

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего
образования

«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет

вычислительной техники

Кафедра

«Вычислительная техника»

Направление подготовки: 09.03.01 «Информатика и вычислительная техника»

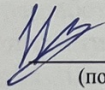
Профиль подготовки: «Вычислительные машины, системы, комплексы и
сети»

БАКАЛАВРСКАЯ РАБОТА

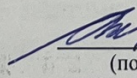
на тему

РОБОТИЗИРОВАННЫЙ КОМПЛЕКС
АВТОМАТИЧЕСКОГО ПРОГРАММИРОВАНИЯ
И ТЕСТИРОВАНИЯ ЭЛЕКТРОННЫХ
МОДУЛЕЙ НА ПЕЧАТНЫХ ПЛАТАХ

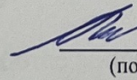
Студент

 20.05.2026 Николаев Александр Алексеевич
(подпись, дата) (ФИО полностью)

Руководитель

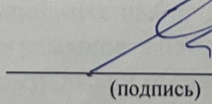
 20.05.2026 Бычков А.С.
(подпись, дата) (фамилия, инициалы)

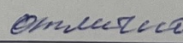
Нормоконтролёр

 20.05.2026 Бычков А.С.
(подпись, дата) (фамилия, инициалы)

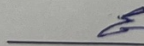
Работа допущена к защите (протокол зас. кафедры от 22.05.2026 г. № 14)

Заведующий кафедрой

 Митрохин М.А.
(подпись) (фамилия, инициалы)

Работа защищена с отметкой  (протокол заседания ГЭК от
02 июня 2026 г. № 2)

Секретарь ГЭК

 Гурин Е.И.
(подпись) (фамилия, инициалы)

Пенза 2026

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет
вычислительной техники

Кафедра
«Вычислительная техника»

«Утверждаю»
Заведующий кафедрой
М.А. Митрохин
«07» апреля 2026 г.

**ЗАДАНИЕ
НА БАКАЛАВРСКУЮ РАБОТУ**

1. Студент Николаев Александр Алексеевич гр. 22BVB1 факультета вычислительной техники.

2. Тема работы: «Роботизированный комплекс автоматического программирования и тестирования электронных модулей на печатных платах»

Тема утверждена приказом ПГУ № 347/о от "7" апреля 2026 г.

3. Руководитель бакалаврской работы к.т.н., доцент кафедры ВТ ПГУ Бычков А.С.

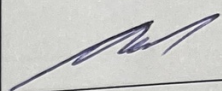
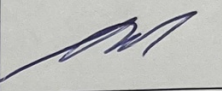
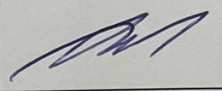
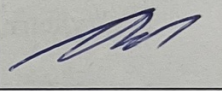
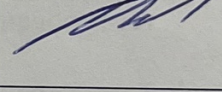
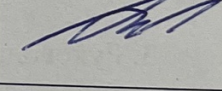
4. Задание на работу (назначение разработки, исходные данные и т.п.)

Разработать роботизированный автоматический комплекс программирования и тестирования электронных модулей на печатных платах. Комплекс должен последовательно перемещать контактное устройство между платами в групповой панели, подключаться к контрольным точкам, выполнять программирование, тестирование, фиксировать ошибки и выводить результаты проверки оператору.

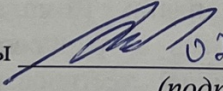
5. Перечень подлежащих разработке вопросов

Произвести анализ существующих технологических решений автоматического контроля печатных плат и обосновать актуальность разработки. Определить назначение, требования и общий состав комплекса. Выполнить выбор аппаратных средств. Разработать общую структуру комплекса. Разработать программное обеспечение для управления комплексом. Произвести опытную проверку работы комплекса и оценить его эффективность.

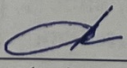
6. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1	Изучение предметной области	32	07.04.2026 – 11.04.2026	
2	Анализ аналогов, актуальности, постановка задачи	24	11.04.2026 – 14.04.2026	
3	Выбор и настройка средств разработки	4	14.04.2026 – 15.04.2026	
4	Определение состава комплекса	30	15.04.2026 – 19.04.2026	
5	Разработка аппаратной части комплекса	100	19.04.2026 – 04.05.2026	
6	Разработка программного обеспечения для управления комплексом	55	04.05.2026 – 11.05.2026	
7	Опытная проверка и оценка эффективности	54	11.05.2026 – 18.05.2026	
8	Оформление пояснительной записки	25	18.05.2026 – 21.05.2026	

Дата выдачи задания " 07 " апреля 2026г.

Руководитель бакалаврской работы  07.04.2026. Бычков А.С.
(подпись, дата)

Задание к исполнению принял студент  07.04.2026 Николаев А.А.
(подпись, дата)

Работу к защите допустить
Декан факультета  Фионова Л.Р.
(подпись, дата)
27.06.2026

Реферат

Бакалаврская работа содержит 97 листов, 30 рисунков, 3 таблицы, 14 источников, 3 приложения.

АППАРАТНО-ПРОГРАММНЫЙ КОМПЛЕКС, АВТОМАТИЧЕСКОЕ ПРОГРАММИРОВАНИЕ, АВТОМАТИЧЕСКОЕ ТЕСТИРОВАНИЕ, ЭЛЕКТРОННЫЙ МОДУЛЬ, ПЕЧАТНАЯ ПЛАТА, СЕРИЙНОЕ ПРОИЗВОДСТВО, ГРУППОВОЙ КОНТРОЛЬ

Цель работы заключается в разработке роботизированного комплекса для автоматического программирования и тестирования электронных модулей на печатных платах.

Объектом исследования в работе являются процессы автоматизации контроля электропараметров и функционирования электронных модулей с использованием координатного манипулятора и специализированного тестового оборудования.

В работе применены методы трёхкоординатного позиционирования на базе ЧПУ-контроллера, автоматического управления технологическими операциями с помощью микроконтроллера Arduino Mega 2560, доработки внутрисхемного программатора PICkit 5 для запуска по внешнему сигналу, а также методы коммутации сигналов через игольчатую контактную матрицу и релейные элементы.

В работе рассмотрены существующие системы контроля печатных плат, проведён анализ их недостатков при групповом контроле панелей, разработан опытный образец комплекса, реализующий последовательное автоматическое программирование и тестирование до 40 модулей, а также обеспечена выдача протокола проверки и режим повторного тестирования неисправных плат.

					ПГУ 1.090301.09.001 ПЗ			
Изм.	Лист	№ докум.	Подп	Дата	Роботизированный комплекс автоматического программирования и тестирования электронных модулей на печатных платах Пояснительная записка	Лит.	Лист	Листов
Разраб.		Николаев А.А.		20.05.20				
Провер.		Бычков А.С.		20.05.20			2	97
Н. Контр.		Бычков А.С.		20.05.20		ФВТ, каф. ВТ гр. 22ВВВ1		
Утверд.		Митрохин М.А.		20.05.20				

Содержание

Перечень сокращений и обозначений	5
Введение.....	6
1 Анализ предметной области	8
1.1 Анализ существующих технологических решений	10
1.2 Анализ актуальности разработки	12
1.3 Постановка задачи	13
1.4 Определение общего состава комплекса	15
1.5 Технические требования	15
1.6 Выбор и настройка средств разработки.....	17
1.6.1 Система автоматизированного проектирования DipTrace.....	17
1.6.2 Среда разработки Arduino IDE	22
1.6.3 Стандартный текстовый редактор Windows «Блокнот»	27
2 Разработка комплекса	29
2.1 Выбор аппаратных частей комплекса	29
2.1.1 Выбор трехкоординатного робот-манипулятора	29
2.1.2 Выбор ЧПУ-контроллера движения	31
2.1.3 Выбор программатора	32
2.1.4 Выбор тестера	33
2.1.5 Выбор управляющего МК.....	34
2.2 Разработка общей структуры комплекса	36
2.3 Разработка контактного тестирующего устройства	37
2.4 Доработка программатора PICkit 5 для автоматического запуска	38
2.5 Разработка системы управления комплексом на базе Arduino Mega 2560 ...	44
2.6 Организация информационного обмена между ЧПУ-контроллером и Arduino Mega 2560	46
2.7 Разработка алгоритма программирования и тестирования модуля	48
2.8 Реализация вывода результатов и режима повторной проверки.....	52
3 Результаты разработки и оценка эффективности комплекса.....	54
3.1 Основные результаты разработки	54
3.2 Результаты опытной проверки партии плат	56

3.3 Оценка эффективности комплекса	57
3.4 Перспективы дальнейшего развития	59
4 Безопасность жизнедеятельности	62
4.1 Характеристика потенциальных рисков при работе с комплексом	62
4.2 Требования безопасности при эксплуатации комплекса	63
4.3 Организация рабочего места оператора	63
4.4 Электробезопасность.....	64
4.5 Пожарная безопасность.....	64
4.6 Санитарно-гигиенические требования	65
Заключение	67
Список использованных источников	68
Приложение А – Схемы электрические.....	70
Приложение Б – Листинг фрагментов программы	75
Приложение В – Презентация.....	93

Перечень сокращений и обозначений

В настоящей работе применяют следующие сокращения и обозначения

АПК – аппаратно-программный комплекс

МК – микроконтроллер

ОЗУ – оперативное запоминающее устройство

ПК – персональный компьютер

ПО – программное обеспечение

САПР – система автоматизированного проектирования

ЧПУ – числовое программное управление

G-code (Geometric Code) – управляющий код, применяемый для задания перемещений в системах ЧПУ

GND (Ground) – общий провод электрической схемы

IDE (Integrated Development Environment) – интегрированная среда разработки

LCD (Liquid Crystal Display) – жидкокристаллический дисплей

M-code (Miscellaneous Code) – вспомогательный управляющий код в системах ЧПУ

RC-цепь (Resistor-Capacitor circuit) – резистивно-ёмкостная цепь

USB (Universal Serial Bus) – универсальная последовательная шина

					ПГУ 1.090301.09.001 ПЗ	Лист
						5
Изм.	Лист	№ докум.	Подпись	Дата		

Введение

В современном приборостроении существует потребность в экономически эффективном автоматизированном контроле функционирования радиоэлектронных модулей и их электропараметров. Это обусловлено тем, что при среднесерийном, а особенно при крупносерийном производстве становится крайне невыгодным применение ручного труда, когда оператор выполняет операции подачи тестовых сигналов в определенные цепи электронного модуля, контролирует выходные сигналы на контрольных точках, сравнивает показания с нормами, заносит результаты проверки в протокол и т. д. Такой подход требует значительных затрат времени, снижает производительность и увеличивает вероятность ошибок, связанных с человеческим фактором.

В настоящее время существуют АПК, позволяющие автоматизировать выполнение указанных операций. По большей части они не применимы при контроле модулей на печатных платах, особенно в групповых сборках, на начальном этапе производства. Они не позволяют сразу исключить ошибки, до установки модуля в конечное изделие, при этом снижая производительность, качество и стабильность технологических процессов.

Таким образом автоматизация производственных процессов приборостроения на всех этапах является одной из актуальных задач современности.

При создании роботизированных автоматических комплексов необходимо обеспечить не только точное позиционирование устройств контроля, но и согласованное функционирование всех элементов комплекса: устройств позиционирования, программирования, тестирования, системы управления и интерфейса оператора.

Ключевую роль в составе такого комплекса играет система управления, обеспечивающая координатное перемещение рабочего органа, выполнение операций программирования и тестирования, а также сбор и отображение результатов проверки. Кроме того, к основным требованиям, предъявляемым к

					ПГУ 1.090301.09.001 ПЗ	Лист
						6
Изм.	Лист	№ докум.	Подпись	Дата		

таким системам, так же относятся быстродействие, надёжность, использование типовых технических решений и сокращение времени разработки. Выполнение последних требований способствует повышению эффективности после внедрения комплекса в производственный процесс.

В результате анализа проблемы был сделан выбор, что разработка роботизированного комплекса автоматического программирования и тестирования с целевым применением для электронных модулей на печатных платах является актуальной задачей, и может быть реализована на базе трехкоординатного манипулятора, на современной элементной базе, и с использованием МК и ПО широкого применения.

