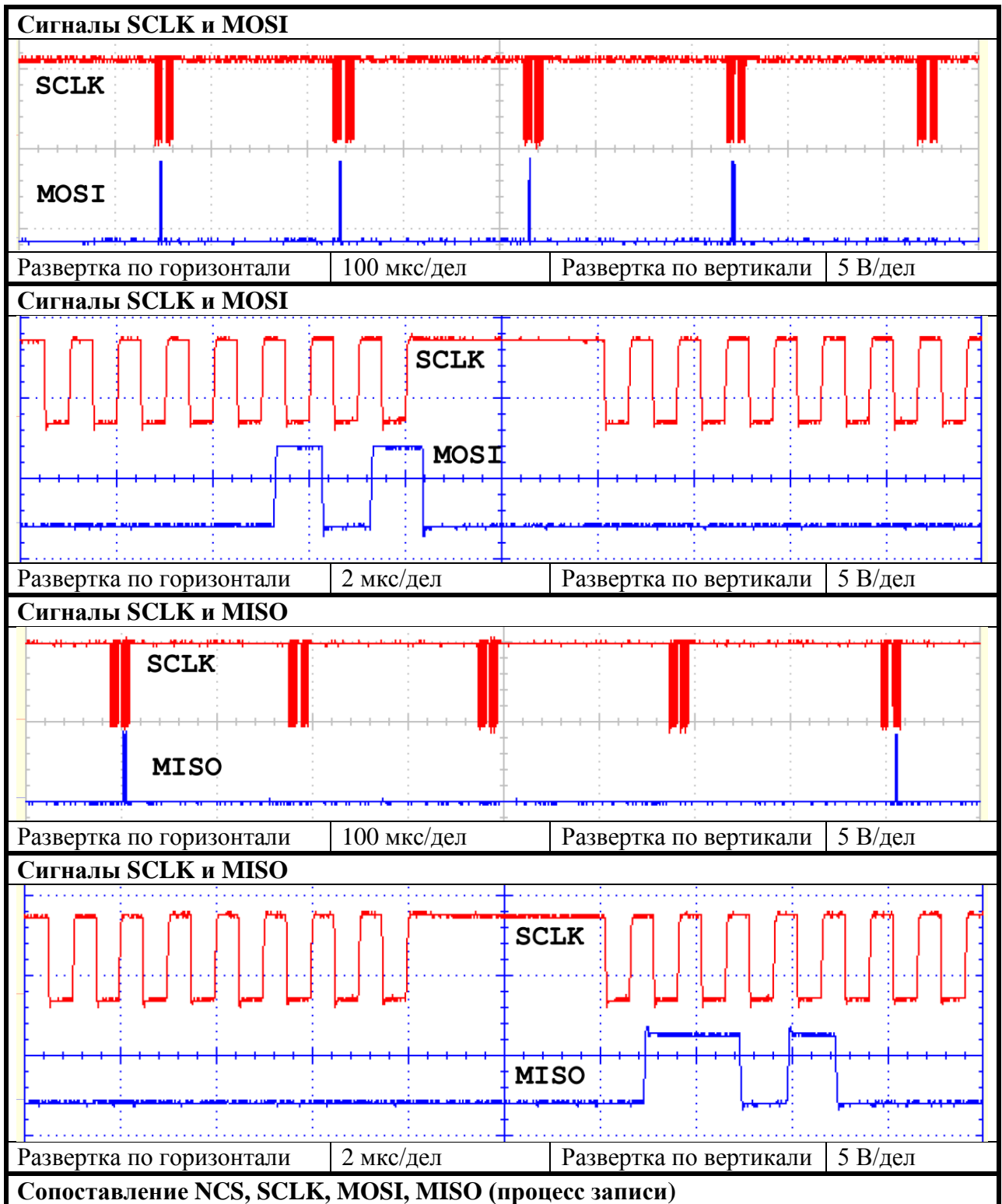
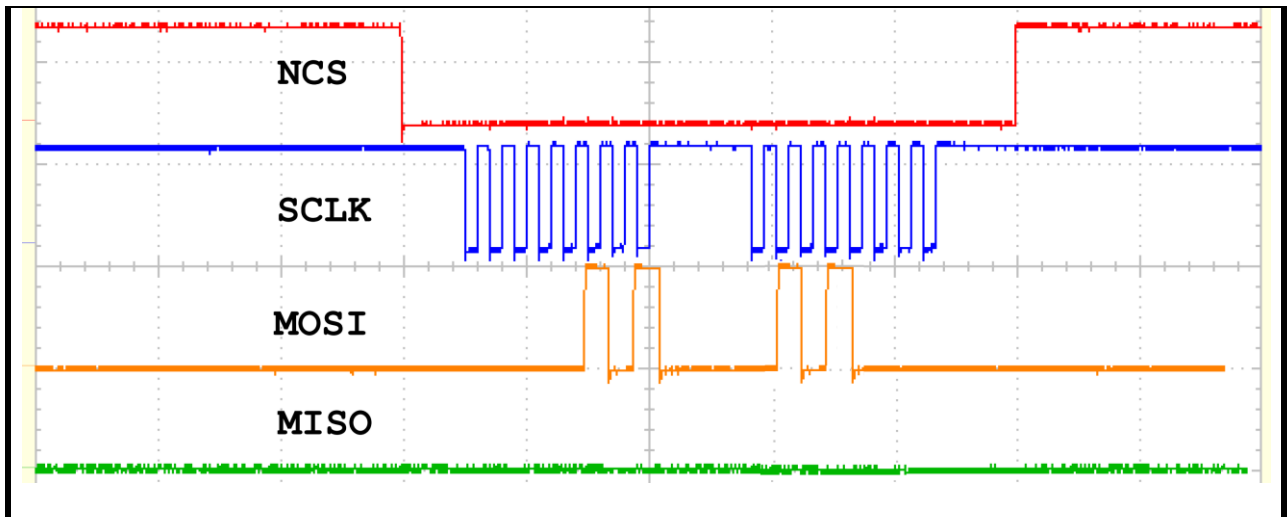


