

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

«Национальный исследовательский ядерный университет «МИФИ»

Факультет очно-заочного (вечернего) обучения

Кафедра 27

«Микро- и наноэлектроники»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к выпускной квалификационной работе
по направлению 140800 – Ядерные физика и технологии
на тему:

АВТОМАТИЗИРОВАННАЯ СИСТЕМА ИЗМЕРЕНИЯ И ТРЕНИРОВКИ
АККОМОДАЦИИ ГЛАЗ

Студент: _____ (_Кузнецов Д.П. _)

Руководитель: _____ (_Саруханов В.А. _)

Рецензент: _____ (_Коблов К.Ю. _)

Заведующий кафедрой: _____ (_Першенков В.С _)

Оценка ГЭК _____ (_____)
расшифровка подписи _____

Секретарь ГЭК _____ (_____)
расшифровка подписи _____

24 мая 2016 г.

Москва

АННОТАЦИЯ

В выпускной квалификационной работе рассмотрено проектирование автоматизированной системы измерения и тренировки аккомодации глаз. Целью работы является создание системы, которая вкупе с законченной конструкционной базой, будет конкурентноспособным прибором на рынке целевой аудитории. В начале осветим общие понятия, поставим задачу, далее рассмотрим имеющиеся выбор компонентов, сформируем компонентную базу, в последующей главе представлены электрические схемы и в конце работы приведем алгоритмическую часть функциональных частей операционной системы, представленных в графах.

ОГЛАВЛЕНИЕ

	Стр.
АННОТАЦИЯ	1
ГЛОССАРИЙ	4
ВВЕДЕНИЕ	5
ГЛАВА 1 ЛИТЕРАТУРНЫЙ ОБЗОР	7
1.1 Введение	7
1.2 Устройство и принцип действия	7
1.3 Режимы работы	9
1.3.1 Контроль аметропии	9
1.3.2 Контроль объема аккомодации	9
1.3.3 Проведение тренировки аккомодации	10
1.3.4 Контроль остроты зрения	10
1.4 Постановка задачи	11
1.4.1 Техническое задание	11
1.4.2 Уточнение задания к дипломной работе	14
1.4.3 Общая структура аппарата	14
ГЛАВА 2 КОМПОНЕНТНАЯ БАЗА	17
2.1 Введение	17
2.2 Контроллер	17
2.3 Привод	18
2.4 Концевой выключатель	20
2.5 Датчики определения слайда	21
2.6 Датчик угла поворота	23
2.7 SD картридер	23
2.8 Джойстик	23
2.9 Дисплей клиента	24
2.10 Заключение	24
ГЛАВА 3 АППАРАТНАЯ ЧАСТЬ	31
3.1 Введение	31
3.2 Датчик угла поворота	31
3.3 VGA	32
3.4 Заключение	33

ГЛАВА 4 ОПЕРАЦИОННАЯ СИСТЕМА	35
4.1 Введение	35
4.2 Структура ОС	35
4.3 Графическая оболочка	37
4.4 Модуль позиционирования	38
4.5 Модуль внешнего запоминающего устройства	39
4.6 Модуль визуализации клиента.....	39
4.7 Заключение	40
ЗАКЛЮЧЕНИЕ	41
СПИСОК ЛИТЕРАТУРЫ	42
ПРИЛОЖЕНИЕ А	43
ПРИЛОЖЕНИЕ В	46

ГЛОССАРИЙ

Аметропия — это изменение преломляющей способности человеческого глаза, следствием которого является то, что задний фокус глаза не попадает на сетчатку при расслаблении аккомодационной мышцы.[1]

Аккомодация — приспособление органа либо организма в целом к изменению внешних условий (значение близко к термину «адаптация»).[2]

Аккомодометр [аккомод(ация)+ греч. metreo измерять, определять] — прибор для исследования аккомодации глаза.[3]

Удерживающий момент - момент необходимый статору для блокировки ротора, когда шаговый двигатель находится под напряжением, но не вращается.[4]

ВВЕДЕНИЕ

В нынешнее время технологии достигли уровня, когда автоматизированные системы используются повсеместно и тем не менее остаются не занятые ниши. На отечественном рынке медицинского оборудования очень маленький выбор устройств, которые могут измерить глубину аккомодации глаза (аккомодометров), а то небольшое количество видов устройств что есть(см. рис.1) не имеют электроприводов из-за чего движение внутренних механизмов осуществляется за счет мускульного усилия пациента, либо врача, так же на таких устройствах крайне не удобно выполнять тренировочные упражнения, которые достаточно важны для увеличения, либо для профилактики уменьшения глубины аккомодации.

Множество других недостатков по сравнению с автоматизированным устройством, где все измерения, запись параметров, создание журнала измерений, математики расчетов средних и их погрешностей, конспект результатов осмотра пациента и много другое делается быстрее, с приложением наименьших усилий врача и пациента. С использованием электроники можно создать востребованное устройство, в этой работе представлен проект создания автоматизированной системы современного аккомодометра(далее аппарата) разработанного по заказу научно-производственной лабораторией "Медоптика".



Рисунок 1 — Аккомодометр АКА-01.

ГЛАВА 1

ЛИТЕРАТУРНЫЙ ОБЗОР

1.1 Введение

Аппарат для тренировки и контроля аккомодации является простым прибором. При работе пациент наблюдает через монокулярную оптическую систему тест-объект. За счет его перемещения изменяется положение наблюданного изображения относительно глаза пациента. Разные тест-объекты, ориентированы на функционально различное назначение.

Фиксируя крайние положения резко наблюданного объекта, можно проанализировать аметропию для дали и для близи, а также объем аккомодации. Плавное перемещение объекта в пределах объема аккомодации обеспечивает возможность тренировки механизма аккомодации глаза. Острая зрения для дали и для близи контролируется при использовании тест-объектов с таблицами различных оптотипов.

Аккомодометр позволяет исследовать астигматизм и определять главные меридианы астигматического глаза. Проводить контроль ночной миопии при пониженной яркости или освещенности тест-объекта.

Все измерения осуществляются субъективным методом, поскольку основаны на оценке пациентом качества наблюданного изображения.

Аппарат рекомендуется для использования в офтальмологической практике при лечебных и оздоровительных мероприятиях. По назначению лечащего врача-офтальмолога аппарат может использоваться в домашних условиях, как правило, для проведения процедур по тренировке аккомодации, закрепляющих амбулаторное лечение. Длительность и периодичность воздействия устанавливаются в зависимости от медицинских показаний.

1.2 Устройство и принцип действия

Аппарат может использоваться для контроля объема аккомодации (как проксиметр), а также для определения аметропии, астигматизма и направ-

ления главных меридианов астигматического глаза (как оптиметр).

Действие аппарата основано на монокулярном наблюдении слайда (или картинки дисплея), изображение которого может перемещаться относительно глаза пациента. На рис.1.1 изображена оптическая схема аппарата. Наблюдаемым тест-объектом может быть как дисплей 5 или освещенный дисплеем слайд 6.

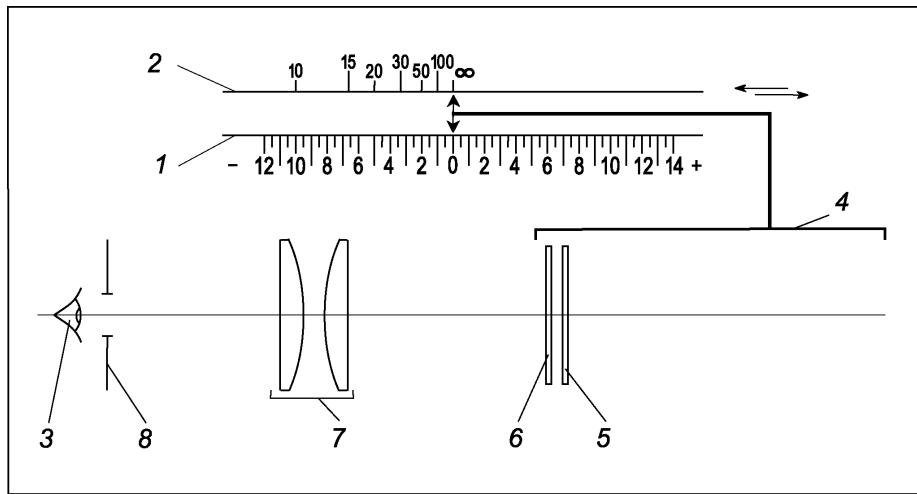


Рисунок 1.1 — Оптическая схема аппарата.

При использовании слайда, между экраном и слайдом помещается матовый светофильтр. Объектив 7 формирует изображение объекта. Это изображение наблюдается глазом 3, который располагается вблизи выходного отверстия 8 оптической системы. Компоненты 5 и 6 конструктивно образуют узел каретки 4, перемещаемый относительно остальной части оптической системы.

Положение каретки контролируется по шкалам 1 и 2. Верхняя шкала 2 оцифрованным в сантиметрах и определяет расстояние S между изображением и глазом. Если объект располагается в фокусе объектива (как это показано на рис. 1.1), то изображение находится в бесконечности ($S = \infty$). При перемещении объекта влево (ближе к глазу) объектив формирует мнимое изображение, когда точка пересечения лучей, прошедших через объектив, находится правее объектива. Расстояние S изменяется от максимального значения (свыше 100 см) до минимальной величины (менее 10 см).

Нижняя шкала 1 оцифрована в диоптриях и предназначена для опреде-

ления аметропии глаза (A). Показания шкал связаны между собой соотношением $A = \frac{-100}{S}$. Расположение объекта в фокусе ($S = \infty$) соответствует нулевому отсчету по диоптрийной шкале: $A = 0$.

Чем ближе каретка к глазу, тем на меньшем расстоянии наблюдается изображение. Так, например, при $A = -2$ дптр изображение находится от глаза на расстоянии $S = 50$ см, при $A = -4$ дптр на расстоянии 25 см. Положительные значения аметропии соответствуют смещению объекта правее фокуса. При этом объектив формирует действительное изображение, расположенное левее глаза.

1.3 Режимы работы

1.3.1 Контроль аметропии

Аметропия определяется в двух точках: в ближней и дальней, максимальная острота зрения соответствует моменту, когда главный фокус глаза располагается на сетчатке, а значит это положение, если оно крайнее, дальнняя или ближняя точка. Ниже на рис.4.9 приведено изображение используемое для тест-объектов.



Рисунок 1.2 — Расположение знаков в тест-объекте.

1.3.2 Контроль объема аккомодации

Объем аккомодации (ΔA см. табл.1.1) — это то расстояние между ближней и дальней точкой, в которой глаз четко видит изображение. Следовательно, находим положения ближней и дальней точек, и их разность равна объему аккомодации.

Возраст, лет	$\Delta A_{\text{дптр}}$	Возраст, лет	$\Delta A_{\text{дптр}}$
10	12 - 14	40	3 - 8
16	10 - 14	45	2 - 6
20	9 - 13	50	1 - 3
25	8 - 12	55	0,75 - 1,75
30	6 - 10	60	0,5 - 1,5
35	5 - 9	•	•

Таблица 1.1 – Возрастные нормы абсолютной аккомодации (по Дуане).

1.3.3 Проведение тренировки аккомодации

Тренировка механизма аккомодации глаза производится при перемещении слайда в пределах установленного объема аккомодации с периодическими попытками расширения его границ. Тренировку проводят поочередно каждым глазом. При тренировке плавно перемещают слайд от дальней границы резкого видения к ближней границе и обратно. Следует стремиться к расширению границ аккомодации: при близорукости – дальней границы, при дальнозоркости – ближней. Приближение слайда к границе аккомодации будет приводить к размытию изображения. Продолжительность тренировки каждого глаза составляет 3 ... 7 мин. В профилактических целях тренировку можно проводить после работы, связанной со значительными зрительными нагрузками.