Целью дипломной работы является разработка такого роботизированного комплекса автоматического программирования и тестирования электронных модулей на печатных платах.

					ПГУ 1.090301.09.001 ПЗ	Лист
						7
Изм.	Лист	№ докум.	Подпись	Дата		

1 Анализ предметной области

В современной промышленности автоматизация производственных процессов является одним из ключевых факторов повышения эффективности производства. Под автоматизацией понимается применение технических, программных и организационных средств, обеспечивающих выполнение производственных, измерительных, транспортных и контрольных операций по заранее заданному алгоритму с частичным или минимальным участием оператора. Основная цель автоматизации заключается в повышении эффективности производства, снижении трудоёмкости технологических операций, обеспечении стабильности качества выпускаемой продукции и увеличении общей производительности предприятия. В условиях современного промышленного производства автоматизация перестала быть лишь средством технического совершенствования отдельных участков и превратилась в необходимый фактор обеспечения конкурентоспособности предприятия.

Наиболее широкое применение автоматизация получила в серийном и крупносерийном производстве, где технологический процесс характеризуется многократным повторением однотипных операций, и высокой интенсивностью выпуска продукции. В таких условиях ручное выполнение операций становится экономически нецелесообразным, поскольку требует значительных затрат времени, большого числа обслуживающего персонала и не позволяет в полной мере обеспечить требуемую точность и повторяемость результата. Автоматизация позволяет исключить или существенно сократить влияние человеческого фактора, снизить вероятность случайных ошибок, уменьшить утомляемость персонала и обеспечить одинаковое качество выполнения операций на протяжении длительного времени.

Польза автоматизации в промышленности заключается не только в ускорении выполнения операций. Ее внедрение обеспечивает комплексный положительный эффект: сокращение времени производственного цикла, повышение точности обработки и измерений, снижение количества брака,

					ПГУ 1.090301.09.001 ПЗ	Лист
						8
Изм.	Лист	№ докум.	Подпись	Дата		

уменьшение затрат на исправление дефектов, улучшение условий труда персонала и повышение достоверности результатов контроля. Кроме того, автоматизированные системы позволяют организовать регистрацию, хранение и обработку результатов в цифровом виде, что особенно важно для анализа качества продукции, отслеживание технологических операций и документирования результатов испытаний.

В серийном и крупносерийном производстве радиоэлектронных модулей одной из наиболее важных операций является контроль электропараметров электронных модулей на печатных платах. На данном этапе необходимо убедиться, что изделие соответствует установленным техническим требованиям, а его основные характеристики находятся в допустимых пределах. Метод контроля, при котором оператор вручную подключает измерительное оборудование, подаёт тестовые сигналы, снимает показания с контрольных точек, сравнивает полученные значения с нормативными и оформляет результаты проверки, при массовом производстве становится менее эффективным, несмотря на его простоту. Это связано с высокой трудоёмкостью, значительными временными затратами на выполнение однотипных операций, зависимостью результата от квалификации и внимательности оператора, а также с вероятностью ошибок при измерении, интерпретации данных и заполнении протоколов.

При увеличении объёмов выпуска продукции недостатки ручного контроля становятся особенно заметными. Даже незначительное увеличение времени проверки одного изделия приводит к существенному росту суммарных затрат времени на всю партию. Кроме того, необходимость постоянного участия оператора ограничивает производительность участка контроля и затрудняет обеспечение одинаковых условий проверки для всех изделий. В результате снижается экономическая эффективность производства, возрастает риск пропуска дефектов, а также усложняется процесс документирования и анализа результатов испытаний.

В связи с этим возникает необходимость применения средств автоматизации именно на этапе контроля радиоэлектронных модулей. Для решения данной задачи используются АПК, способные автоматически выполнять подачу тестовых сигналов, считывание параметров, сравнение полученных результатов с эталонными значениями и формирование протокола проверки. Использование таких комплексов позволяет существенно повысить производительность контроля, обеспечить повторяемость условий испытаний, уменьшить влияние человеческого фактора и повысить достоверность результатов. Дополнительным преимуществом является возможность интеграции контрольного оборудования с вычислительными средствами, обеспечивающими обработку, визуализацию и сохранение данных.

1.1 Анализ существующих технологических решений

Для проверки функционально насыщенных крупногабаритных плат, имеющих сложную схемотехнику, чаще всего применяют системы контроля с использованием так называемых «летающих пробников» (см. рисунок 1). Данный комплекс имеет возможность гибкого перепрограммирования под очередной модуль, в том числе другого типа, и имеет широкие возможности по формированию тестовых сигналов. Однако скорость тестирования здесь принципиально мала, требуется точная механическая калибровка. Поэтому чаще всего такие комплексы применяют для поиска неисправностей на печатной плате, извлеченной уже после выходного контроля всего изделия.

Модули на платах средних и малых размеров, на которых при разработке минимизировано количество контрольных точек, можно тестировать в полуавтоматическом комплексе (см. рисунок 2). Основным элементом здесь является матрица с контактными иглами и зажимным устройством.

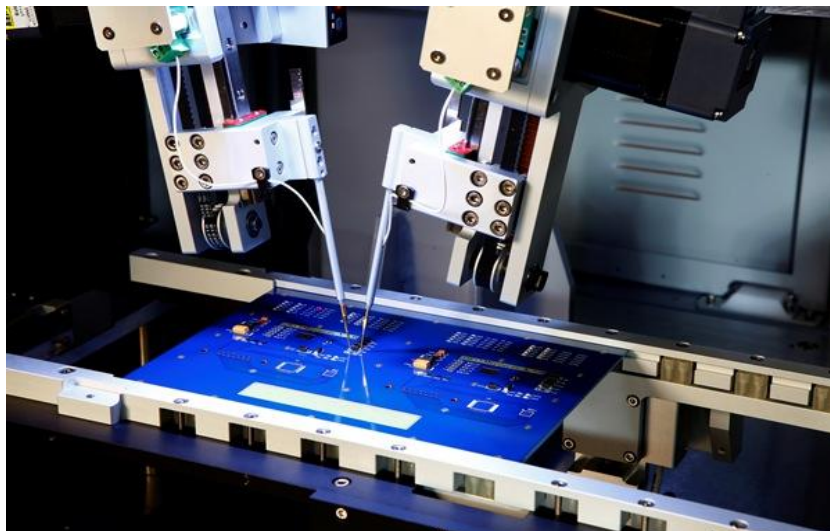


Рисунок 1 – АПК типа «Летающий пробник»

Оператор устанавливает тестируемую плату на матрицу с подпружиненными иглами так, чтобы контактные площадки модуля попали на контактные иглы оснастки. Далее он закрепляет модуль, тем самым прижимая его к иглам. В конце операции, с помощью специализированного тестера или ПК с внешним модулем тестирования, производится проверка модуля.

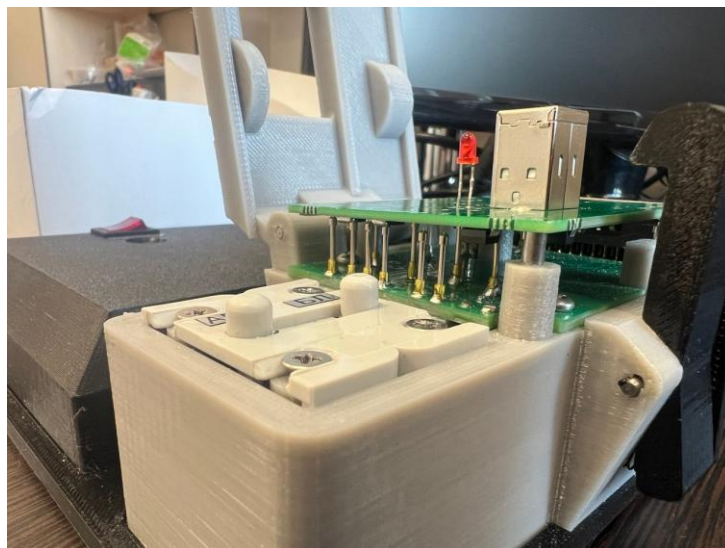


Рисунок 2 – Полуавтоматический комплекс с игольчатой матрицей

Данный вид контроля, безусловно, экономит время, так как исключается подача оператором сигналов, снятие показаний и т. д. Но подключение очередного модуля производится вручную, причем каждого по очереди из всей партии,

которая при серийном производстве может достигать больших объемов. Таким способом практически невозможно тестировать групповые сборки электронных модулей в виде панелей, так как контактное поле слишком велико и требует многонаправленного коммутатора.

Для более наглядного сопоставления рассмотренных решений целесообразно представить их основные преимущества и ограничения в виде таблицы 1.

Таблица 1 – Сравнительный анализ существующих технологических решений

Аналог	Основные возможности	Ограничения для поставленной задачи
АПК типа «летающий пробник»	Гибко перепрограммируется под разные платы, подходит для сложных модулей и позволяет формировать тестовые сигналы	Имеет малую скорость проверки, требует точной калибровки и не рассчитан на потоковый групповой контроль панелей
Полуавтоматический комплекс с игольчатой матрицей	Быстрое подключение и не требует подачи тестовых сигналов	Требуется ручной установки каждой платы, плохо подходит для групповых сборок

Таким образом, анализ автоматических аппаратов контроля электронных модулей на печатных платах показал, что существующие решения не в полной мере удовлетворяют требованиям достижения максимального экономического эффекта при среднесерийном и крупносерийном производстве. Следовательно, существует необходимость в разработке более эффективного автоматизированного комплекса.

1.2 Анализ актуальности разработки

В среднесерийном и крупносерийном производстве, как правило, однотипные платы изготавливаются не единичными экземплярами, а в составе групповых сборок (см. рисунок 3), что позволяет повысить эффективность на линиях поверхностного монтажа. В этих условиях, как выше сказано, применение поочередного контроля отдельных модулей становится недостаточно

эффективным, поскольку увеличивает трудоемкость операций и снижает общую производительность.



Рисунок 3 – Групповая сборка плат

Поэтому при серийном и особенно крупносерийном производстве радиоэлектронных модулей средней сложности на печатных платах, разумным видится групповой контроль максимально большого количества однотипных модулей без очередной установки и снятия их с игольчатой матрицы.

В связи с этим возникает необходимость создания автоматизированного комплекса, обеспечивающего групповой контроль однотипных модулей. Разработка такого комплекса направлена на повышение эффективности контроля, снижение трудоёмкости операций и улучшение качества выпускаемой продукции.

1.3 Постановка задачи

Комплекс создается с целью автоматизации производственных процессов контроля электропараметров и функционирования электронных модулей, при которой достигается максимально возможные повышение производительности, снижение трудоемкости работ при данных процессах и повышение качества конечной продукции.

Комплекс должен производить поточный контроль электропараметров и функционирования электронных модулей на печатных платах, собранных в единую панель (до ее разделения), посредством:

- автоматического позиционирования контактной игольчатой матрицы по трем координатам, под управлением ЧПУ-контроллером на основе программы, написанной на G-кодах. G-код – условное именование языка программирования устройств с числовым программным управлением (ЧПУ);
- последовательного выполнения операций для каждой платы: подключение к контактным площадкам, запуск программирования, запуск тестирования, сохранение и выдачу результатов проверки;
- автоматической обработки нештатных ситуаций в процессе работы.

Комплекс должен обеспечивать:

- минимальный объем ручных операций, ограничивающийся в основном запуском начала тестирования, посредством нажатия кнопки, установка и снятием панелей с электронными модулями;
- последовательное перемещение контактного тестирующего устройства от модуля к модулю и подключение контактными иглами к очередному модулю;
- программирование электронного модуля;
- тестирование электропараметров и функций электронного модуля и выдачу данных о результатах тестирования;
- защиту от повреждения модулей контактными иглами при сбоях программы манипулирования;
- сигнализацию оператору об окончании процесса и выдачу протокола проверки.

В составе контактного тестирующего устройства должен быть программатор для конкретного типа электронного модуля.