Рисунок 4.5 – а - структура пакета для перемещения по плоскости манипулятора; б - диаграмма области «старт» пакета

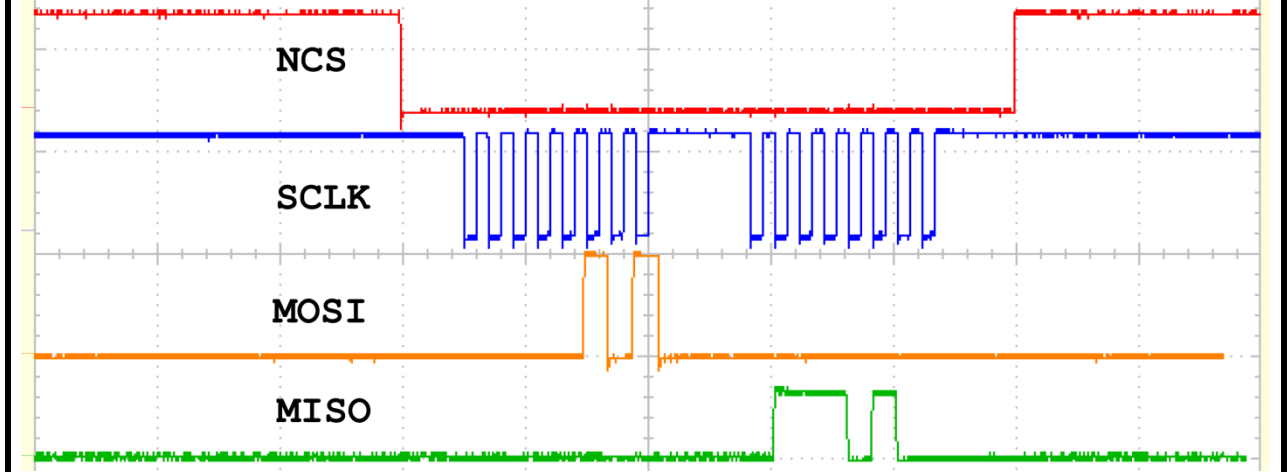
Таблица 4.2 – Временные диаграммы на оптическом датчике