1.3.4 Контроль остроты зрения

Контроль остроты зрения проводится с помощью специальных слайдов см. рис.1.3. Особенности построения оптической системы аппарата обеспечивают сохранение углового размера знака при перемещении слайда. За счет этого один и тот же слайд может использоваться для контроля зрения вдаль и вблизи.

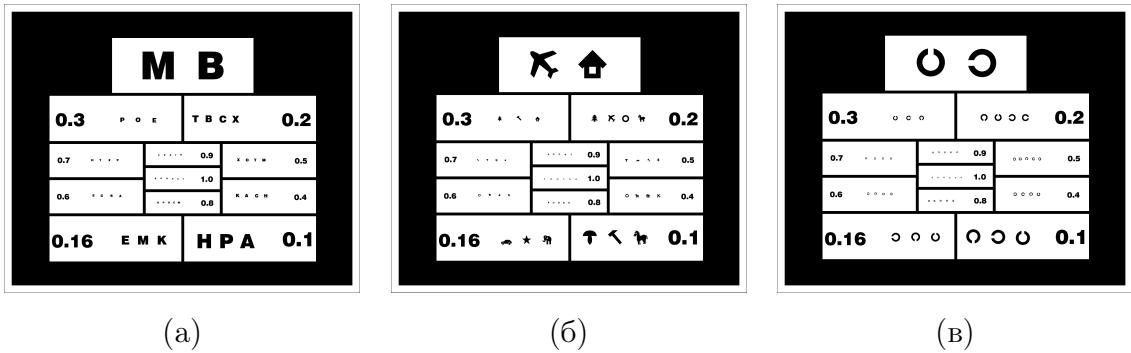


Рисунок 1.3 — Тест-объекты для контроля остроты зрения.

1.4 Постановка задачи

1.4.1 Техническое задание

Научно-производственная лаборатория "Медоптика" поставила следующее техническое задание:

1. Разработать модуль управления аппарата для тренировки и анализа аккомодации.
2. Модуль управления должен включать основной блок и блок интерфейсов.
3. Требования к основному блоку:
 - 3.1. Основной блок модуля управления должен размещаться на подвижной каретке. Электронное управление основным блоком и его питание должно поступать от блока интерфейса.
 - 3.2. Управление перемещением каретки должно осуществляться основным блоком. Управление перемещением каретки должно осуществляться тремя способами:
 - (1) с помощью рукоятки с угловым поворотом;
 - (2) посредством внешнего джойстика и клавиатуры, подключаемых к блоку интерфейсов;
 - (3) посредством команды, сформированной программой интерфейса.

Выбор способа управления перемещением выполняется с помощью программы интерфейса.

3.3. Основной блок должен содержать цветной светоизлучающий дисплей и слайд, располагаемый перед дисплеем; дисплей и слайд предназначены для формирования объекта наблюдения. При наличии слайда дисплей должен формировать одноцветные изображения для подсветки частей слайда; подсвеченные части слайда становятся объектом наблюдения. При отсутствии слайда дисплей должен формировать полноцветное изображение, которое становится объектом наблюдения в этом случае.

3.4. Основной блок должен передавать результаты измерений на монитор VGA, а хранение результатов должно осуществляться на извлекаемой флеш-памяти.

3.5. Основной блок должен при включении устанавливать каретку в позицию реперной точки.

3.6. Основной блок должен содержать узел распознавания слайда. Узел распознавания слайда должен определять отсутствие слайда и номер слайда при его наличии.

4. Требования к блоку интерфейсов:

4.1. Блок интерфейсов должен размещаться в стационарной задней части корпуса аппарата.

4.2. Блок интерфейсов должен содержать блок питания для создания всех необходимых вторичных напряжений для питания основного блока и блока интерфейсов.

5. Требования к конструкции модуля управления:

5.1. Между основным блоком и блоком интерфейсов связь должна обеспечиваться с помощью плоского шлейфа.

5.2. Основной блок должен иметь минимальные габариты, соизмеримые с применяемым дисплеем и габаритами каретки. Согласовывается с конструктором.

5.3. Конструкция модуля должна включать концевые контактные выключатели для электронной (программной) фиксации граничных положений каретки.

5.4. Конструкция аппарата должна содержать шаговый двигатель для перемещения каретки. Предоставляется Заказчиком (тип двигателя ука-

жет конструктор). Шаговый двигатель расположен неподвижно в корпусе аппарата.

6. Требования по питанию:

6.1. Входное напряжение питания должно быть в диапазоне от 5 до 12 В для возможности применения стандартного адаптера на 220В. Стандартный адаптер (источник питания) должен размещаться в стационарной задней части корпуса аппарата.

6.2. Блок питания должен содержать защиту от короткого замыкания и от переполюсовки входного напряжения.

7. Требования к функциональной части:

7.1. Должно быть обеспечено выполнение алгоритмов измерения, тренировки и индикации результатов в соответствии с описанием работы Аппарата. Желательным является предоставление заказчику возможности для самостоятельного корректирования/развития алгоритмов работы Аппарата после завершения проекта.

7.2. Прошивка модуля управления должна обеспечивать управление перемещением каретки вперед и назад (или вверх и вниз) в прямой зависимости от органов управления (п.3.2).

7.3. При срабатывании концевых выключателей двигатель перемещения каретки должен останавливаться.

7.4. Прошивка модуля управления должна определять номер установленного слайда.

7.5. Прошивка должна обеспечивать засветку определенных частей слайда в зависимости от предварительно установленных центров для объектов на слайде. Имеется два типа слайдов:

- с набором из 10 ... 15 мелких объектов в центральной зоне слайда; расположение центров одинаковое для всех слайдов этого типа;

- с одним большим объектом в центре слайда.

1.4.2 Уточнение задания к дипломной работе

В рамках дипломной работы, не ставится задача создать законченный аппарат, поскольку конструкционная разработка ведется отдельно, достаточно разработать электронику и создать операционную систему с целью демонстрации функциональных возможностей на макете и дальнейшей отладкой программного обеспечения.

1.4.3 Общая структура аппарата

Система автоматизации должна управлять устройством, общие принципы которого описаны выше, так чтобы выполнение всех необходимых функций, так же описанных выше, было простым, удобным, быстрым, эффективным как для врача (далее администратора), так и пациента (далее клиента).

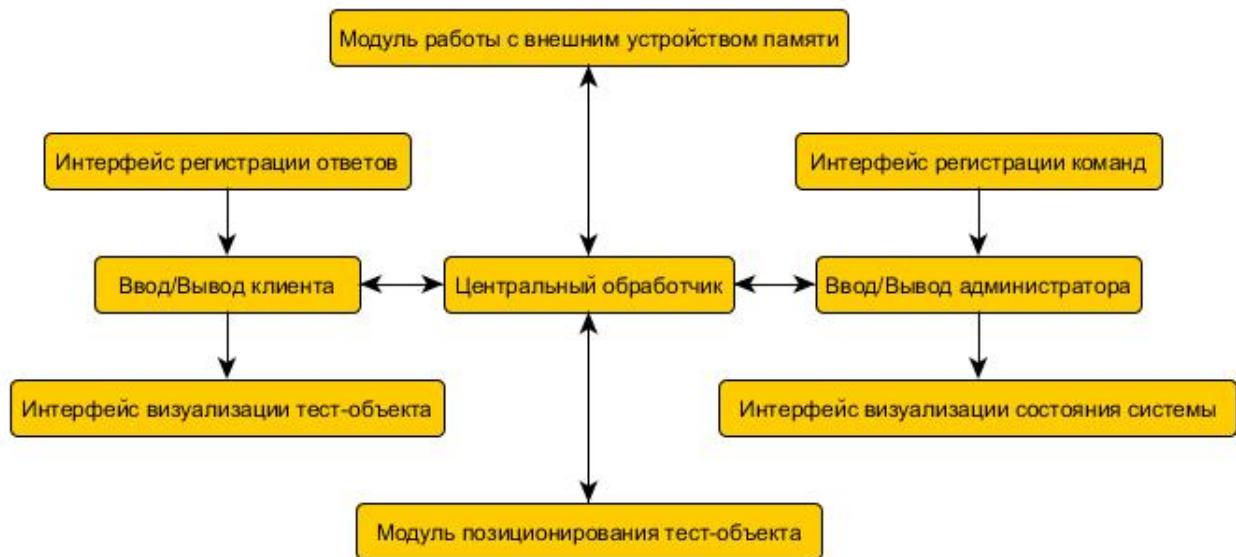


Рисунок 1.4 — Функциональная структура устройства.

На блок-схеме (см.рис.1.4) показана функциональная структура устройства. Центральный обработчик - основной блок, выполняет задачи транзакций между модулями, статистическую обработку данных, расчет графиче-

ской оболочки, управление периферией и работу с внешним запоминающим устройством.

На блок-схеме(см.рис.1.5) показана аппаратная структура устройства. Далее в следующих главах мы выберем компонентную базу, подробно изучим элементы аппаратной структуры устройства, спроектируем электрическую схему и перейдем к операционной системе.

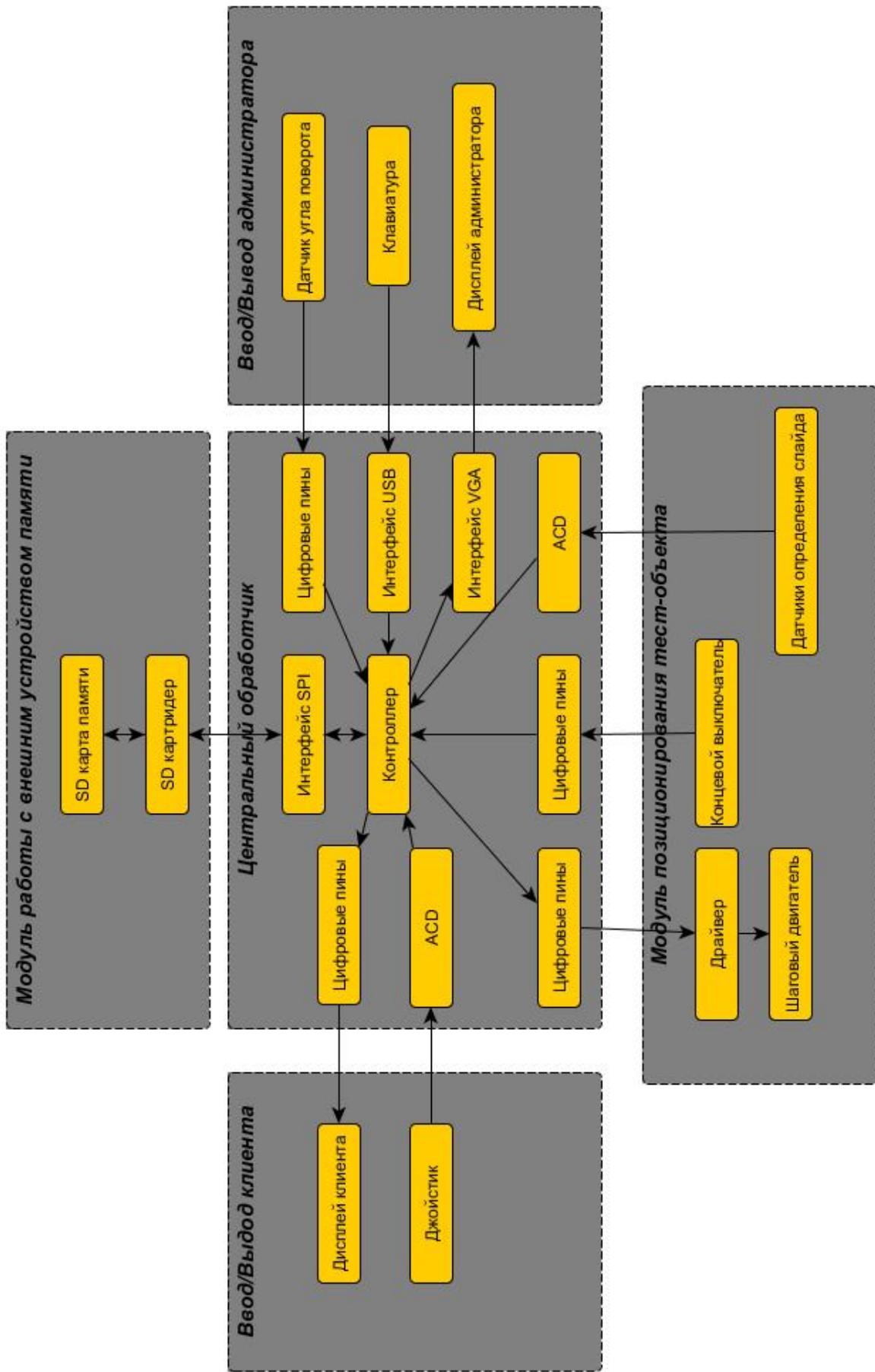


Рисунок 1.5 — Аппаратная структура устройства.