В составе контактного тестирующего устройства должен быть тестер для конкретного типа электронного модуля.

1.4 Определение общего состава комплекса

Для реализации автоматического программирования и тестирования электронных модулей на печатных платах в составе разрабатываемого комплекса необходимо предусмотреть ряд функциональных устройств, каждое из которых обеспечивает выполнение определённого этапа технологического процесса.

Комплекс должен содержать координатное исполнительное устройство, обеспечивающее автоматическое перемещение контактного тестирующего устройства между проверяемыми модулями и его точное позиционирование относительно контактных площадок платы. Для выполнения данной функции необходим трёхкоординатный линейный робот-манипулятор.

Одной из основных операций комплекса является программирование электронных модулей, в состав контактного тестирующего устройства должен входить программатор.

Для контроля электропараметров электронных модулей и проверки их соответствия заданным значениями в составе комплекса необходим тестер.

Для согласования работы всех узлов комплекса требуется отдельный управляющий МК, координирующий последовательность операций, обмен сигналами между исполнительными и контрольными элементами, обработку состояний системы и выдачу информации оператору.

1.5 Технические требования

Технические требования к манипулятору:

- тип: трёхкоординатный линейный робот-манипулятор;
- точность позиционирования (повторяемость): ± 50 микрон;
- управление движением с помощью ЧПУ-контроллера;
- диапазон перемещений манипулятора в пределах рабочей области 400х600х100 мм.

Технические требования к управляющему МК:

- тактовая частота 16 МГц;
- количество дискретных входов/выводов не менее 25;
- количество аналоговых входов не менее 2;
- объём флеш-памяти не менее 128 КБ;
- объём ОЗУ не менее 8 КБ.

Технические требования к программатору:

– программатор должен поддерживать программирование МК семейства ATmega;

– программатор должен иметь возможность автоматического запуска по внешнему сигналу от управляющего МК и считывания данных о состоянии программирования.

Технические требования к контактной игольчатой матрицы:

- количество игл в матрице: не менее 15;
- диаметр игл от 1мм до 3 мм;
- длина игл не менее 20 мм;
- подпружиненный ход игл от 4 до 6 мм;
- габариты не более 120х120х160мм;
- масса в сборе не более 2кг.

Технические требования к электропитанию:

– для тестера должен быть применен отдельный источник питания, с подачей постоянного напряжения 20-24 В;

– МК и программатора постоянное напряжение 5 В;

– ЧПУ-контроллера постоянное напряжение 12 В;

– драйверов шаговых двигателей манипулятора от сети 220 В переменного тока.

Необходимо использовать несколько программ управления:

- ПО общего управления системой;
- ПО трехосевого манипулятора;

					ПГУ 1.090301.09.001 ПЗ	Лист
						16
Изм.	Лист	№ докум.	Подпись	Дата		

- ПО тестера.

Дополнительные требования к комплексу:

- наличие светозвуковой индикации;
- применение не менее 5 USB-накопителей типа micro SSD;
- наличие LCD-дисплея для вывода информации о ходе проверки и результатах.

1.6 Выбор и настройка средств разработки

Одним из этапов разработки комплекса является выбор и настройка средств разработки, используемых для проектирования электронных узлов, создания общего управляющего ПО и подготовки программ управления движением манипулятора. От выбранных средств зависят эффективность выполнения проектных работ, минимизация времени на отладку и возможность оперативного внесения изменений в разрабатываемую систему. В связи с этим необходимо произвести поиск и обосновать выбор средств разработки, применённых при создании комплекса.

В результате проделанной работы были выбраны и установлены следующие средства разработки:

1.6.1 Система автоматизированного проектирования DipTrace

Для проектирования электронных узлов комплекса была выбрана и установлена САПР DipTrace. Выбор данной САПР обусловлен тем, что она позволяет выполнять основные этапы разработки электронной части устройства в одной программной среде: создавать схемную документацию, разрабатывать печатные платы, формировать библиотеки компонентов и подготавливать файлы для изготовления плат [1]. Дополнительным преимуществом является то, что DipTrace используется и другими инженерами предприятия, что упрощает обмен проектной документацией, совместное использование библиотек и работу с файлами единого формата.

Перед началом работы с САПР DipTrace необходимо выполнить её установку на ПК [2]. Установка приложения представлена на рисунках 4-8.

ПРОГРАММА	ЗАГРУЗИТЬ
DipTrace 4.3.0.6 (Win/Linux)	 32-BIT  64-BIT

Рисунок 4 – Выбор установочного файла на официальном сайте



Рисунок 5 – Окно принятия лицензионного соглашения



Рисунок 6 – Выбор каталога установки

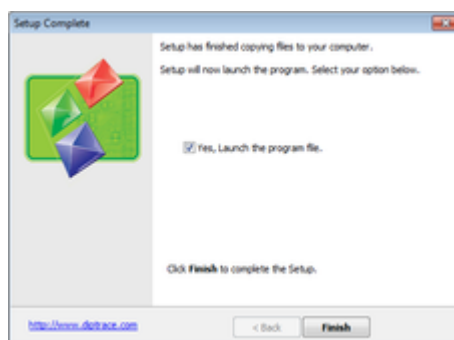


Рисунок 7 – Завершение установки

Изм.	Лист	№ докум.	Подпись	Дата

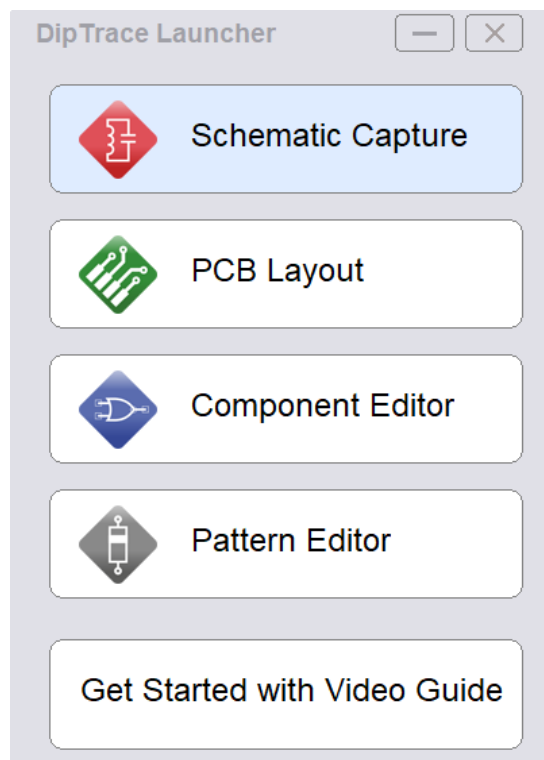


Рисунок 8 – Главное окно САПР DipTrace

Работа в САПР DipTrace начинается с редактора схем, в котором выполняется создание электрических схем устройства. Программа позволяет размещать компоненты, соединять их электрическими цепями, задавать обозначения элементов и формировать структуру будущего электронного узла. Благодаря наглядному графическому представлению соединений упрощается проверка схемных решений и снижается вероятность ошибок на этапе проектирования. Главное окно редактора схем показано на рисунке 9.

САПР DipTrace предоставляет разработчику редактор компонентов, который предназначен для создания и редактирования условных графических обозначений элементов. Это удобно в тех случаях, когда требуемый компонент отсутствует в стандартном наборе библиотек или необходимо использовать собственное обозначение, соответствующее особенностям разрабатываемого устройства. Окно редактора компонентов показано на рисунке 10.

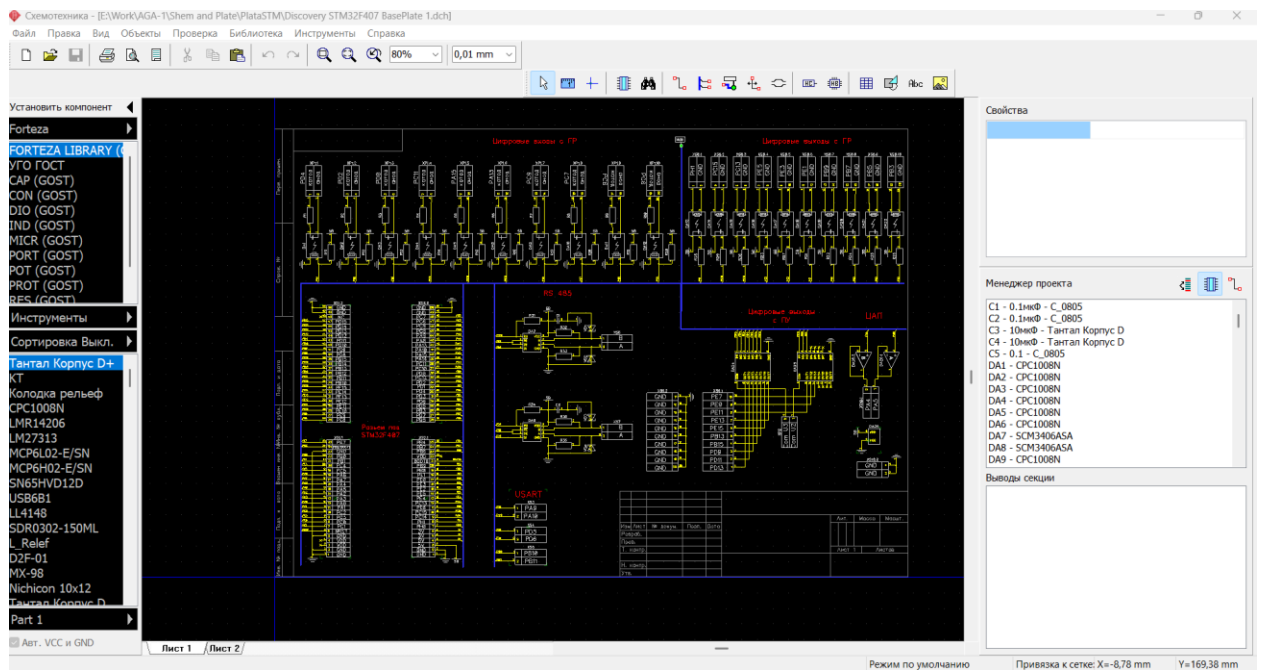


Рисунок 9 – Главное окно редактора схем

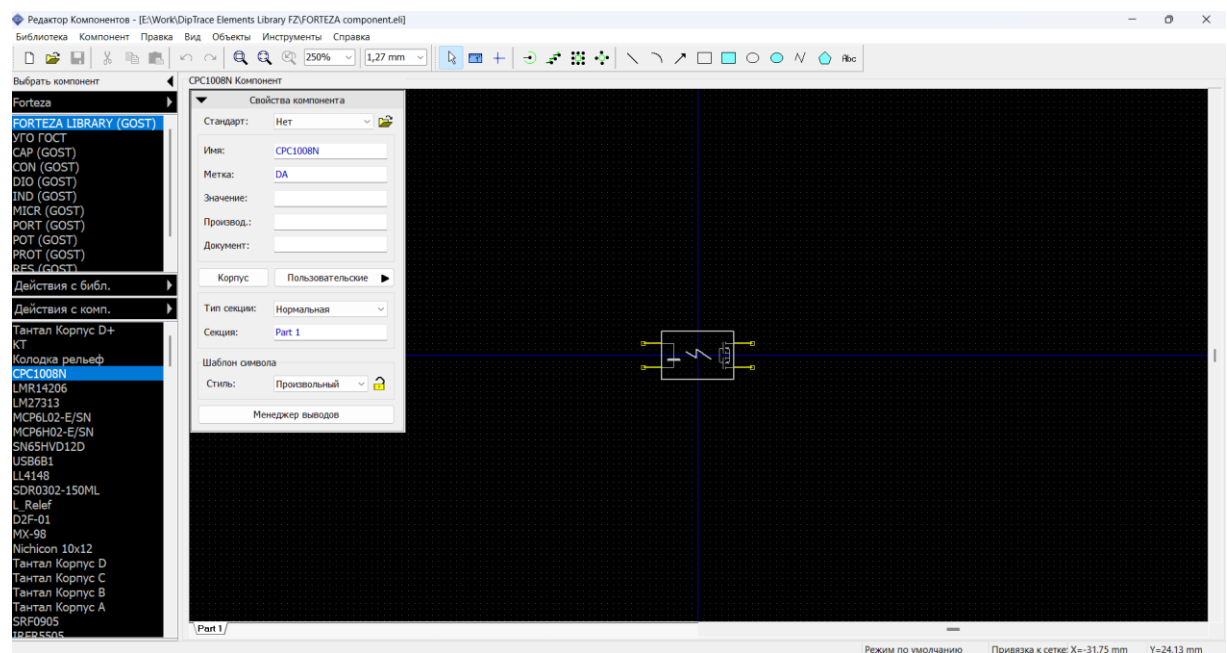


Рисунок 10 – Главное окно редактора компонентов

Помимо условных графических обозначений компонентов, в DipTrace можно создавать и редактировать их посадочные места. Это особенно важно при разработке печатных плат, поскольку корректно выбранное или созданное посадочное место обеспечивает правильное размещение элемента на плате. Возможность самостоятельного создания посадочных мест позволяет

адаптировать проект под применяемую элементную базу и конструктивные особенности разрабатываемого узла. Окно редактора посадочных мест показано на рисунке 11.