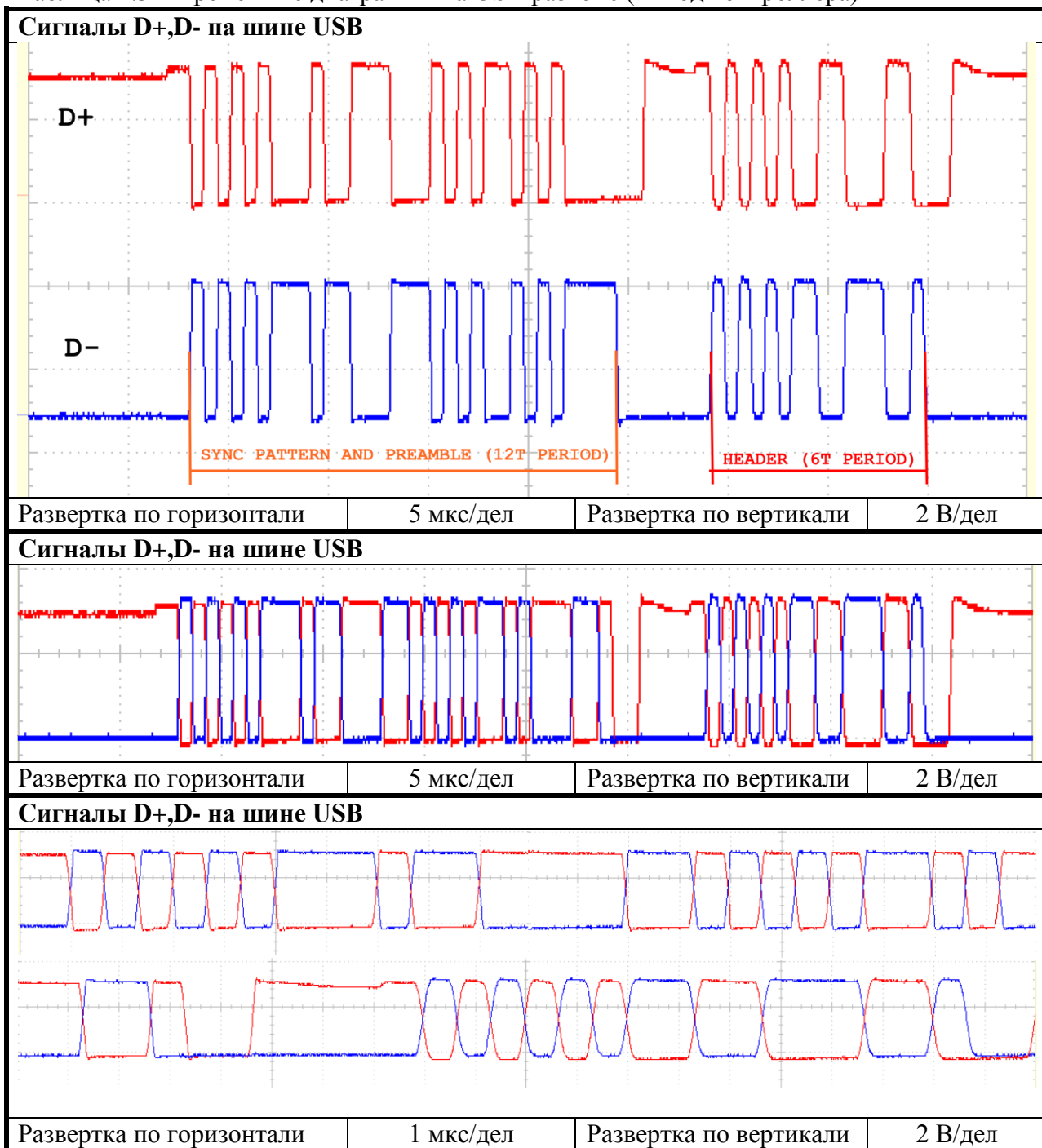
Развертка по горизонтали	5 мкс/дел	Развертка по вертикали	5 В/дел
--------------------------	-----------	------------------------	---------