ГЛАВА 2

КОМПОНЕНТНАЯ БАЗА

2.1 Введение

Выбор компонентной базы — важная задача, для выполнения которой необходимо учитывать один, два или комплекс параметров. Так же каждый компонент требует индивидуального подхода. В выборе компонентной базы мы будем руководствоваться следующими параметрами:

- простота интеграции;
- дешевизна;
- эргономичность.

2.2 Контроллер

Если выбирать один контроллер, то он должен поддерживать интерфейс USB, SPI, иметь достаточное количество цифровых и аналоговых входов/выходов(в соответствии с предыдущей главой см.рис.1.5 на странице 16), а кроме этого, частота тактирования должна быть достаточно высокой, чтобы поддерживать VGA интерфейс. На рынке есть процессор Atmel SAM3X8E ARM Cortex-M3, на его базе плата DUE V1.6 от Elecfreaks(см. рис.2.1) со следующими характеристиками, см. табл.2.1[5].

На плате DUE V1.6 имеется 54 цифровых входа/выхода, 12 аналоговых входов, 4 UARTa (аппаратных последовательных порта), а генератор тактовой частоты 84 МГц, связь по USB с поддержкой OTG, 2 ЦАП (цифроаналоговых преобразователя), 2 TWI, разъем питания, разъем SPI, разъем JTAG, кнопка сброса и кнопка стирания. Так же 12 аналоговых входов, каждый из которых может обеспечить разрешение 12 бит (т.е. 4096 различных значений).

Так как эта линейка контроллеров от Elecfreaks очень популярна, DUE V1.6 выпускается большими сериями и дешевле конкурентов, кроме того, для него существует масса библиотек и сопутствующих технических реше-

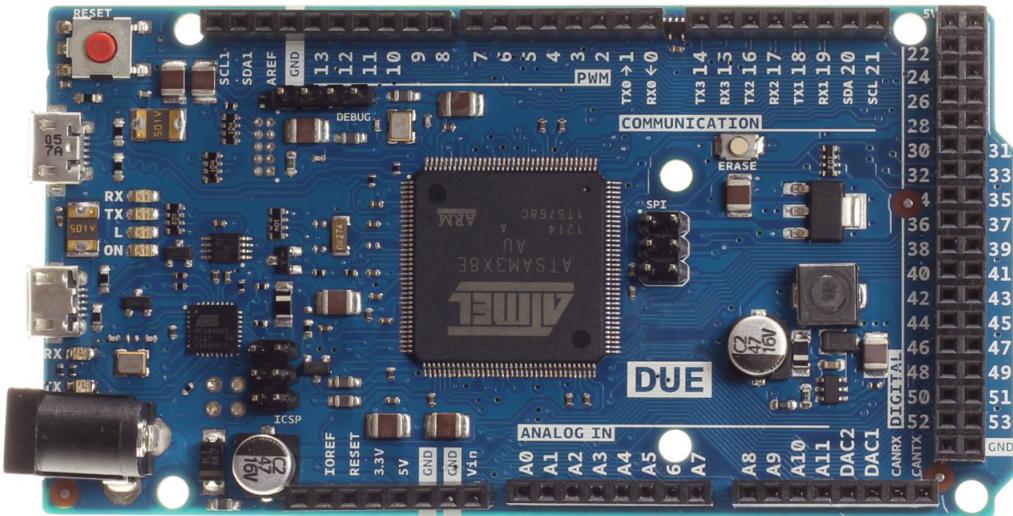


Рисунок 2.1 — Фотография платы.

ний, которые позволяют уменьшить срок выхода на рынок и повысить ремонтопригодность. Надо отметить простоту интеграции в разработку благодаря проекту "DueVGA" с открытым кодом на сайте распределённой системы управления версиями GitHub[6]. Этот проект имеет уже готовую библиотеку для работы с видео интерфейсом VGA. Остановим выбор на этом контроллере.

2.3 Привод

Привод должен не только выполнять задачу передачи врачающего момента, но так же, чтобы упростить модуль позиционирования тест-объекта, имеет смысл использовать двигатель с управляемым углом поворота. Одно из лучших решений данной задачи - это шаговый двигатель. Большую часть рынка занимают гибридные 2-х/3-х/5-ти фазные шаговые двигатели. Производители заверяют, что погрешность таких двигателей не больше 5% из-за технологических несовершенств зубцов(точек интенсивных магнитных полей, к которым разворачивается ротор). Имеем следующие параметры системы(см. табл.2.2)

Расчет момента удержания шагового двигателя, из уравнения 2.1, из 2.2 получим момент вращения вала $M = 18.33$ [Н*см].

Микроконтроллер	AT91SAM3X8E
Рабочее напряжение	3,3 В
Входное напряжение (рекомендуемое)	7-12 В
Входное напряжение (предельное)	6-20 В
Цифровые Входы/Выходы	54
Аналоговые входы	12
Аналоговые выходы	2 (ЦАП)
Общий выходной постоянный ток	50 мА
Постоянный ток через вывод 3,3 В	800 мА
Постоянный ток через вывод 5 В	800 мА
Флеш-память	512 КБ
ОЗУ	96 КБ(64 КБ и 32 КБ)
Тактовая частота	84 МГц

Таблица 2.1 — Характеристики платы.

Скорость передвижения каретки V , см/сек	10
Радиус передачи момента вращения от вала к ремню R , см	0.5
Вес каретки m , Н	30
Коэффициент усилия для передвижения каретки θ	1.1
КПД передачи η	0.9

Таблица 2.2 — Параметры системы

Из-за физических особенностей шаговых двигателей, крутящий момент двигателя при малых частотах вращения вала примерно равен удерживающему моменту. И наоборот, выходной крутящий момент уменьшается с увеличением частоты вращения вала, так же и выходная мощность, и удерживающий момент становится одним из наиболее важных параметров шагового двигателя. На графике зависимости крутящего момента от частоты вращения для гибридных шаговых двигателей 2.2 видно, что при частоте $\nu = 191$ крутящий момент ослабляется на 15%, наш привод должен разви-

вать удерживающий момент не меньше $M_p = 18.3 * 1.15 = 21$ [Н*см]

$$\nu = \frac{V * 60}{2 * \pi * R} = \frac{10 * 60}{2 * \pi * 0.5} = 191 \quad (2.1)$$

$$M = \frac{m * \theta * R}{\eta} = \frac{30 * 1.1 * 0.005 * 100}{0.9} = 18.33 \quad (2.2)$$

Шаговый двигатель 17HS8401(см. рис.2.3) развивает удерживающий крутящий момент $M_p = 52$ [Н*см], минимальный угол поворота 1.8 градуса, погрешность $\xi = \frac{1.8*0.05*2*\pi*0.5}{360} = 8 * 10^{-3}$ [мм]. 17HS8401 применяется в 3D принтерах и, как следствие активно используется, имеет отработанные решения, готовые драйверы — это упростит встраивание в наш проект.

Драйвер — компонент, цель применения которого — упростить подключение шагового двигателя к контроллеру. Напряжение управления немощным шаговым двигателем обычно находится в диапазоне от 12 до 48 вольт, а значит перед контроллером необходим усилительный каскад, если не использовать драйвер. Одно из готовых решений для шагового двигателя 17HS8401 - сборка MP1510 с драйвером A4988 (см. рис.2.4). A4988 и модуль драйвера шагового двигателя(MP1510) содержат набор усилительных каскадов, которые при появлении командного сигнала подают напряжение в 12 вольт на вход двигателя. Контроллер управляет драйвером импульсами напряжения 5 вольт и периодом не меньше 2 мкс.

2.4 Концевой выключатель

Конструкция концевого выключателя (см. рис. 2.5, табл. 2.3) оптимизирована для использования в системах управления:

- малогабаритный прочный корпус (обычно изготавливаемый из металла) имеет элементы конструкции, позволяющие легко закрепить и сориентировать в пространстве;
- индикация работы (поданного питания) и срабатывания датчика выполнены при помощи ярких разноцветных светодиодов;
- подключение через общераспространённые однопиновые разъёмы.

Оптопара	TCST2103
Номинальный ток, А	50mA
Контактная группа	Фото прерыватель
Рабочее напряжение DC, V	3,2-5
Размеры (HxLxW), мм	7x32x15

Таблица 2.3 — Характеристики концевого выключателя

2.5 Датчики определения слайда

При помещении слайда в каретку система определяет диапозитив, его пространственную ориентацию (см. рис. 2.7), для этого у каждого слайда должен быть оригинальный идентификационный номер и атрибут, по которому можно определить его расположение в каретке. Количество возможных диапозитивов должно быть не меньше 20-ти.

Размещение по периметру слайда светопроницаемых/частично светопроницаемых/светонепроницаемых ячеек (см. рис. 2.6) дает возможность закодировать идентификационный номер и определить пространственную ориентацию слайда, для считывания ячеек используются оптопары. На одну сторону диапозитива помещается 4 ячейки, размещать ячейки можно только с верху и низу слайда. Частично светопроницаемые ячейки позволяют использовать многоуровневую(трехуровневую) кодировку.

Рассчитаем количество идентификационных номеров, учитывая, что когда все оптраны получают сигнал неискаженный ячейками - каретка без слайда.

- для одной декодирующей группы (4 оптрана):
 - при вращении в горизонтальной плоскости (см. рис. 2.7)
 - * для 2-х уровней логики возможно различить 7 слайдов
(комбинаций $2^4 - 1 = 15$, где 1-случай когда каретка пуста)
 - * для 3-х уровней логики возможно различить 40 слайдов
(комбинаций $3^4 - 1 = 80$, где 1-случай когда каретка пуста)
 - при вращении в 2-х плоскостях (см. рис. 2.7)

Оптрон	1	2	3	4	5
Погрешность, В	0,3	0,4	0,1	0,7	0,5

Таблица 2.4 — Погрешности разных оптронов KTIR0521DS.