После завершения разработки электрической принципиальной схемы выполняется переход к этапу проектирования печатной платы. В редакторе печатных плат осуществляется размещение компонентов, задание геометрии платы, трассировка электрических цепей. Окно редактора печатных плат представлено на рисунке 12.

Использование одной САПР DipTrace, позволяет выполнять разработку схемы и платы в одном ПО, что делает процесс проектирования более удобным и снижает вероятность ошибок.

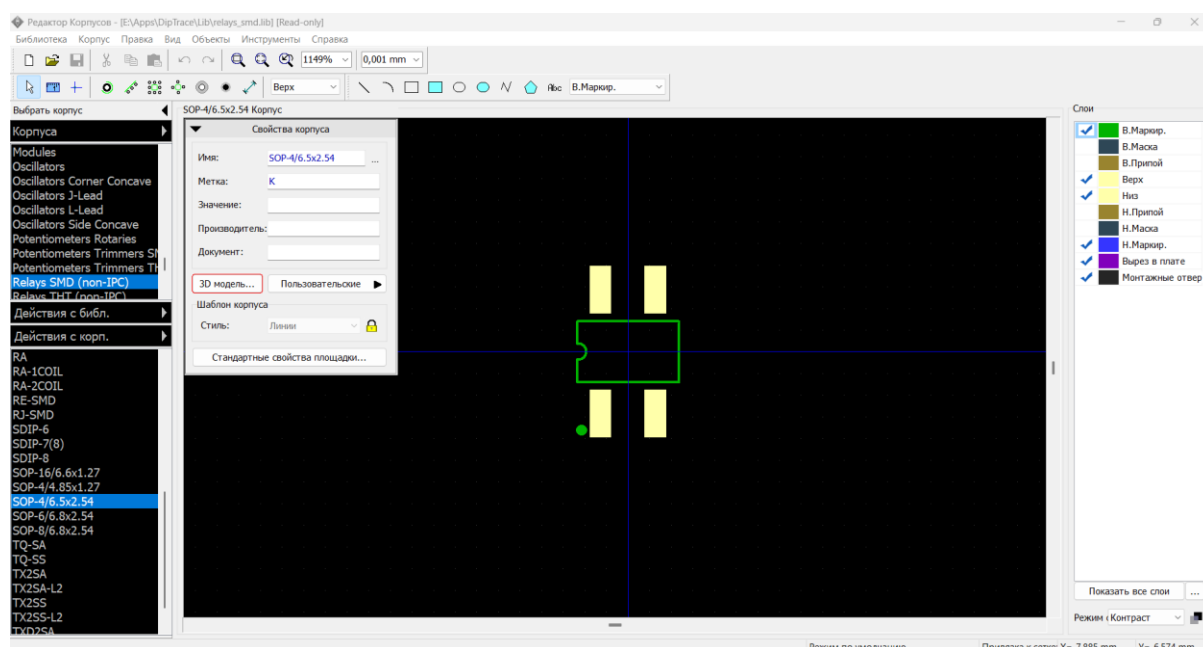


Рисунок 11 – Главное окно редактора посадочных мест

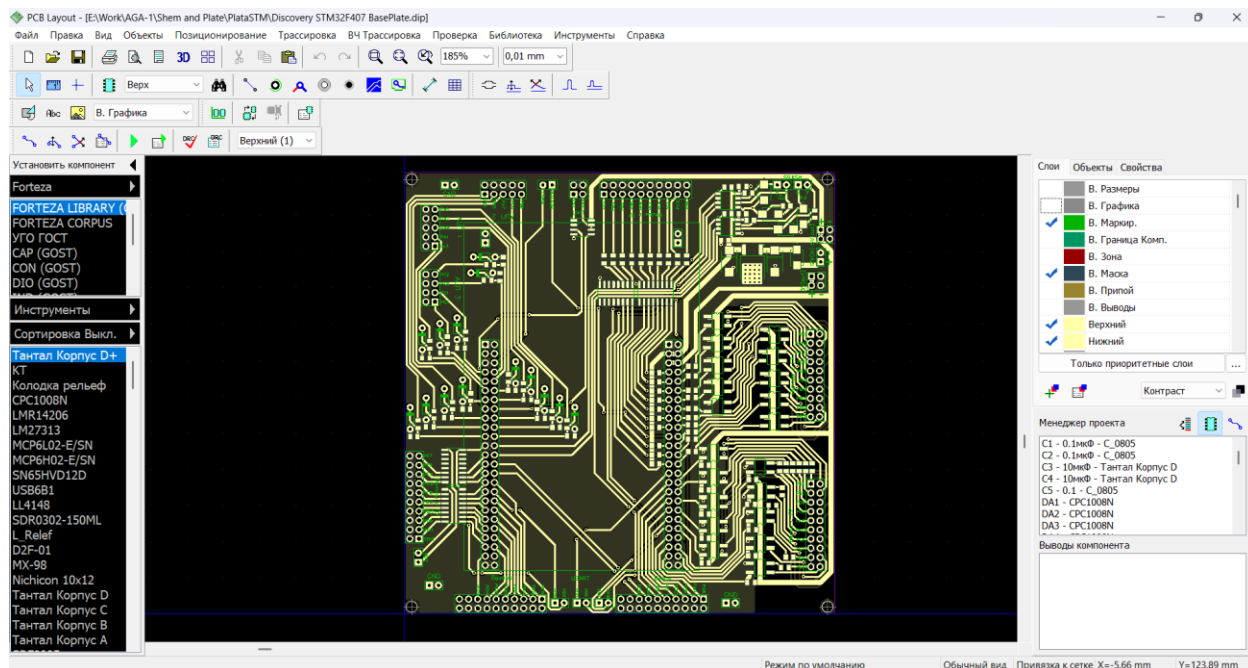


Рисунок 12 – Главное окно редактора печатных плат

1.6.2 Среда разработки Arduino IDE

Для разработки ПО управляющего МК была выбрана среда Arduino IDE. Данная среда использовалась для создания, редактирования, компиляции и загрузки основного алгоритма в отладочную плату Arduino Mega 2560. Выбор Arduino IDE обусловлен тем, что эта среда ориентирована на работу с платами семейства Arduino, отличается простотой освоения и содержит все основные средства, необходимые для разработки встроенного ПО.

Программирование в Arduino IDE осуществляется на языке C/C++ с использованием стандартных библиотек Arduino.

Для начала работы со средой Arduino IDE необходимо установить её на ПК [3]. Установка приложения представлена на рисунках 13-17.

Работа в Arduino IDE осуществляется в виде создания проекта, называемого скетчем, показанным на рисунке 18.