Сопоставление NCS, SCLK, MOSI, MISO (процесс чтения)



Развертка по горизонтали	5 мкс/дел	Развертка по вертикали	5 В/дел
--------------------------	-----------	------------------------	---------

Таблица 4.3 – Временные диаграммы на USB разъеме (выход контроллера)



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 заикливать 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

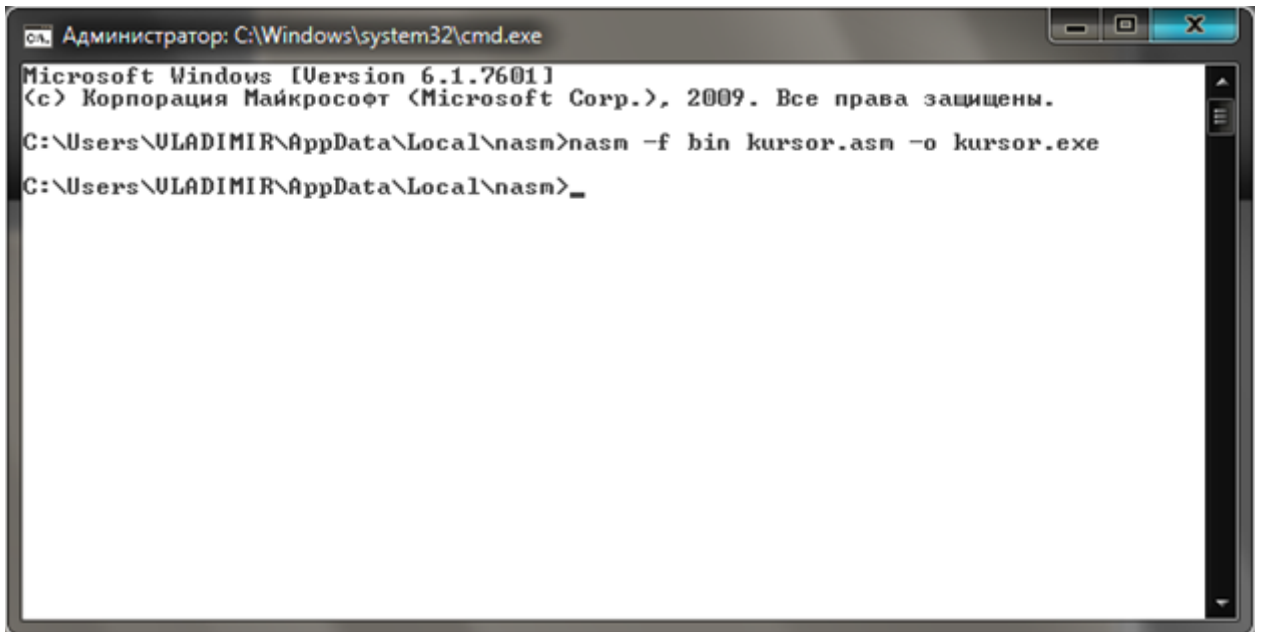
Таблица 5.1 – Листинг программы вывода точки в месте щелчка мышки

```
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
```



```
Администратор: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
(c) Корпорация Майкрософт (Microsoft Corp.), 2009. Все права защищены.
C:\Users\VLADIMIR\AppData\Local\nasm>nasm -f bin cursor.asm -o cursor.exe
C:\Users\VLADIMIR\AppData\Local\nasm>_
```