- * для 2-х уровней логики возможно различить 2 слайда (комбинаций $2^4 - 2^2 = 4$, где 2^2 -комбинаций не распознаваемых из-за симметрии)
- * для 3-х уровней логики возможно различить 36 слайдов (комбинаций $3^4 - 3^2 = 72$, где 3^2 -комбинаций не распознаваемых из-за симметрии)
- для 2-х декодирующих групп (8 оптронов):
 - при вращении в горизонтальной плоскости (см. рис. 2.7)
 - * для 2-х уровней логики возможно различить 127 слайдов (комбинаций $2^8 - 1 = 255$, где 1-случай когда каретка пуста)
 - при вращении в 2-х плоскостях (см. рис. 2.7)
 - * для 2-х уровней логики возможно различить 120 слайдов (комбинаций $2^8 - 2^4 = 240$, где 2^4 -комбинаций не распознаваемых из-за симметрии)

На следующем снимке осциллографа (см. рис. 2.8) показано как выходной сигнал с оптрона KTIR0521DS изменяется при изменения светопроницаемой способности кодирующей ячейки.

Измерениях выходных сигналов для одинаковых степеней проницаемости и разных оптронов KTIR0521DS показали погрешность (см. табл. 2.4), среднеквадратичная погрешность $S_{n<X>} = \frac{S_n}{\sqrt{n}} = \sqrt{\frac{\sum_{n=0}^{\infty} (X - \langle X \rangle)^2}{n*(n-1)}} = 0,288$ вольт, тогда для вероятности попадания в доверительный интервал 99% ($t = 4.6$, см.таблицу коэффициентов Стьюдента) получим интервал $\Delta X = t * S_{n<X>} = 1.32$ вольта. Выходной сигнал оптрона находится в диапазоне от 0 до 4.5 вольт, следовательно, мы можем использовать трехуровневую систему кодировки - 40 различимых вариантов диапозитивов.

2.6 Датчик угла поворота

Инкрементальный датчик угла поворота (энкодер), в проекте его основная задача - дублировать главную систему управления кареткой со стороны администратора, на тот случай, если такое управление окажется субъективно удобным. Энкодер должен будет иметь не менее 20 импульсов поворота на 360 градусов для плавности управления.

Датчик угла поворота PEC16-4220F-S0024 (см. рис. 2.9) имеет 24 импульса поворота на круг, кодирование сигнала передается по 2 битам в коде Грея, напряжение питания 5 вольт. С точки зрения конструкции у этого энкодера есть необходимая крепежная штанга. Устройство дешево и долговечно.

2.7 SD картридер

В современных SD картах есть два режима работы SD и SPI. Последний рассчитан на 8 разрядные микроконтроллеры или на те микроконтроллеры, у которых нет аппаратной поддержки SD режима. Для подключения SD карты мы будем использовать монтажный модуль SD card reader SPI интерфейса (см. рис. 2.10).

2.8 Джойстик

Аналоговые джойстики — сенсоры выдающие сигнал значением от нуля до максимума в зависимости от угла отклонения ручки (стика): чем больше рукоять отклонена, тем больше значение напряжения, в пределах опорного. Двумерные джойстики позволяют передавать значения ориентации стика в плоскости. Так же двумерные джойстики могут иметь кнопку, нажатие на которую производится утоплением стика, например (см. рис. 2.11) основанный на 2-х регулируемых делителях напряжения (2 оси задающие ориентацию на плоскости) и ключа(кнопка).

2.9 Дисплей клиента

Дисплей играет важную роль вывода информации, дисплей клиента будет не только выводить информацию, отображая изображение, а также при вставке слайда его задачей будет освещать необходимый сектора пленки диапозитива. OEL(Organic Electro Luminescent) дисплеи обладают характерной для этих органических матриц высокой насыщенностью, четкостью, яркостью и контрастностью изображения.

Дисплей (см. рис. 2.12) с разрешением 160 x 128 и цветовой палитрой в 262144 цвета, имеет единственный недостаток — 35 пиновый интерфейс подключения, что в свою очередь замедлит встраивание, но это не критично.

2.10 Заключение

Подведем итоги, выбрана компонентная база (см. табл. 2.5) в соответствии с техническим заданием заказчика, на основе которой будет построена система управления аппаратом.

Компонент	Артикул
Контроллер	Due
Шаговый двигатель	17HS8401
Концевой выключатель	TCST2103
Датчики определения слайда	KTIR0521DS
Датчик угла поворота	PEC16-4220F-S0024
SD картридер	SD Card Module
Джойстик	KeysSJoys
Дисплей клиента	UG-6028GDEAF01

Таблица 2.5 — Таблица компонентов.

В следующей главе детально рассмотрим аппаратную часть с приведением электрических схем и принципов их построения, а так же приведем

те элементы электроники которые не попали в эту главу.

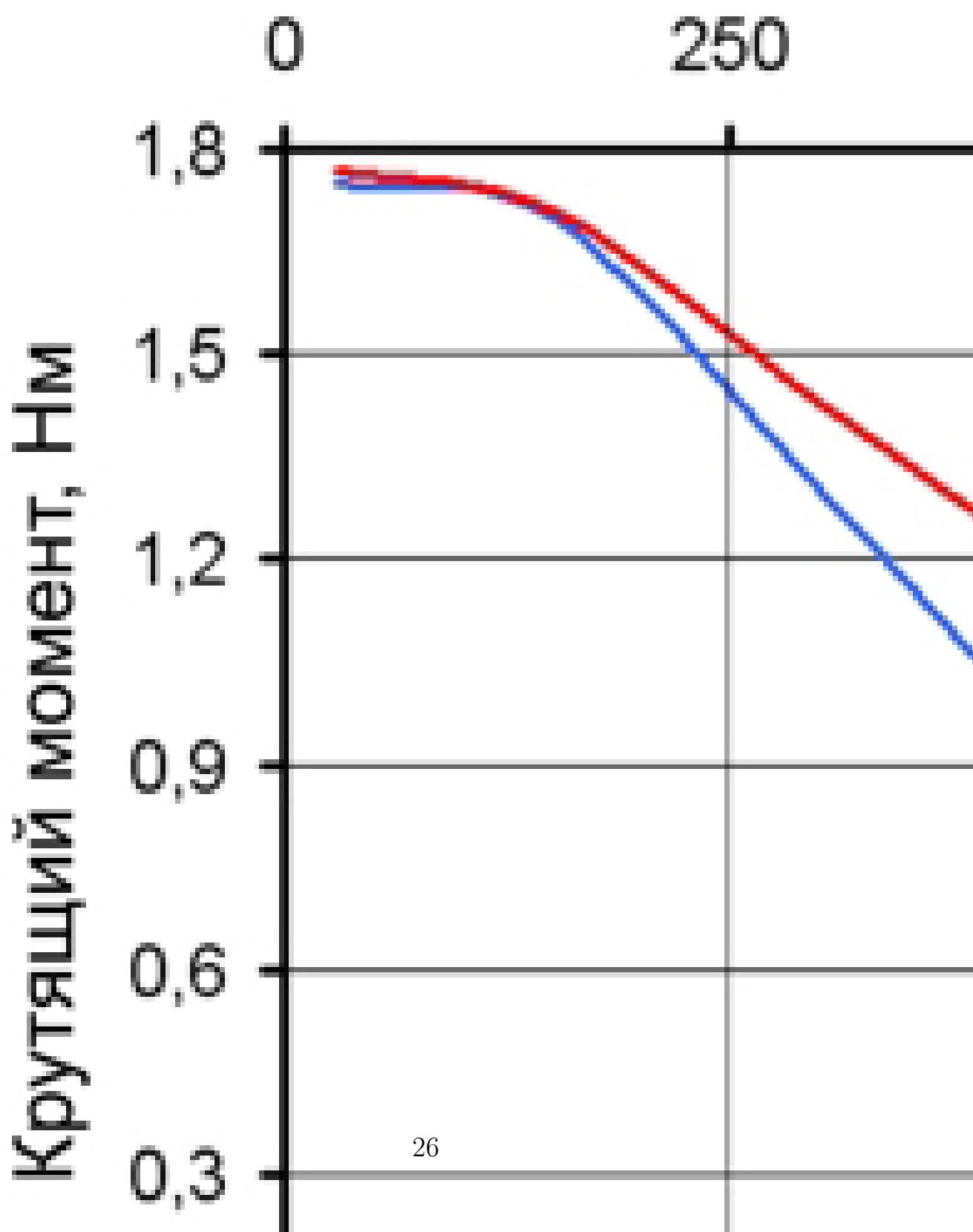




Рисунок 2.3 — Фотография шагового двигателя.

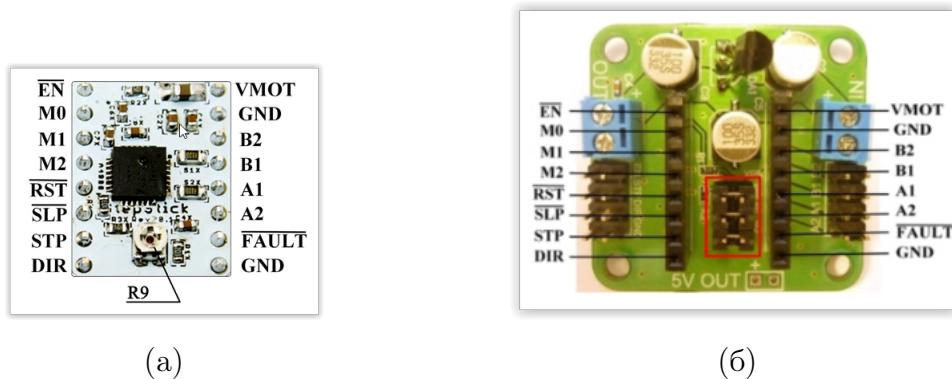


Рисунок 2.4 — (а) Драйвер шагового двигателя A4988; (б) Универсальный модуль подключения драйвера ш.д. MP1510



Рисунок 2.5 — Оптический ограничитель передвижения каретки.

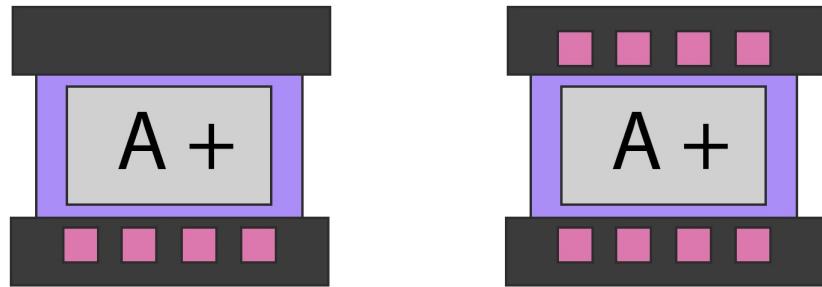


Рисунок 2.6 — Слева одна декодирующая группа каретки (4 оптрана), справа две декодирующие группы каретки (8 оптранов).