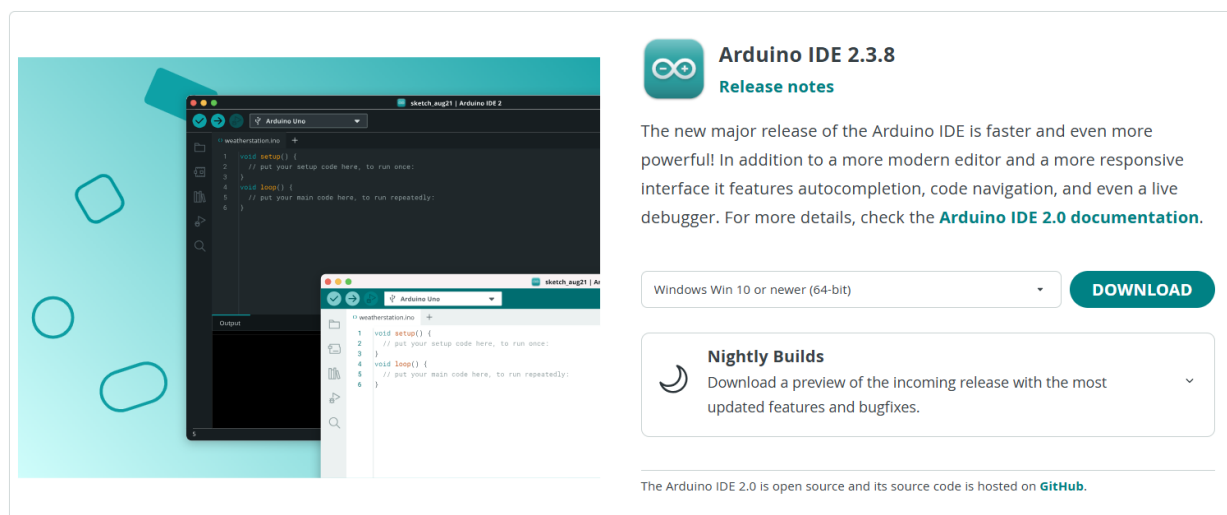


Рисунок 13 – Выбор установочного файла на официальном сайте

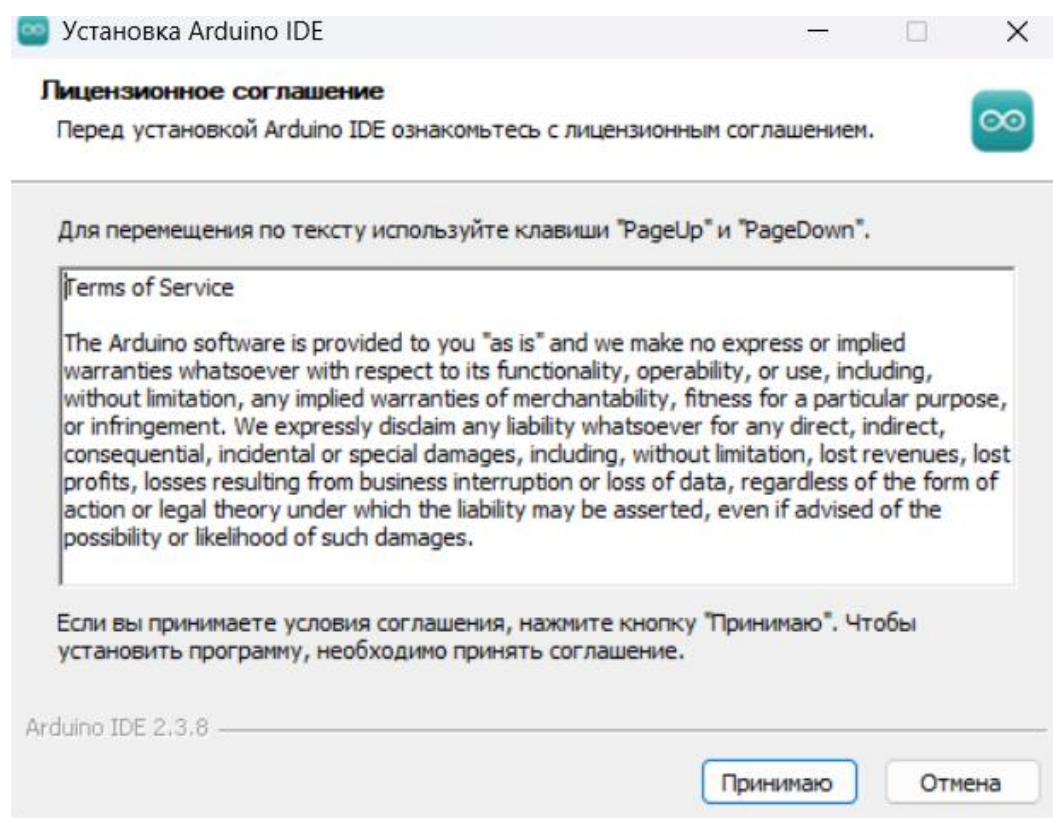


Рисунок 14 – Окно принятия лицензионного соглашения

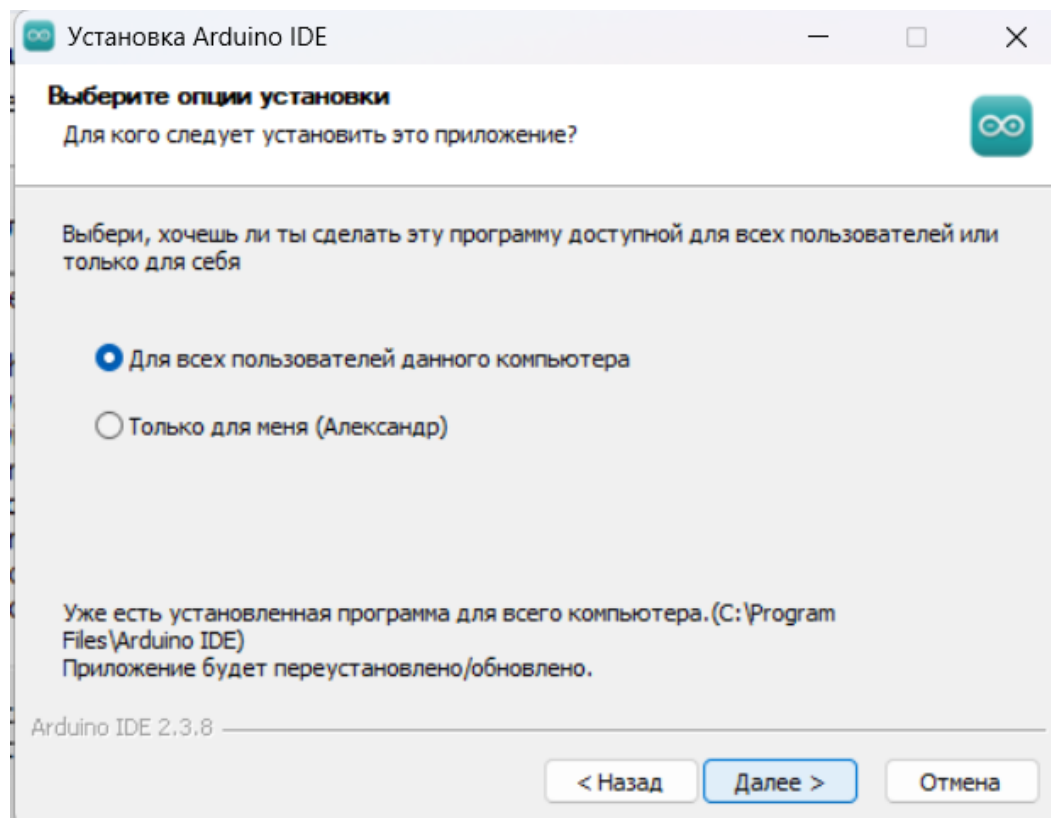


Рисунок 15 – Выбор опции установки

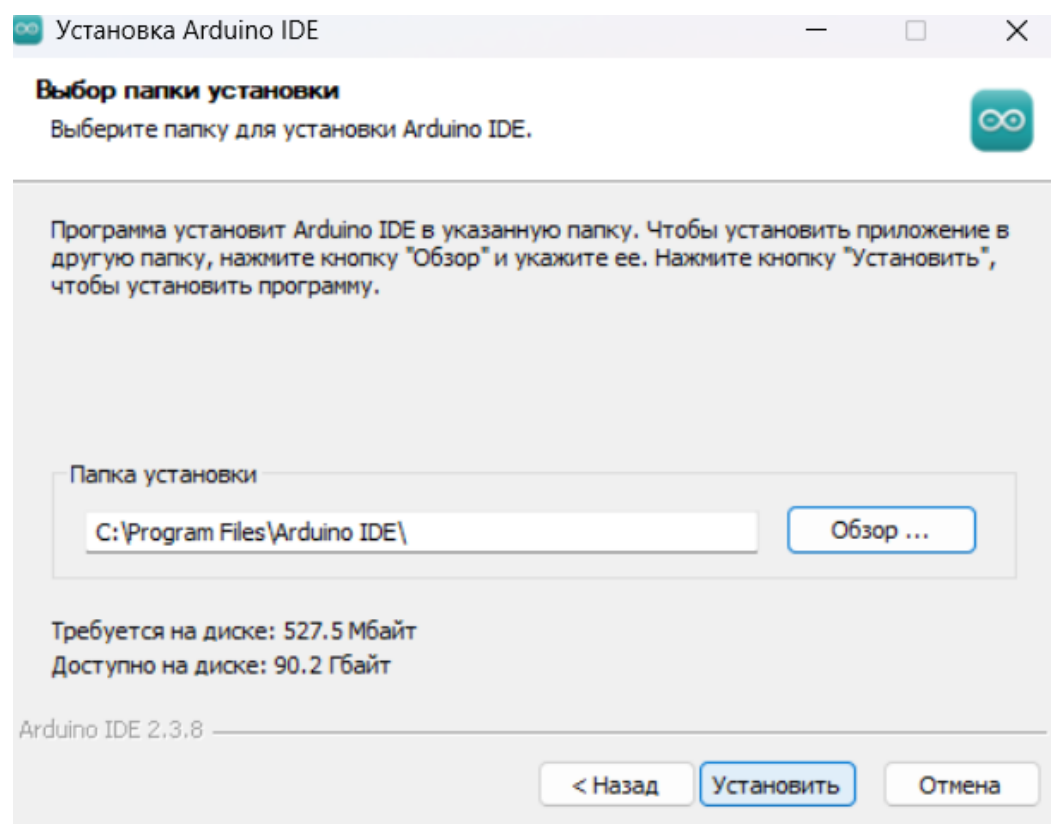


Рисунок 16 – Выбор каталога установки

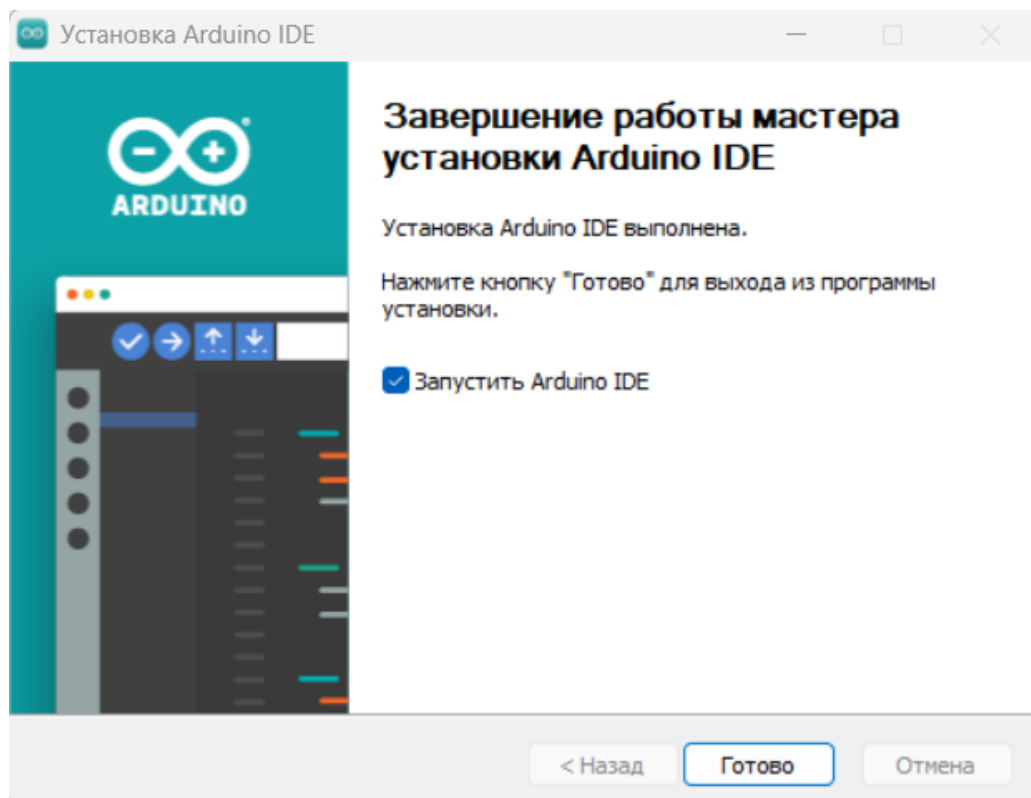


Рисунок 17 – Завершение установки

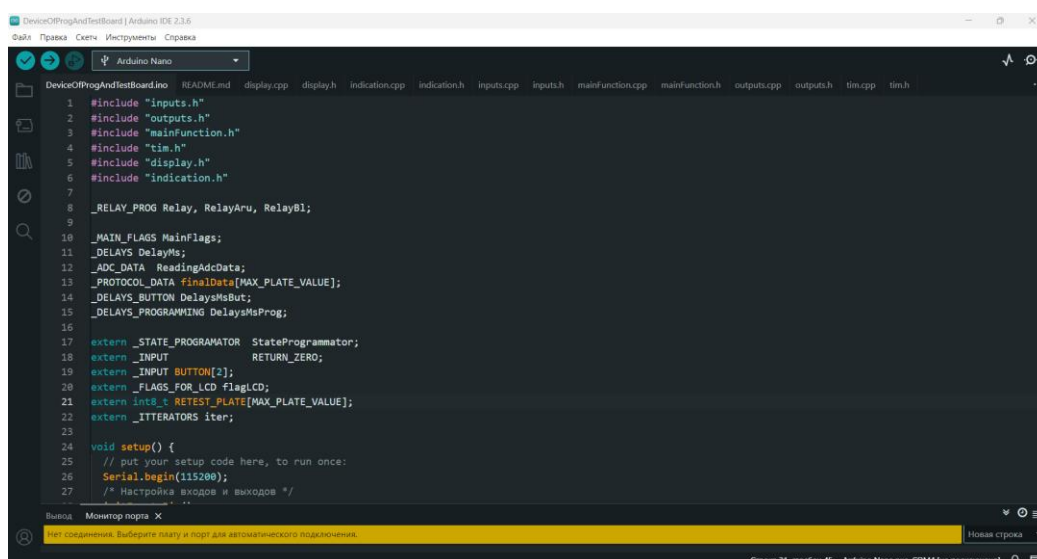


Рисунок 18 – Главное окно Arduino IDE с открытым скетчем

Перед загрузкой программы в МК в Arduino IDE необходимо выполнить настройку параметров подключения. Для этого в среде выбирается используемая плата, в данном случае Arduino Mega 2560, а также COM-порт, к которому она подключена. Корректная настройка этих параметров необходима для

последующей компиляции и загрузки программы в МК. Выбор платы и COM-порта в среде Arduino IDE показан на рисунке 19.