Рисунок 5.1 – Компиляция программы в NASM

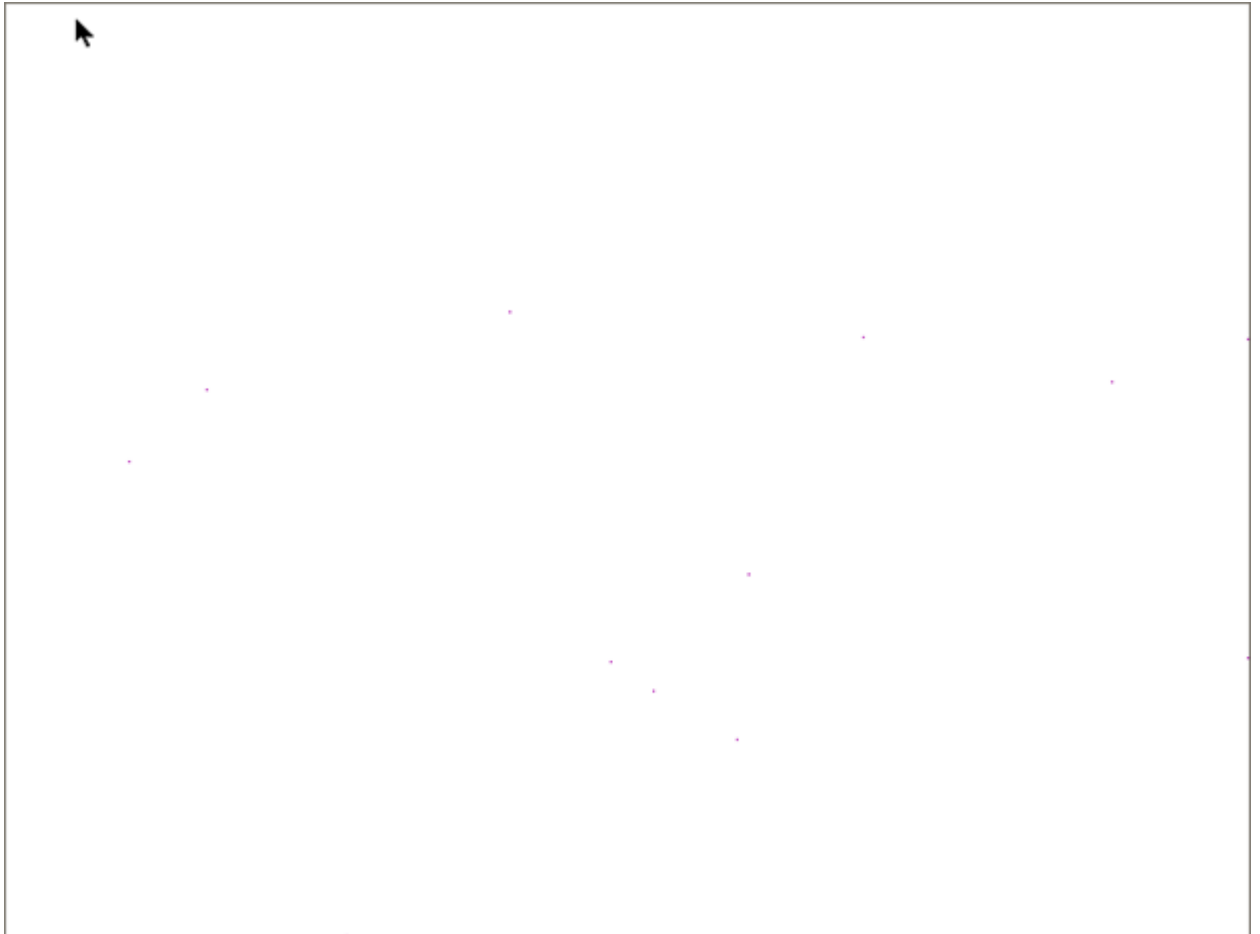


Рисунок 5.2 – Работа программы

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. Агуров В.П. Интерфейсы USB. Практика использования и программирования. – СПб.: БХВ-Петербург, 2004. – 576 с.: ил.
5. http://www.freescale.com/files/microcontrollers/doc/ref_manual/DRM055.pdf