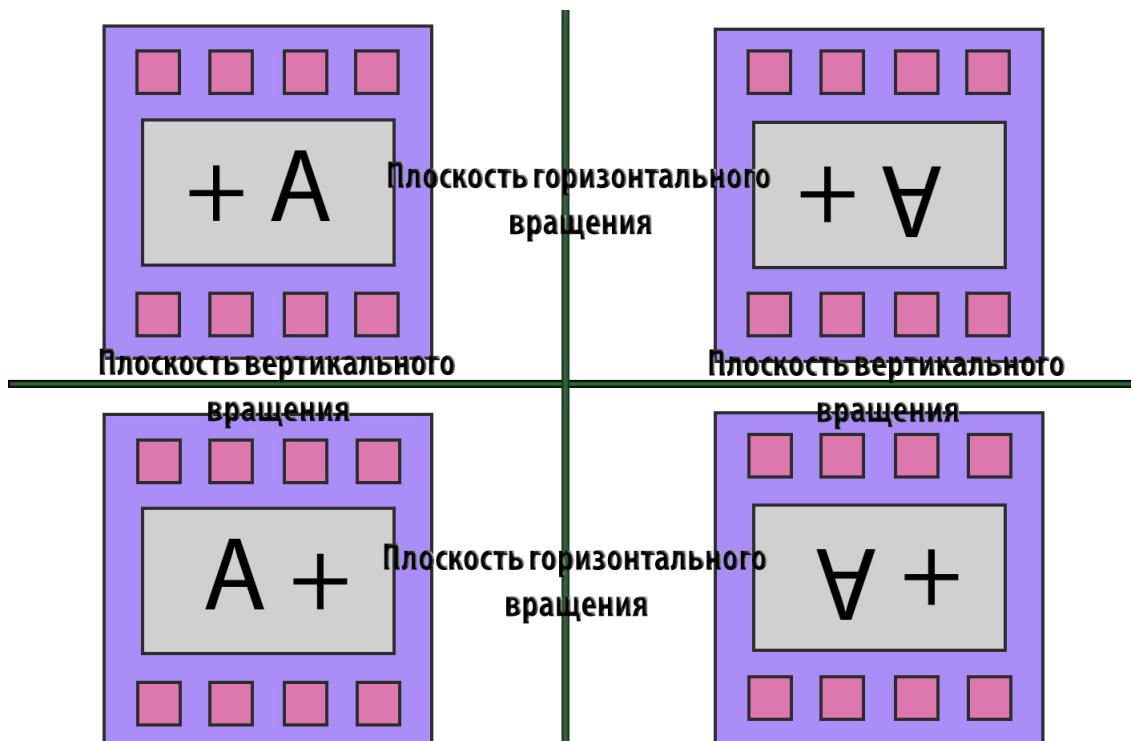


Рисунок 2.7 — Вращение в двух плоскостях плоскости.

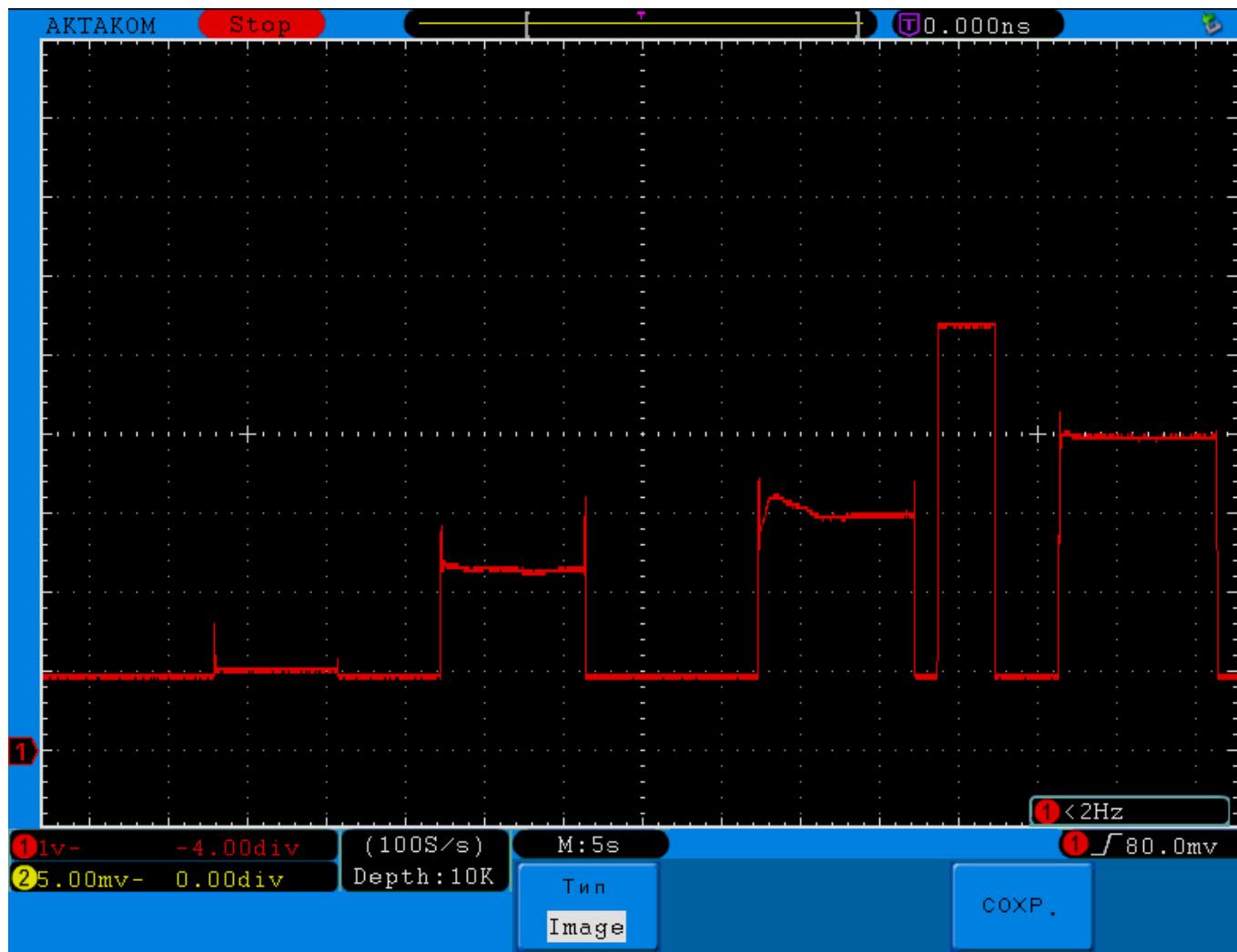


Рисунок 2.8 — Изменение выходного сигнала оптрона KTIR0521DS при разной светопроницаемой способности кодирующей ячейки.



Рисунок 2.9 — Датчик угла поворота PEC16-4220F-S0024.



Рисунок 2.10 — Монтажный модуль SD картридера.



Рисунок 2.11 — Монтажный модуль двумерного Джойстика.



Рисунок 2.12 — OEL дисплей SAS1-I003-A UG-6028GDEAF01.

ГЛАВА 3

АППАРАТНАЯ ЧАСТЬ

3.1 Введение

В данной работе объединительной платой назовем группу электрических разъемов соединенных друг с другом, напрямую или посредством специальных электрических цепей, использующихся в качестве основы, чтобы соединить несколько печатных плат вместе и составить полную систему. Полная схема объединительной платы представлена в приложении А. Далее в этой главе рассмотрим специальные соединительные электрические цепи.

3.2 Датчик угла поворота

При снятии с датчика угла поворота (энкодера) сигналов во время переключения барабана наблюдается дребезг контактов, так как обработка сигнала контроллером ведется по прерыванию фронта канала А энкодера, необходимо создать надежную защиту от появления шумовых фронтов. Для исключения возможности попадания дребезга были использованы цепочка фильтра низких частот и триггер Шмидта (SN54AC14FK), имеющих гистерезис как функцию преобразования сигнала (см. рис. 3.2).

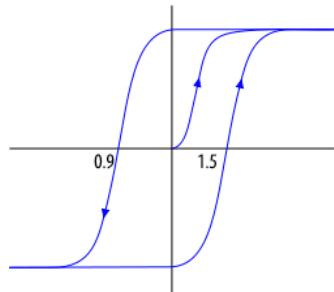


Рисунок 3.1 — Петля гистерезиса SN54AC14FK.

Дребезг может длиться порядка 10 мс, время за которое конденсатор зарядится на 99% $\tau = 3 * C * R = 3$ мс, триггер Шмидта имеет толщину гистерезисной петли от 0,4 до 1,5 вольта (см. рис. 3.1).

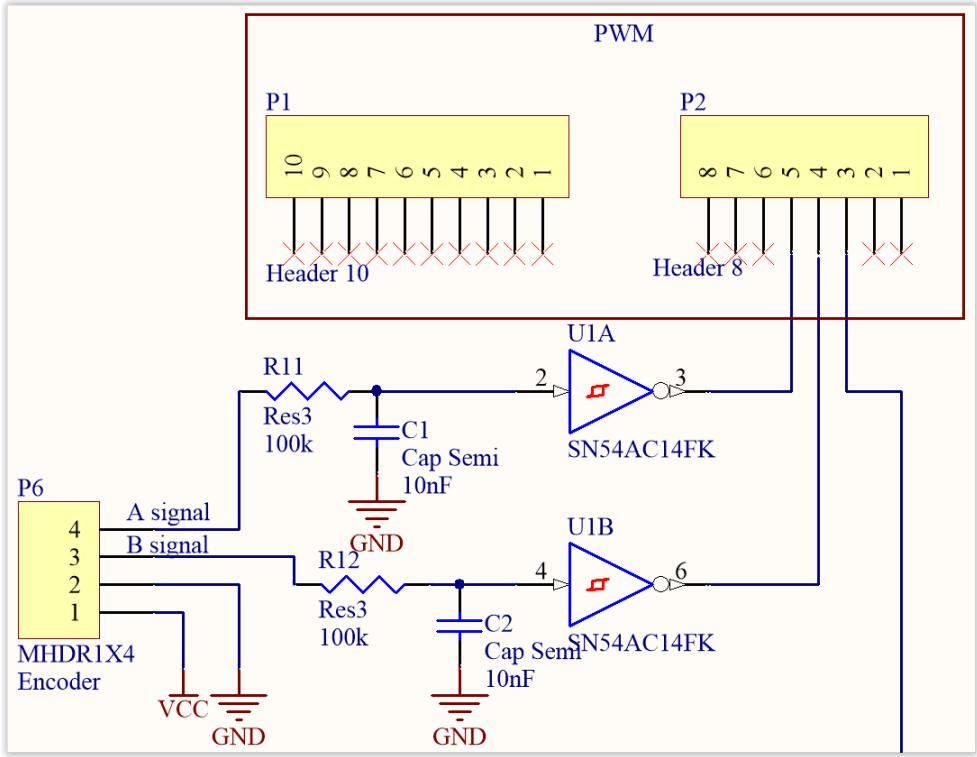


Рисунок 3.2 — Специальная соединительная схема датчика угла поворота.

3.3 VGA

Видеоадаптер VGA, в отличие от предыдущих видеоадаптеров, использует аналоговый сигнал для передачи цветовой информации. Переход на аналоговый сигнал был обусловлен необходимостью сокращения числа проводов в кабеле.

Stimmer, автор открытой библиотеки кодов DueVGA, использовал изящное решения задачи подключения к контроллеру VGA интерфейса - программный делитель напряжения. Его механизм заключается в том, что к выходу подключены резисторы разного номинала, на которые подается напряжение 0 или 5 вольт, в зависимости от того на какие резисторы и какое напряжение (0 или 5) подается, на выходе мы получаем разное напряжение и, как итог, разные цвета отображаемого изображения. Разворотка строк и фреймов задается сигналами Hsync и Vsync. Специальная соединительная схема VGA-контроллеров содержит соединенные группы резисторов реализующие это решение (см. рис. 3.3).