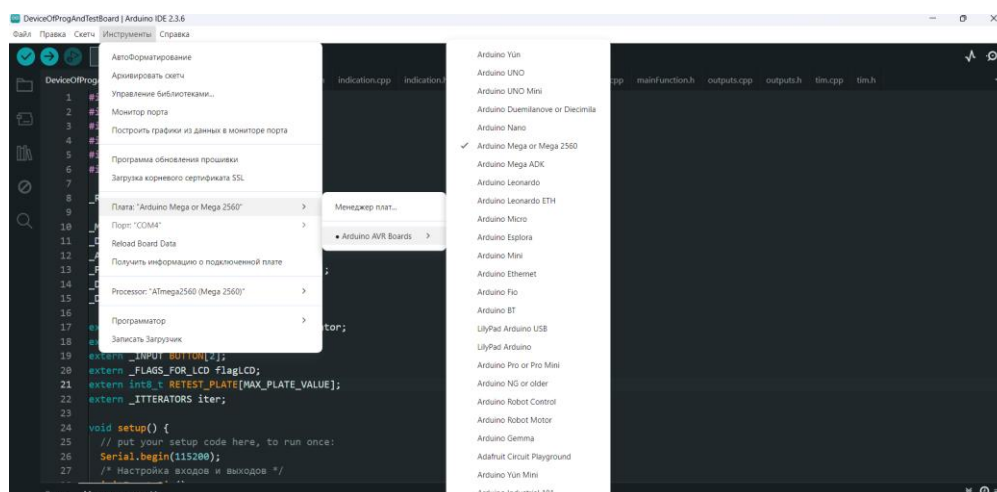


Рисунок 19 – Выбор платы и COM-порта в среде Arduino IDE

После завершения написания программы и настройки параметров подключения в Arduino IDE выполняется компиляция скетча и его загрузка в МК, посредством нажатия на кнопку запуска данных процессов (см. рисунок 20). Результат успешной компиляции и загрузки показан на рисунке 21.

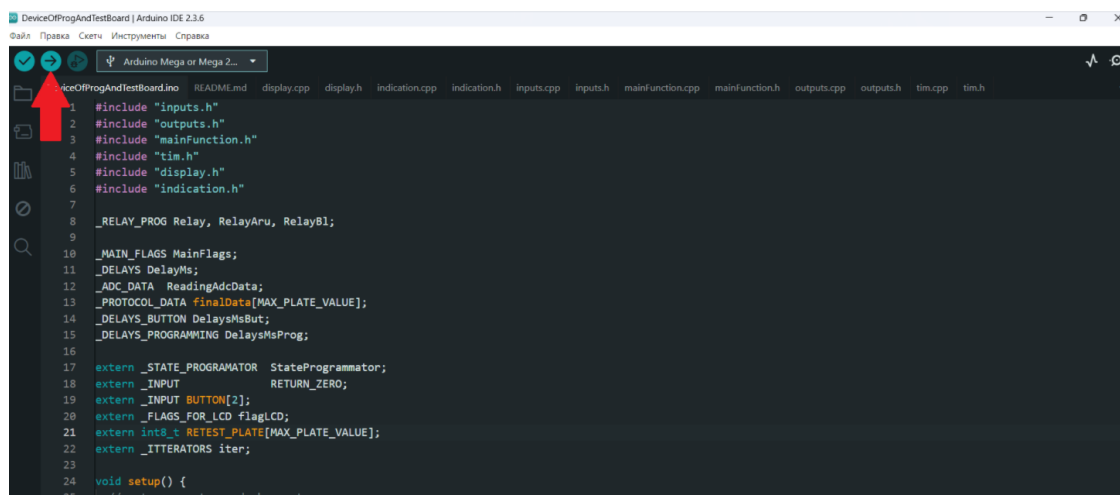


Рисунок 20 – Запуск компиляции и загрузки программы в МК

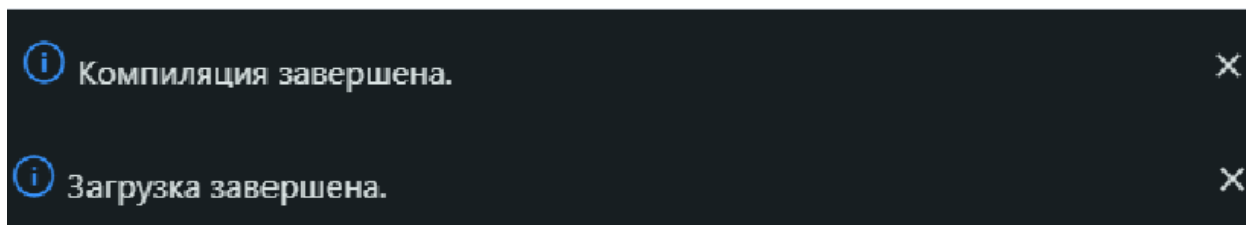


Рисунок 21 – Результат успешной компиляции и загрузки программы в МК

1.6.3 Стандартный текстовый редактор Windows «Блокнот»

Для подготовки программ управления манипулятором, использовался стандартный текстовый редактор Windows «Блокнот». Выбор данного средства обусловлен тем, что программа для ЧПУ-контроллера представляет собой обычный текстовый файл, содержащий последовательность команд на G- и M-кодах [4, 5]. Поэтому применение более сложных программных средств в данном случае не требуется.

В текстовом редакторе формировалась управляющая программа, определяющая перемещения манипулятора по заданным координатам, а также выполнение вспомогательных команд, связанных с работой комплекса. После создания или редактирования файл сохранялся на USB-накопитель microSD, который затем устанавливался в ЧПУ-контроллер для выполнения программы.

Использование редактора «Блокнот» является удобным и достаточным решением, поскольку позволяет быстро создавать, изменять и сохранять текст управляющей программы без применения специализированного ПО. Пример написания программы на G- и M-кодах в текстовом редакторе Windows «Блокнот» представлен на рисунке 22.

```

G04P500
M211
M210
M213
G04P500
M111
M110
M113
G04P500
G01 X65.06181 Y-24.01168 F2000 (A1)
G01 Z9 F1500
G04P500
G91 (Переход к относительным координатам)
G01 Z-9 F150 (Опускание головы к контактным площадкам)
G04P200
M214
G04P200
M114
M414 ("Ждем программирования и тестирования")
G01 Z9 F700 (Поднимаем на высоту перемещения между платами)
G01 X36.15 F1000
G01 Z-9 F150
G04P200
M214
G04P200
M114
.....

```

Строка 1, столбец 1 | 3 519 символов | Обычный текст | 100% | Windows (CRLF) | UTF-8

Рисунок 22 – Программа на G- и M-кодах в текстовом редакторе Windows «Блокнот»

2 Разработка комплекса

Для реализации АПК автоматического программирования и тестирования электронных модулей на печатных платах необходимо было определить состав его аппаратной и программной частей. Выбор соответствующих аппаратных средств осуществлялся с учетом функциональных требований к комплексу, особенностей технологического процесса, необходимости обеспечения надежной работы, удобства настройки и экономической целесообразности применения.

Аппаратная часть комплекса должна обеспечивать выполнение основных технологических операций, связанных с позиционированием рабочего органа, программированием и тестированием электронных модулей, а также согласованной работой всех исполнительных устройств.

2.1 Выбор аппаратных частей комплекса

2.1.1 Выбор трехкоординатного робот-манипулятора

Одним из основных элементов разрабатываемого комплекса является трёхкоординатный робот-манипулятор, предназначенный для последовательного автоматического перемещения игольчатого контактного устройства в пределах рабочей зоны и его точного позиционирования относительно электронных модулей, а именно его контактных площадок, предназначенных для контроля. Необходимость применения именно такого исполнительного механизма обусловлена тем, что комплекс должен автоматически подводить иглы контактного устройства к каждой проверяемой плате, обеспечивать их опускание на контактные площадки модуля и последующий подъём перед переходом к следующей плате.

Для решения этой задачи в качестве базовой механической платформы был выбран трёхкоординатный ЧПУ фрезер CNC-4060AL3 (см. рисунок 23). Использование данного устройства обусловлено тем, что оно изначально представляет собой линейную трёхкоординатную систему перемещения, обеспечивающую движение рабочего органа по осям X, Y и Z в пределах заданной

рабочей зоны. При этом для применения в составе разрабатываемого комплекса штатная фрезерная головка была демонтирована, а на её место установлено специально разработанное контактное устройство с игольчатой матрицей, предназначенное для программирования и тестирования электронных модулей.



Рисунок 23 – Фрезер CNC-4060AL3

Существенным аргументом в пользу выбора CNC-4060AL3 является возможность его адаптации под задачи автоматизированного комплекса без разработки механической части с нуля. Использование готовой координатной платформы позволило сократить сроки разработки, упростить сборку системы и получить исполнительный механизм, уже обладающий необходимой кинематикой для точного позиционирования контактного устройства над электронными модулями.

Дополнительным основанием в пользу такого выбора являются и заданные технические требования. Манипулятор должен обеспечивать перемещение в пределах рабочей области $400 \times 600 \times 100$ мм, иметь повторяемость позиционирования порядка ± 50 мкм и работать под управлением ЧПУ-контроллера. Эти требования соответствуют характеру выполняемой задачи, так как для надёжного подключения игл к контактным площадкам необходимо достаточно точное и воспроизводимое подведение рабочего органа к каждой позиции.

2.1.2 Выбор ЧПУ-контроллера движения

Для управления перемещением выбранного манипулятора в составе комплекса необходим специализированный контроллер движения, обеспечивающий формирование команд перемещения по координатам, выполнение управляющей программы и взаимодействие с другими узлами системы. При разработке комплекса рассматривался вариант реализации управления движением непосредственно с помощью МК. Однако такой подход потребовал бы формирования импульсных последовательностей для шаговых двигателей, учёта параметров драйверов и реализации алгоритмов разгона и торможения, что существенно усложнило бы программную часть системы и увеличило бы время разработки.

В качестве альтернативы был выбран покупной стандартизированный ЧПУ-контроллер движения SMC4-4-16A16B [6] (см. рисунок 24). Его применение позволило вынести задачи координатного управления в отдельный узел и тем самым разграничить функции системы: ЧПУ-контроллер отвечает за перемещение манипулятора, а управляющий МК – за выполнение технологических операций программирования и тестирования. Такое разделение функций упростило построение комплекса и повысило удобство его настройки.

В составе комплекса контроллер SMC4-4-16A16B используется преимущественно для управления движением манипулятора по координатам и выполнения команд, связанных с работой привода. При этом важной его функцией является участие в информационном обмене с управляющим МК. После достижения заданной позиции ЧПУ-контроллер передаёт сигнал о завершении перемещения на МК, которая запускает последующие операции программирования и тестирования. После их завершения управляющий МК формирует ответный сигнал, разрешающий переход к следующей координате. Такой принцип взаимодействия обеспечивает согласованную работу механической и технологической частей комплекса.



Рисунок 24 – ЧПУ-контроллер движения SMC4-4-16A16B

Дополнительным преимуществом является использование стандартных G- и M-кодов, что значительно упрощает разработку программ перемещения и облегчает адаптацию комплекса под особенности расположения модулей на панели.

2.1.3 Выбор программатора

Разрабатываемый комплекс предназначен не только для тестирования, но и для автоматического программирования электронных модулей, в его состав обязательно должен входить программатор. Основным требованием к нему является поддержка программирования МК того типа, которые применяются в проверяемых электронных модулях. Кроме того, программатор должен обеспечивать внутрисхемное подключение к модулю и возможность работы в составе автономного комплекса без постоянного подключения к ПК.

Выбор программатора так же обусловлен необходимостью минимизации ручных операций. В качестве него был выбран внутрисхемный отладчик/программатор PICkit 5 [7] (см. рисунок 25). Он обеспечивает надёжное внутрисхемное программирование МК без снятия электронного модуля с панели,

что особенно важно в условиях серийного производства. Однако для внедрения в состав комплекса стандартного исполнения программатора недостаточно. Для его использования в автоматическом режиме требуется доработка, обеспечивающая возможность запуска процесса программирования по сигналу, формируемому управляющим МК комплекса и считыванию сигналов, отвечающих за контроль состояния процесса программирования. Следовательно, программатор должен быть адаптирован к работе в составе автоматической системы, где запуск программирования осуществляется по команде управляющего МК, который также считывает и обрабатывает состояние данного процесса.



Рисунок 25 – Внутрисхемный отладчик/программатор PICkit 5

Таким образом, выбор внутрисхемного отладчика/программатора PICkit 5 обусловлен его совместимостью с используемыми МК, возможностью автономного применения без постоянного подключения к ПК, пригодностью для внутрисхемного программирования электронных модулей и возможностью доработки для запуска по сигналу от управляющего МК комплекса.

2.1.4 Выбор тестера

Тестер, предназначенный для проверки электрических параметров и функционирования электронных модулей после их программирования, является одним из основных компонентов аппаратной части комплекса. Его наличие обусловлено основной задачей комплекса – автоматическим контролем исправности электронных модулей на печатных платах до их дальнейшего

использования в производстве. При этом тестер должен работать в составе общего цикла, автоматически подключаться к контактным площадкам проверяемого модуля, выполнять проверку по заданной методике и обеспечивать получение информации о результатах тестирования.

В составе разрабатываемого комплекса был использован специализированный тестер, ранее разработанный и применявшийся на предприятии, для ручной проверки соответствующих электронных модулей. Выбор именно данного устройства обусловлен тем, что оно уже было адаптировано под конкретный тип проверяемых изделий и обеспечивало реализацию необходимой методики контроля. Конструктивно тестер был объединён с контактной игольчатой матрицей, что позволило использовать его в составе игольчатого контактного устройства комплекса.

Применение готового специализированного тестера вместо разработки нового тестирующего устройства или формирования всех тестовых сигналов непосредственно средствами управляющего МК позволило существенно сократить сроки разработки комплекса. В случае реализации функций тестирования только средствами управляющего МК потребовалась бы дополнительная разработка аппаратных и программных средств формирования тестовых сигналов, схем измерения и алгоритмов контроля, что заметно усложнило бы проект. Использование уже существующего тестера, прошедшего практические испытания на предприятии, позволило сосредоточиться на задаче его интеграции в состав автоматизированного комплекса.

2.1.5 Выбор управляющего МК

Для координации работы программатора, тестера и ЧПУ-контроллера в составе комплекса необходим отдельный управляющий МК, реализующий общий алгоритм функционирования системы. Он должен обеспечивать приём и выдачу дискретных сигналов, управление исполнительными элементами, обработку состояний периферийных устройств и синхронизацию всех этапов технологического цикла.

В качестве управляющего МК был выбран ATmega2560. Данный выбор обусловлен достаточным количеством цифровых и аналоговых входов и выходов, необходимым объёмом памяти и возможностью реализации алгоритма управления всеми технологическими процессами комплекса. Для практической реализации была использована отладочная плата Arduino Mega 2560 [8] (см. рисунок 26). Применение готовой отладочной платы позволило упростить разработку аппаратной части, исключить необходимость самостоятельного проектирования и изготовления отдельной микроконтроллерной платы, а также сократить сроки отладки системы. Дополнительным преимуществом является наличие удобной и простой среды разработки Arduino IDE, обеспечивающей быстрое создание, загрузку и корректировку ПО управляющего МК.

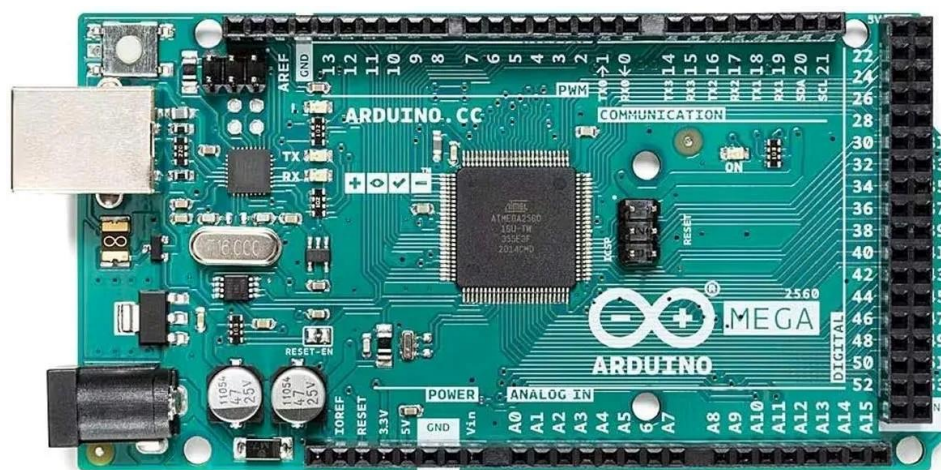


Рисунок 26 – Отладочная плата Arduino Mega 2560

Следует отметить, что использование Arduino Mega 2560 позволило логически отделить задачи общего управления комплексом от задач координатного движения. Такое разделение функций между SMC4-4-16A16B и Arduino упростило разработку, сократило время настройки и обеспечило гибкость при возможной модернизации системы. Именно поэтому использование Arduino Mega 2560 в данном проекте является не только технически достаточным, но и экономически целесообразным решением.

2.2 Разработка общей структуры комплекса

После выбора основных аппаратных компонентов была выполнена реализация роботизированного комплекса автоматического программирования и тестирования электронных модулей на печатных платах. При разработке комплекса основное внимание уделялось согласованной работе трёх основных частей: координатного манипулятора, контактного тестирующего устройства и управляющего МК. В итоге комплекс был разработан по следующим основным принципам.

Общая структура комплекса построена с разделением функций между отдельными устройствами. А именно, управление перемещением рабочего органа выполняет ЧПУ-контроллер SMC4-4-16A16B, который обрабатывает координаты перемещения по заранее подготовленной программе на G- и M-кодах. Arduino Mega 2560 осуществляет общее управление, отвечает за выполнение технологических операций: проверку правильности позиционирования, запуск программирования, запуск тестирования, считывание результатов, обработку ошибок и выдачу информации оператору. Специализированное тестирующее устройство, примененное из состава оборудования предприятия. Оно обеспечивает по собственному ПО тестирование электропараметров и функционирование электронных модулей. Схема электрическая структурная комплекса представлена в приложении А.1.

Такое разделение позволило исключить необходимость реализации сложных алгоритмов управления шаговыми двигателями непосредственно в программе управляющего МК. ЧПУ-контроллер выполняет только задачи позиционирования, а Arduino Mega 2560 управляет технологическим циклом проверки. За счёт этого была упрощена программная часть комплекса и повышено удобство его настройки.

Работа комплекса начинается после установки групповых панелей с электронными модулями на рабочую поверхность манипулятора. Оператор устанавливает панели на фиксирующие элементы, включает питание комплекса и

процесс тестирования движения. При первом включении выполняется сброс ЧПУ-контроллера общей кнопкой сброса, после чего нажимается кнопка запуска. Управляющая программа движения автоматически считывается контроллером с установленного USB-носителя. При проверке последующих панелей оператору достаточно установить новые панели и нажать кнопку запуска.

В зависимости от количества проверяемых модулей используется соответствующая программа перемещения. В начале управляющей программы формируется двухбитный код, по которому управляющий МК определяет вариант работы комплекса и количество проверяемых позиций. Это позволяет использовать один и тот же алгоритм Arduino для разных вариантов проверки, например для 20 или 40 модулей.

2.3 Разработка контактного тестирующего устройства

Для подключения к проверяемым электронным платам было разработано контактное тестирующее устройство, устанавливаемое на подвижную часть трёхкоординатного манипулятора. Данное устройство выполняет функции рабочего органа комплекса и обеспечивает механическое и электрическое соединение с контактными площадками проверяемой платы.

В состав контактного тестирующего устройства входят игольчатая контактная матрица, доработанный программатор, специализированный тестер и элементы коммутации. Игольчатая матрица содержит 29 подпружиненных контактных игл, которые при опускании рабочего органа подключаются к контактными площадкам электронного модуля. Использование подпружиненных игл позволяет компенсировать небольшие отклонения по высоте и обеспечить надёжный контакт с платой.

Особенностью реализации контактного устройства является то, что часть контактных точек используется как при программировании, так и при последующем тестировании модуля. Поэтому в состав устройства были включены коммутационные реле, обеспечивающие переключение общих контактных линий между программатором и тестером. Управление переключением выполняется

управляющим МК. Благодаря этому исключается одновременное подключение программатора и тестера к одним и тем же цепям проверяемого модуля.

Точность попадания контактных игл на контактные площадки обеспечивается за счёт координатного позиционирования манипулятора по программе на G-кодах. Перед началом проверки панели устанавливаются на рабочей поверхности в заданном строго пространственном положении, а координаты каждой платы заранее задаются в управляющей программе движения. При переходе между платами в пределах одной панели используются относительные перемещения, а переход между панелями выполняется по абсолютным координатам. Такой подход упрощает подготовку программы перемещения и позволяет последовательно обрабатывать все модули в групповой сборке.

Для контроля корректности контакта после опускания игольчатой матрицы используется сигнал с контрольной точки проверяемого модуля. Такой способ позволяет программно определить, что контактные иглы попали на необходимые площадки. Подробно данный этап рассмотрен при описании алгоритма программирования и тестирования модуля.

2.4 Доработка программатора PICkit 5 для автоматического запуска

Для программирования электронных модулей в составе роботизированного комплекса автоматического программирования и тестирования электронных модулей был использован внутрисхемный программатор PICkit 5. В штатном режиме данный программатор рассчитан на работу с участием оператора: оператор подаёт питание на программатор, устанавливает USB-накопитель с требуемой программой, запускает процесс программирования и визуально контролирует состояние устройства по светодиодной индикации.

При использовании программатора в составе автоматического комплекса такой способ работы не подходит, так как после подключения игольчатой матрицы

к очередному электронному модулю запуск программирования должен выполняться автоматически, без участия оператора. Кроме того, управляющий МК должен получать информацию о состоянии программатора и принимать решение о дальнейшем ходе технологического цикла. В связи с этим была выполнена аппаратная доработка программатора PICkit 5.

Основная задача доработки заключалась в автоматизации без изменения внутреннего алгоритма работы самого программатора. Необходимо было обеспечить автоматическую подачу питания на программатор, имитацию нажатия кнопки запуска программирования, считывание состояния светодиодной индикации, передачу полученных сигналов на входы управляющего МК Arduino Mega 2560, а также электрическое согласование и гальваническую развязку цепей программатора и управляющей системы.

Для решения данной задачи была разработана дополнительная плата согласования, представленная на рисунках 27, 28, схема электрическая принципиальная представлена в приложении А.3.

Она была разработана в САПР DipTrace и конструктивно установлена непосредственно на программатор PICkit 5. Такое размещение позволило сократить длину соединительных проводов и выполнить подключение к необходимым цепям программатора: цепи питания, контактам штатной кнопки запуска и цепям светодиодной индикации. При разработке учитывались как электрические, так и конструктивные требования. Плата была выполнена таким образом, чтобы её можно было закрепить на программаторе с использованием имеющихся технологических отверстий. Это позволило обеспечить жёсткое механическое соединение платы доработки с программатором и повысить надёжность конструкции.

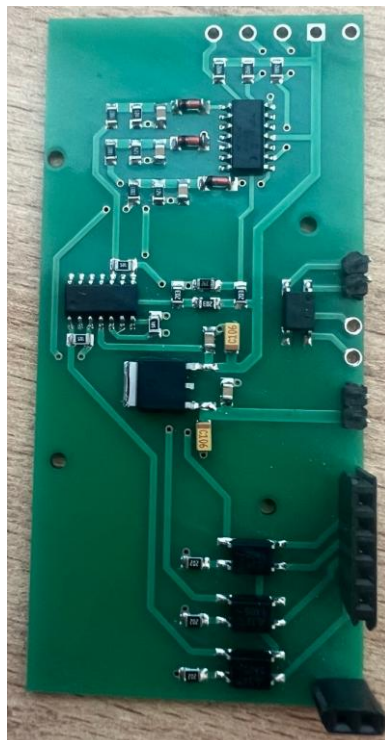


Рисунок 27 – Верхний слой платы согласования

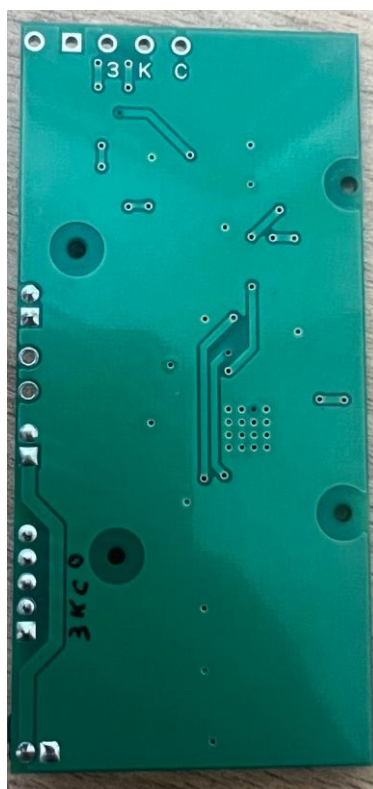


Рисунок 28 – Нижний слой платы согласования

Размер платы был выбран с учётом габаритов программатора PICkit 5 и небольшого запаса для размещения разъёмов и элементов схемы. Печатная плата выполнена двухслойной. На верхнем слое расположены основные электронные компоненты и сигнальные проводники, а нижний слой использован в качестве общего проводника GND. Такое конструктивное решение упрощает трассировку, улучшает помехоустойчивость схемы и обеспечивает более стабильную работу цепей обработки сигналов светодиодной индикации.