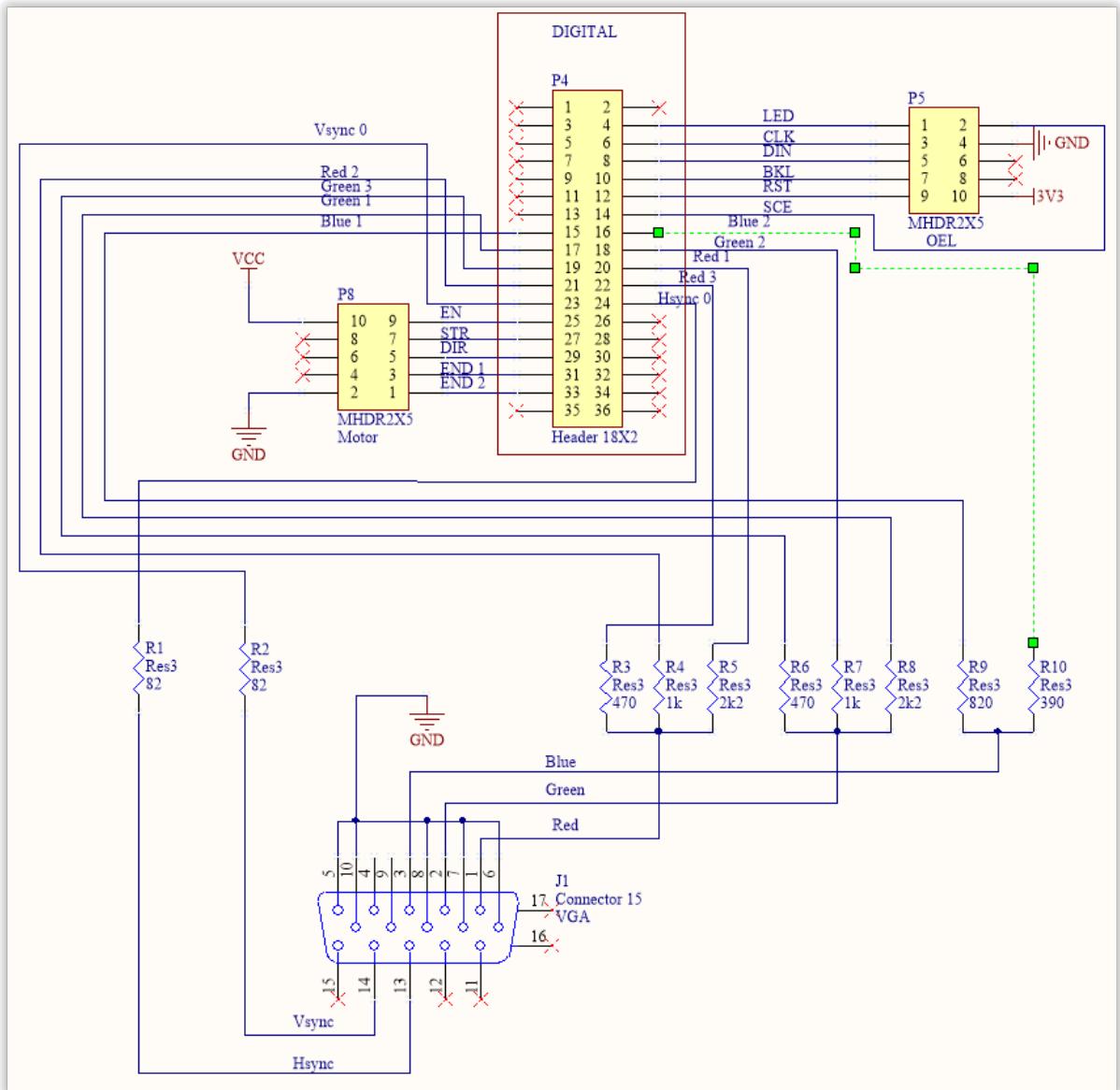


Рисунок 3.3 — Специальная соединительная схема VGA

3.4 Заключение

Полностью рассмотрена реализуемая аппаратура, полная электрическая схема объединительной платы представлена в приложении А. Готовая электрическая схема позволит уйти от быстро изготовленных и менее надежных монтажных плат (см. рис. 3.4), которые часто становятся причиной плохих соединений и, как результат, замедляют процесс разработки, ну и конечно, к моменту производства должна быть готова нормальная разводка плат для массового выпуска.

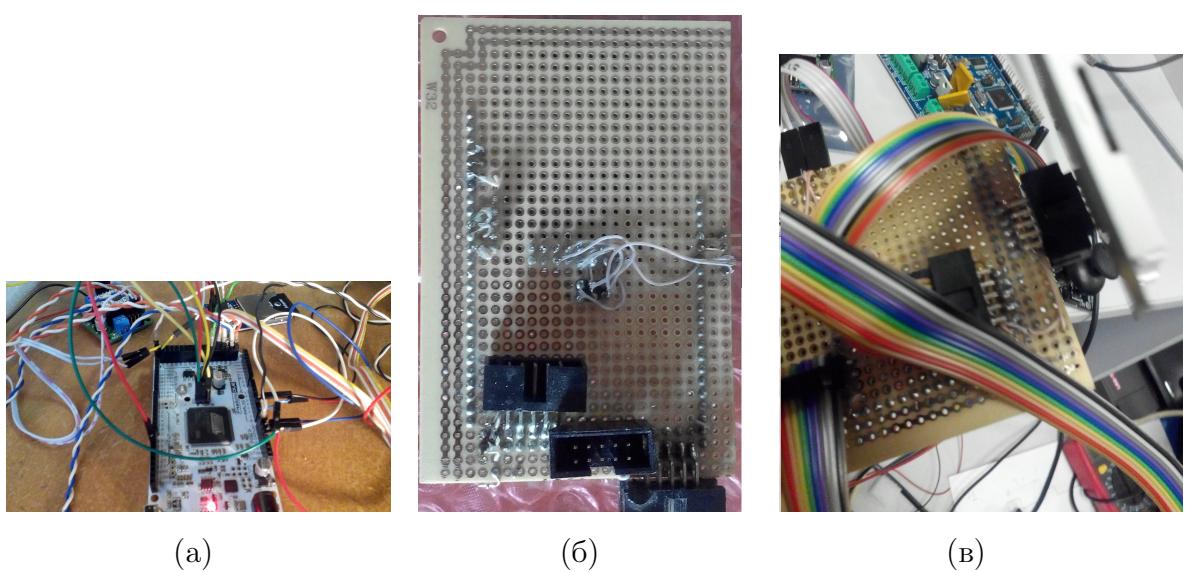


Рисунок 3.4 – (а) - Без объединенной монтажной платы;
(б) - Объединенная монтажная плата без шин;
(в) - Объединенная монтажная плата с шинами.

В следующей главе рассмотрим реализацию ОС и ее функций.

ГЛАВА 4

ОПЕРАЦИОННАЯ СИСТЕМА

4.1 Введение

Под операционной системой подразумевается комплекс взаимосвязанных функций, предназначенных для управления ресурсами устройства и организации взаимодействия с администратором и клиентом. В этой главе мы рассмотрим из чего состоит написанная операционная система.

4.2 Структура ОС

Назовем единицей системы состояние, в котором администратор/клиент обладает определенным набором функций, активируемых через нажатие на клавиатуру, джойстик и энкодер, т.е. являющихся обработчиком прерываний интерфейсов управления.

Сегмент - это структура, внутри которой расположен закольцованный граф, любая вершина которого связана с двумя вершинами "Переход" а так же внешние вершины "Возврат" связаны с первой вершиной внутреннего графа (см. рис.4.1).

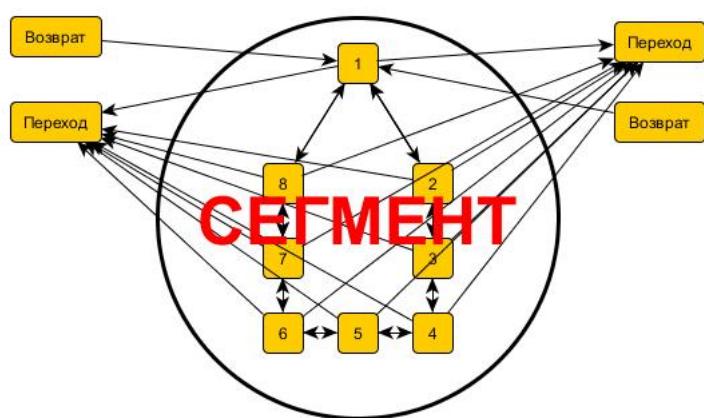


Рисунок 4.1 — Сегмент системы состоит из графа состояний.

Каждый сегмент имеет уникальное графическое представление. Два сегмента связаны друг с другом через вершины "Переход" и "Возврат",

где направление перехода от вершины "Переход" к вершине "Возврат". При переходе от одного сегмента к другому в вершине "Переход" вызываются все необходимые функции (функция завершения выполняемых процессов, функция сохранения актуальных данных, функция очистки выделенной динамической памяти) для завершения работы сегмента. Далее в вершине "Возврат" вызываются все необходимые функции (функции выделения динамической памяти под новые сущности, функция инициализации, функции загрузки информации предыдущей сессии) для загрузки и запуска следующего сегмента (см. рис.4.2).

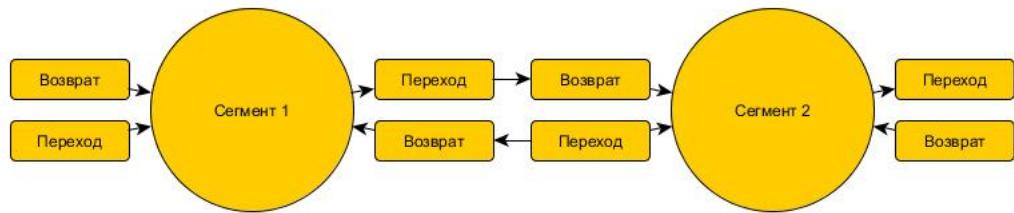


Рисунок 4.2 — Связи двух сегментов.

Каждый сегмент ункален и отвечает за свое графическое окно(окно ввода данных клиента, окно тренировки, окно измерения и т.д.), связи сегментов эквиваленты переходам между окнами (см. рис.4.3).

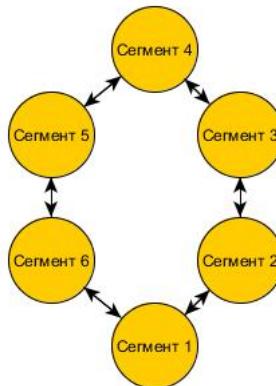


Рисунок 4.3 — Граф связей сегментов.

4.3 Графическая оболочка

Вся графика была написана на основе библиотеки VGA внутри которой была только попиксельная прорисовка и терминальный способ вывода сообщений.

Графическая оболочка, которая была создана для проекта, это набор созданных форм графического интерфейса, все написанные формы хранятся в библиотеке VGA_windows. Форма обычна заполняет каким-либо контентом весь экран, контент представляет собой набор объектов класса CntrlSubGroup из библиотеки cntrlsubgroup (см. Приложение B), есть также цветовые стили, которые хранятся в библиотеке colorsourse. Ниже приведены фотографии, сделанные вовремя работы системы(см. рис.4.4,4.5,4.6)



Рисунок 4.4 – Окно ввода данных клиента 1.

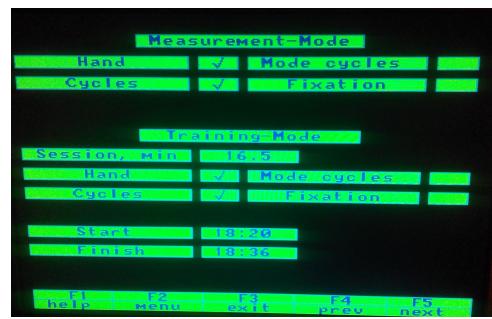


Рисунок 4.5 – Окно ввода данных клиента 2.

Представленные снимки показывают стиль с зеленым бэкграундом объектов, цветовой стиль интерфейса можно поменять.

Библиотека VGA была без поддержки русского языка, кодовые страницы для русского пришлось внедрять отдельно, а так же дополнять функции

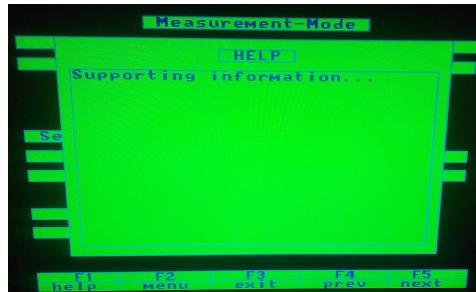


Рисунок 4.6 — Окно помощи.

цией трансляции, для ввода текста с клавиатуры

4.4 Модуль позиционирования

Модуль управления положением (см. табл.4.7), запускается только при нахождении пользователя в определенном сегменте системы. Запуск начинается с инициализации (определение точки отсчета и возврат на базовую.) Сигналы управления исполняются в следующем приоритете:

- сигнал концевого выключателя;
- клавиатура;
- энкодер;
- джойстик.



Рисунок 4.7 — Модуль позиционирования.

Этот модуль использует opensource библиотеку KeyBoard и написанные библиотеки:

- управление двигателем и считывание сигналов с концевого выключателя (A4988);

- считывание сигналов с джойстика, энкодера обработка прерываний клавиатуры (Keyes_Sjoys);

4.5 Модуль внешнего запоминающего устройства

Базируется на opensource библиотеки SD. Модуль внешнего запоминающего устройства содержит набор функций упрощающих работу с SD картой, например, запись конспекта осмотра больного, чтения из памяти файлов изображения и параметров слайда для модуля визуализации клиента (см. стр. 39), сохранение статистики, сохранение постоянных настроек и т.д.

4.6 Модуль визуализации клиента

Объединяет систему распознавания слайдов и OEL дисплей, он так же как и модуль позиционирования включен в строго определенные сегменты. Для расширенной работы этого модуля необходимо наличие SD карты с информацией по набору слайдов, либо набор файлов изображения которые будет показывать OEL. В случае отсутствия SD система не будет работать в тех режимах, в которых используется содержимое SD.

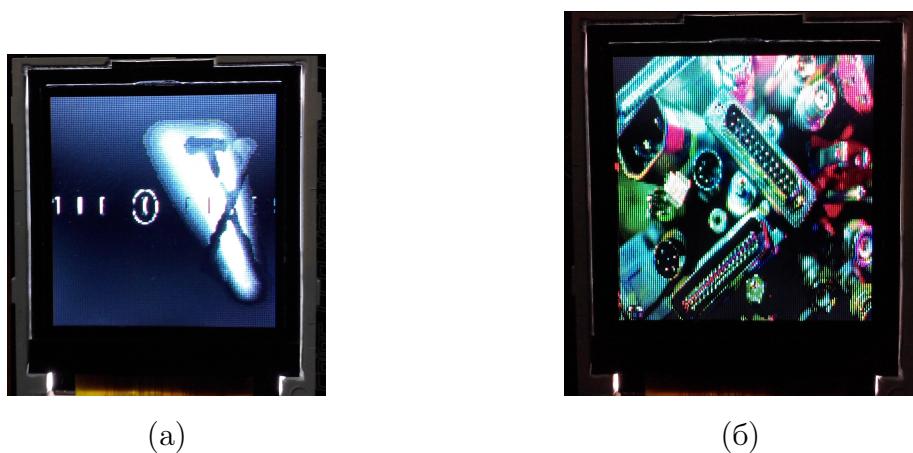


Рисунок 4.8 – (а,б) - загруженные изображения в OEL дисплей.

Инициализация слайда происходит при считывании оптопарами светопроницаемых ячеек, далее определение идентификационного номера, после

чего по номеру в библиотеке наборов слайдов (эта библиотека содержит информацию по каждому слайду и расположена на SD карте) определяется расположение тестовых изображений в плоскости слайда. Так как слайд стоит перед дисплеем, дисплей может подсвечивать определенные области слайда, тем самым делая видимым определенную картинку, подсвечивание происходит в случайном порядке.

4.7 Заключение

В этой главе была рассмотрена структура и функции ОС. Ограничения контроллера привели к небольшому усложнению системы (создание функций освобождения и выделения памяти под необходимые ресурсы), но благодаря тому, что современные языки программирования, в нашем случае C++, дают возможность абстрагироваться посредством создания классов, шаблонов и т.д., задача была решена. Многие решения были приняты с учетом будущего развития системы.

ЗАКЛЮЧЕНИЕ

В результате дипломной работы была создана автоматизированная система измерения и тренировки аккомодации глаз. В рамках дипломной работы поставленная задача выполнена, по мере ее создания было использовано много интересных решений (подавление помех сигнала энкодера, создание напряжение для передачи сигнала цвета VGA, организация ОС и т.д.). При углублении в тему проекта со стороны заказчика было встречено и принято несколько расширяющих задач (подключение USB HUB, отображение на мониторе тест-картинок и т.д.).

На данном этапе заказчик создает для разработанной системы конструкцию, которая заменит макетную платформу, в тоже время идет отладка программного обеспечения и разработка печатной платы и решение других задач, нацеленные на приведение проекта к завершающей стадии готового продукта.

СПИСОК ЛИТЕРАТУРЫ

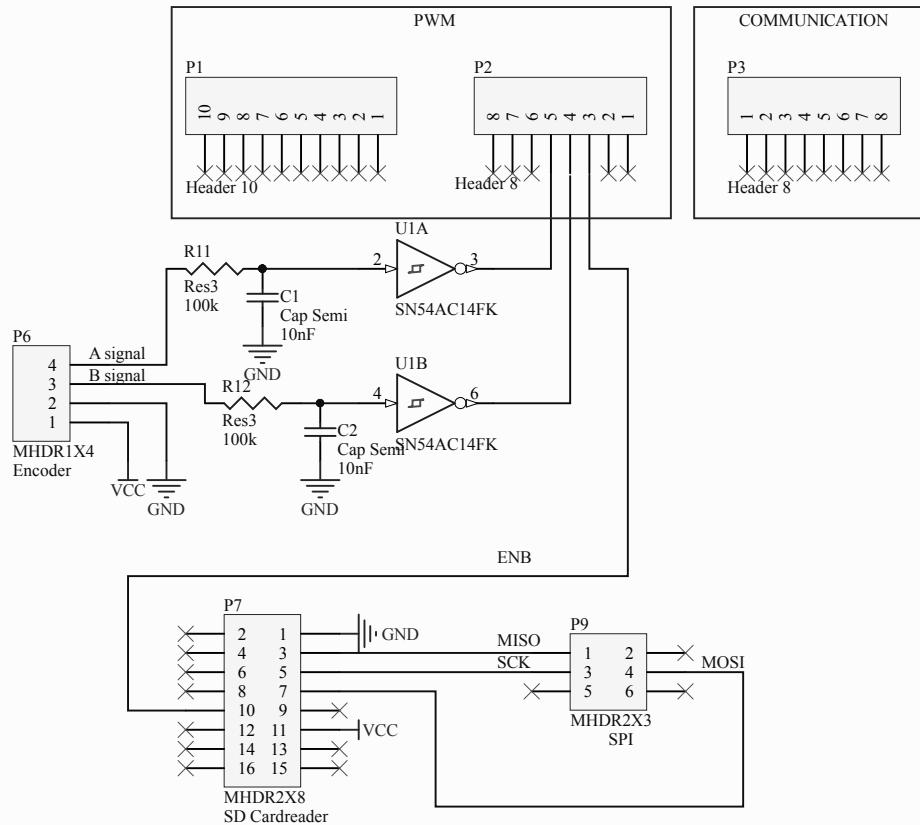
1. Энциклопедический словарь Брокгауза и Ефона : в 86 т. •, 1890—1907.
2. Большая советская энциклопедия. «СОВЕТСКАЯ ЭНЦИКЛОПЕДИЯ», 1926 — 1990.
3. Малая медицинская энциклопедия. «СОВЕТСКАЯ ЭНЦИКЛОПЕДИЯ», 1991-1996.
4. Co. Changzhou Fulling Motor. Базовые знания о шаговых двигателях. 2015. URL: <http://www.motor-fulling.ru/stepper-motor-knowledge.html/122697>.
5. Arduino. Arduino Due. 2015. URL: <http://arduino.ru/Hardware/ArduinoDue>.
6. stimmer. DueVGA. 2013. URL: <https://github.com/stimmer/DueVGA>.

ПРИЛОЖЕНИЕ А

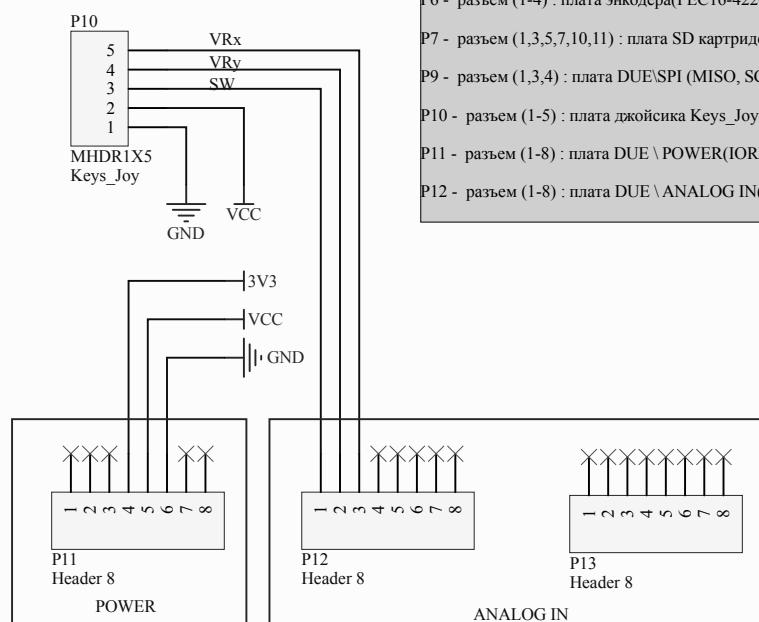
Таблица 1.2
Таблица коэффициентов Стьюдента

Число изме- рений	Надежность, α							
	0,6	0,7	0,8	0,9	0,95	0,98	0,99	0,999
2	1,38	2,0	3,1	6,3	12,7	31,8	62,7	53,7
3	1,06	1,3	1,9	2,9	4,3	7,0	9,9	31,6
4	0,98	1,3	1,6	2,4	3,2	4,5	5,8	12,9
5	0,94	1,2	1,5	2,1	2,8	3,7	4,6	8,6
6	0,92	1,2	1,5	2,0	2,6	3,4	4,0	6,9
7	0,91	1,1	1,4	1,9	2,4	3,1	3,7	6,0
8	0,9	1,1	1,4	1,9	2,4	3,0	3,5	5,4
9	0,89	1,1	1,4	1,9	2,3	2,9	3,4	5,0
10	0,88	1,1	1,4	1,8	2,3	2,8	3,3	4,8
11	0,88	1,1	1,4	1,8	2,2	2,7	3,2	4,6
12	0,88	1,1	1,4	1,8	2,2	2,7	3,1	4,5
13	0,87	1,1	1,4	1,8	2,2	2,7	3,1	4,3
14	0,87	1,1	1,4	1,8	2,2	2,7	3,0	4,2
15	0,87	1,1	1,3	1,8	2,1	2,6	3,0	4,1
16	0,87	1,1	1,3	1,8	2,1	2,6	3,0	4,1
17	0,87	1,1	1,3	1,7	2,1	2,6	2,9	4,0
18	0,86	1,1	1,3	1,7	2,1	2,6	2,9	4,0
19	0,86	1,1	1,3	1,7	2,1	2,6	2,9	3,9
20	0,86	1,1	1,3	1,7	2,1	2,5	2,9	3,9
∞	0,84	1,0	1,3	1,6	2,0	2,3	2,6	3,3

Рисунок 4.9 — Таблица коэффициентов Стьюдента.



P1 - разъем (1-10) : плата DUE \ PWM(8-13),GND,AREF,SDA1, пустой;
 P2 - разъем (1-8) : плата DUE \ COMMUNICATION(RX,TX),PWM(0-7);
 P3 - разъем (1-8) : плата DUE \ COMMUNICATION(14-21);
 P6 - разъем (1-4) : плата энкодера(PEC16-4220F-S0024) \ GND, VCC, A signal, B signal;
 P7 - разъем (1,3,5,7,10,11) : плата SD картридер \ GND, MISO, SCK, MOSI, ENB, VCC;
 P9 - разъем (1,3,4) : плата DUE \ SPI (MISO, SCK, MOSI);
 P10 - разъем (1-5) : плата джойстика Keys_Joy \ GND, VCC, SW, VRy, VRx;
 P11 - разъем (1-8) : плата DUE \ POWER(REF, RESET, 3V3, 5V, GND, VIN);
 P12 - разъем (1-8) : плата DUE \ ANALOG IN(0-7);
 P13 - разъем (1-8) : плата DUE \ ANALOG IN(0-7);

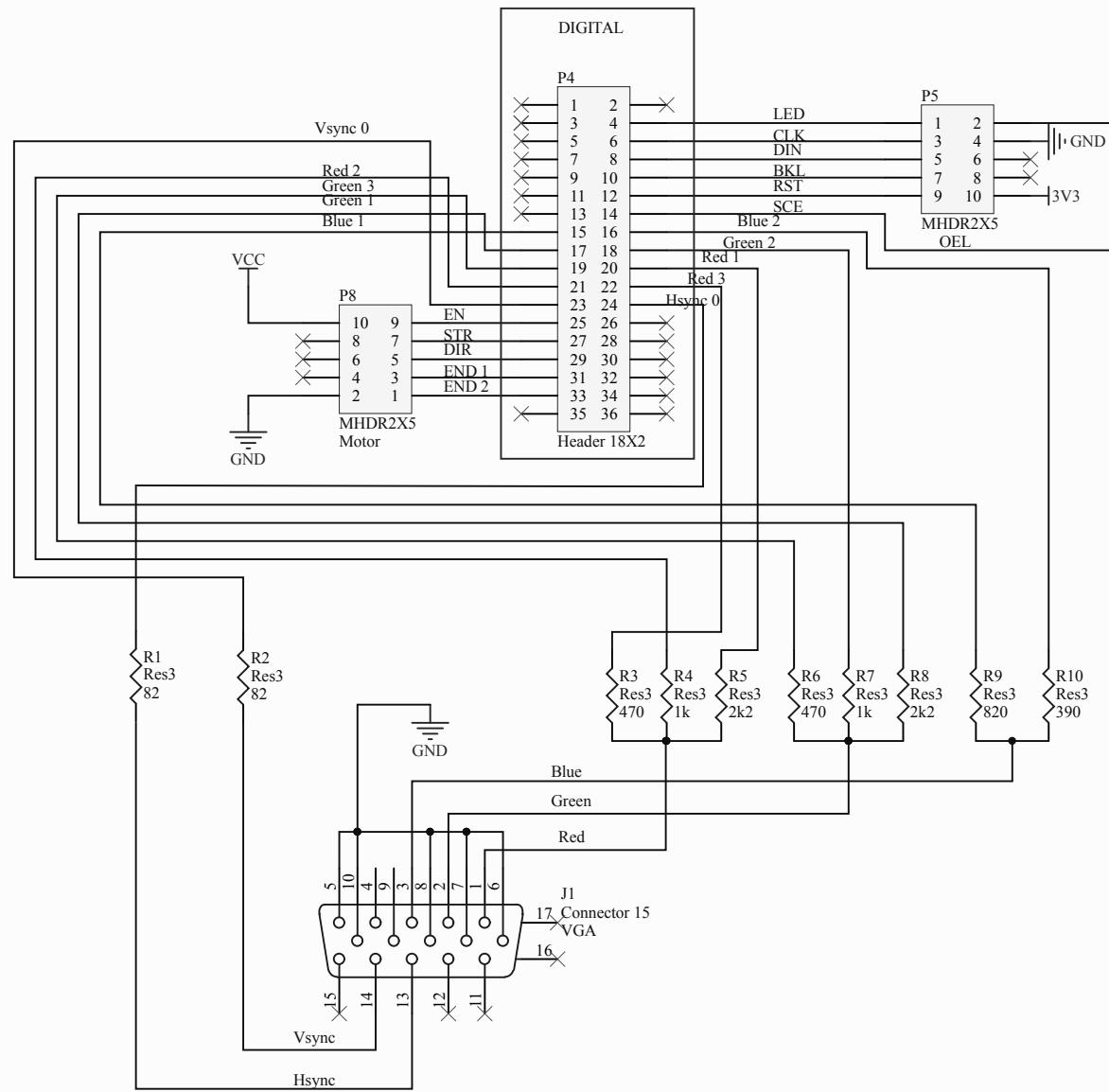


Инв. № подп	Подп. и дата	Взам. инв. № инв. № подп. и дата	Подп. и дата
Разраб.	Кузнецов Д.П.		
Пров.	Саруханов В.А.		
Н.контр.			
Утв.			

Объединительная
плата

Лит. Лист Листов

1 2



P4 - разъем (1-36) : плата DUE \ DIGITAL(VCC, VCC, 22-53,GND,GND);

P5 - разъем (1-5,9,10) : плата OEL дисплея\ LED, SCE, CLK, DIN, BKL, RST, 3V3 ;

P8 - разъем (1,2,3,5,7,9,10) : плата драйвера и концевых выключателей \ GND, END2, END1, DIR, SRT, EN, VCC;

P6 - разъем (1-3,5-8,10,13,14) : VGA \ Red, Green, Blue, GND, GND, GND, GND, GND, Hsync, Vsync.

Инв. № подл	Подп. и дата	Взам. инв. № подл	Подп. и дата	Изм/лист	№ докум.	Подп. Листа
Разраб.	Кузнецов Д.П.					
Пров.	Саруханов В.А.					
Н.контр.						
Утв.						

Объединительная
плата

Лит.	Лист	Листов
Т	2	2

ПРИЛОЖЕНИЕ В

Приведен класс с помощью которого создаются элементы графического интерфейса.

```
#ifndef CNTRLSUBGROUP_H_INCLUDED
#define CNTRLSUBGROUP_H_INCLUDED
#include <VGA.h>
#include <string.h>
#include "intrfsubgroup.h"

class CntrlSubGroup
{
public:
    CntrlSubGroup(Color_Style style, Metric_Parameters metric)
        :ISG(IntrfSubGroup_Init_Parameters(style, metric, &state, text)),
     metric(metric), state(false), modification(false){};
    CntrlSubGroup(Color_Style style, Metric_Parameters metric,
                  const char*_text) :ISG(IntrfSubGroup_Init_Parameters(style,
                  metric, &state, text)), metric(metric), state(false),
                  modification(false){ strcpy_center_text(_text); };
    CntrlSubGroup(Color_Style style, Metric_Parameters metric,
                  int _text) :ISG(IntrfSubGroup_Init_Parameters(style, metric,
                  &state, text)), metric(metric), state(false), modification(false)
    { strcpy_center_text(_text); };
    CntrlSubGroup();
    ~CntrlSubGroup();
    void refresh();
    void update();
    bool strcpy_center_text(const char* _text);
```

```
bool strcpy_center_text(int _ntext);
bool strcpy_text(const char* _text);
bool strcpy_text(int _ntext);
void enable();
void disable();
void state_switch();

private:
void add_space(int n);
void itoa(int n, char s[]);
void reverse(char s[]);
static const int string_length = 30;
IntrfSubGroup ISG;
Metric_Parameters metric;
IntrfSubGroup_State state;
char text[string_length];
bool modification;
};

#endif
```

```
#include"cntrlsubgroup.h"
```

```
CntrlSubGroup::CntrlSubGroup()
{
}
```

```
CntrlSubGroup::~CntrlSubGroup()
{
```

```
}

void CntrlSubGroup::refresh()
{
    ISG.update();
    modification = false;
}

void CntrlSubGroup::update()
{
    if (modification) ISG.update();
    modification = false;
}

bool CntrlSubGroup::strcpy_center_text(const char*_text)
{
    if (metric.height*metric.width > strlen(_text))
    {
        if (metric.height == 1)
        {
            add_space((metric.width - strlen(_text)) / 2);
            strcat(text, _text);
        }
        else
            strcpy(text, _text);
        modification = true;
        return true;
    }
    else
        return false;
}

bool CntrlSubGroup::strcpy_center_text(int _ntext)
{
    char _text[string_length];
    itoa(_ntext, _text);
```

```

if (metric.height*metric.width > strlen(_text) + 1)
{
if (metric.height == 1)
{
add_space((metric.width - strlen(_text)) / 2);
strcat(text, _text);
}
else
strcpy(text, _text);
modification = true;
return true;
}
else
return false;
}

bool CntrlSubGroup::strcpy_text(const char*_text)
{
if (metric.height*metric.width > strlen(_text))
{
strcpy(text, _text);
modification = true;
return true;
}
else
return false;
}

bool CntrlSubGroup::strcpy_text(int _ntext)
{
char _text[string_length];
itoa(_ntext, _text);
if (metric.height*metric.width > strlen(_text))
{

```

```
strcpy(text, _text);
modification = true;
return true;
}
else
return false;
}
void CntrlSubGroup::enable()
{
state.enable();
modification = true;
}
void CntrlSubGroup::disable()
{
state.disable();
modification = true;
}
void CntrlSubGroup::state_switch()
{
state.reverse();
modification = true;
}
void CntrlSubGroup::add_space(int n)
{
for (size_t i = 0; i < n; i++)
text[i] = ' ';
text[n] = 0;
}
void CntrlSubGroup::itoa(int n, char s[])
{
int i, sign;
```

```
if ((sign = n) < 0) /* записываем знак */
n = -n;           /* делаем n положительным числом */
i = 0;
do {           /* генерируем цифры в обратном порядке */
s[i++] = n % 10 + '0'; /* берем следующую цифру */
} while ((n /= 10) > 0); /* удаляем */
if (sign < 0)
s[i++] = '-';
s[i] = '\0';
reverse(s);
}

void CntrlSubGroup::reverse(char s[])
{
int i, j;
char c;

for (i = 0, j = strlen(s) - 1; i < j; i++, j--) {
c = s[i];
s[i] = s[j];
s[j] = c;
}
}
```