Питание платы согласования осуществляется напряжением 12 В, от отдельного источника питания. Для формирования напряжения 5 В на плате установлен стабилизатор LP2950CDT-5.0G [9]. Полученное напряжение 5 В используется для питания программатора, микросхем, цепей подтяжек, делителей напряжения и остальных элементов схемы согласования. Применение стабилизатора напряжения позволяет получить стабильное питание, необходимое для корректной работы как самого программатора, так и схемы обработки сигналов.

Входные и выходные цепи платы выведены на разъёмы. Разъём XP1 используется для подачи питания 12 В на плату доработки. Разъём XP3 предназначен для подачи напряжения 5 В на программатор. Разъём XP2 применяется для подключения цепи имитации кнопки запуска к управляющему МК. Контакты P1 и P2 соединяются проводами с контактами штатной кнопки запуска на плате программатора. Разъём XS2 используется для передачи сигналов состояния светодиодной индикации программатора на Arduino Mega 2560. Разъём XS1 предназначен для вывода напряжения 12 В с платы доработки при необходимости подключения дополнительных цепей.

Запуск программирования реализован путём имитации нажатия штатной кнопки программатора. Для этого к контактам кнопки параллельно подключено оптореле HSSR-S1A05L [10]. При поступлении управляющего сигнала от Arduino Mega 2560 оптореле замыкает цепь кнопки, и программатор воспринимает это как обычное нажатие. Таким образом, управляющий МК может запускать

программирование автоматически, не воздействуя механически на кнопку программатора.

Использование оптореле в цепи запуска обусловлено несколькими причинами. Во-первых, оно обеспечивает электрическую (гальваническую) развязку между управляющим МК и цепями программатора. Во-вторых, в отличие от электромагнитного реле, оптореле не имеет механических контактов, поэтому отсутствует дребезг при переключении. В-третьих, такое решение имеет меньшие габариты и удобно для размещения на небольшой печатной плате, установленной непосредственно на программаторе.

Считывание состояния программатора выполнено по сигналам его светодиодной индикации. В процессе работы PICkit 5 использует несколько режимов индикации. Зелёный светодиод соответствует нормальному состоянию программатора и готовности к работе. Красный светодиод соответствует ошибке. Синий светодиод соответствует этапу инициализации. В процессе программирования одновременно активны синий и красный каналы, что визуально воспринимается как фиолетовая индикация.

Особенность данной индикации заключается в том, что светодиоды управляются не постоянным уровнем, а импульсными сигналами. Поэтому простое подключение входов Arduino Mega 2560 к цепям светодиодов не позволило бы получить устойчивые логические уровни. Для надёжного определения состояния программатора на плате была применена особая схема обработки сигналов на микросхемах MСР604 [11].

Первая часть схемы на MСР604 используется для детектирования импульсных сигналов светодиодной индикации. Сигналы с каналов зелёного, красного и синего светодиодов поступают на входы компараторов. Далее с помощью диодов и RC-цепей выполняется преобразование пачек импульсов в более длительный сигнал, соответствующий активности определённого канала индикации. Фильтрующие цепи позволяют сгладить импульсный характер

управления светодиодами и получить сигнал, удобный для дальнейшей обработки.

После первого этапа обработки полученный сигнал может иметь не крутые фронты и небольшие пульсации на протяжении всего импульса. Для формирования устойчивого цифрового сигнала используется вторая часть схемы на МСР604, выполненная с положительной обратной связью. В таком включении компараторы работают как триггеры Шмитта. Это позволяет получить сигналы с крутыми фронтами, пригодные для передачи на входы управляющего МК.

После обработки сигналы состояния программатора через элементы опторазвязки передаются на входы Arduino Mega 2560. Благодаря этому обеспечивается согласование уровней и защита управляющего МК от возможного влияния цепей программатора. В результате Arduino получает не импульсные сигналы светодиодов, а устойчивые цифровые сигналы, по которым можно программно определить состояние PICkit 5.

После изготовления платы была выполнена её сборка и проверка работоспособности. Проверка проводилась поэтапно. Сначала контролировалось наличие питающих напряжений, затем проверялась работа цепи имитации кнопки запуска и цепей считывания светодиодной индикации программатора.

Для проверки выходных сигналов платы был использован тестовый модуль, позволяющий визуально отобразить состояние сигналов, соответствующих светодиодной индикации программатора. Модуль подключался к выходному разъёму платы согласования. Если состояние светодиодов тестового модуля совпадало с состоянием светодиодов программатора, плата считалась исправной и пригодной для установки в состав роботизированного комплекса автоматического программирования и тестирования электронных модулей. Подключение и тестирование платы согласования показано на рисунке 29.

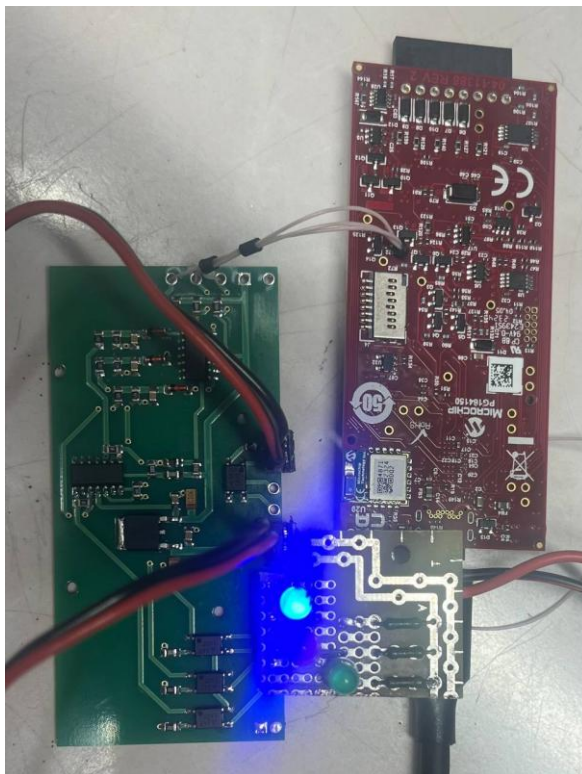


Рисунок 29 – Проверка работоспособности платы согласования

Такой способ проверки позволил убедиться, что плата доработки корректно считывает состояние PICkit 5, правильно формирует выходные сигналы для управляющего МК и обеспечивает требуемое согласование уровней. В результате была подтверждена возможность использования разработанного адаптера в составе автоматизированной системы.

2.5 Разработка системы управления комплексом на базе Arduino Mega 2560

Arduino Mega 2560 является центральным элементом системы управления технологическим процессом. Он обеспечивает связь между ЧПУ-контроллером, программатором, тестером, контактной матрицей, органами управления и средствами индикации.

Схема электрическая соединений входных и выходных сигналов Arduino Mega 2560 со всей периферией представлена в приложении А.4.

С помощью аналоговых входов выполняется контроль напряжений, необходимых для определения правильности подключения к проверяемому модулю. Цифровые входы используются для считывания сигналов состояния программатора, тестера и команд от ЧПУ-контроллера.

Выходные сигналы Arduino Mega 2560 используются для управления питанием отдельных узлов, переключения контактных линий, запуска программирования, имитации нажатия кнопок тестера, разрешения дальнейшего движения манипулятора, а также управления светозвуковой индикацией. Таким образом, Arduino Mega 2560 выполняет роль координатора всех технологических операций комплекса.

ПО управляющего МК реализовано в виде последовательного алгоритма с разделением работы на основные этапы и подэтапы. В программе выделены этапы ожидания готовности манипулятора, проверки позиции, программирования, тестирования, анализа результата, перехода к следующей плате и возврата в нулевую точку. Фрагмент кода основного алгоритма программы представлен в приложении Б.1.

Во время работы программы постоянно выполняются служебные функции, не зависящие от текущего этапа проверки. К ним относятся считывание контрольных напряжений, анализ состояния программатора, анализ состояния тестера, проверка команды возврата в нулевую точку, обработка кода выбранной программы и передача текущего состояния на средства отображения информации. Основной технологический цикл реализован через переключение этапов работы, что упрощает структуру программы и делает алгоритм более понятным.

В программе предусмотрены структуры данных для временного хранения результатов проверки, номеров неисправных плат и кодов ошибок. Подробно порядок вывода результатов и организация повторной проверки рассмотрены далее.

2.6 Организация информационного обмена между ЧПУ-контроллером и Arduino Mega 2560

Одной из основных задач при реализации комплекса стала организация согласованной работы ЧПУ-контроллера SMC4-4-16A16B и отладочной платой Arduino Mega 2560. ЧПУ-контроллер управляет перемещением манипулятора, а Arduino Mega 2560 управляет операциями программирования и тестирования только после того, как рабочий орган занял требуемое положение.

Взаимодействие между устройствами организовано с помощью цифровых сигналов. После перемещения к очередной плате и опускания контактной матрицы ЧПУ-контроллер формирует сигнал о достижении заданной позиции. Получив данный сигнал, Arduino Mega 2560 запускает технологический цикл проверки: выполняет контроль правильности подключения, программирование, тестирование и запись результата. После завершения этих операций Arduino Mega 2560 формирует ответный сигнал, разрешающий ЧПУ-контроллеру продолжить движение к следующей позиции. Листинг фрагмент управляющей программы движения на G-кодах приведён ниже:

```
G04P500
M211
M210
G04P500
M111
M110
G04P500
G01 X65.06181 Y-24.01168 F2000
G01 Z9 F1500
G04P500
G91
G01 Z-9 F150
G04P200
M214
G04P200
M114
M414
G01 Z9 F700
G01 X36.15 F1000
G01 Z-9 F150
```

В данном фрагменте в начале программы формируется код, по которому Arduino Mega 2560 определяет выбранный режим проверки. Далее ЧПУ-контроллер выполняет подход к первой плате, устанавливает рабочий орган в заданное положение, переходит к относительным координатам и опускает контактную матрицу к контактным площадкам модуля. После этого с помощью команды на M-коде формируется сигнал для Arduino Mega 2560 о готовности позиции. Команда ожидания используется для остановки выполнения программы движения до тех пор, пока управляющий МК не завершит программирование и тестирование текущего модуля и не подаст сигнал перехода на следующую позицию. Фрагмент кода общения с ЧПУ-контроллером представлен в приложении Б.2.

Кроме сигнала разрешения движения к следующей плате, в алгоритме предусмотрена последовательность аварийного возврата манипулятора в нулевую точку. Данная последовательность формируется Arduino Mega 2560 при возникновении аварийной ситуации, например при ошибке программатора, системной неисправности тестера, нарушении нормального хода проверки или обнаружении нескольких неисправных плат подряд. Перед подачей команды возврата управляющий МК сначала формирует сигнал сброса ЧПУ-контроллера. Это необходимо для остановки выполнения текущей программы движения и перевода ЧПУ-контроллера в безопасное состояние. После чего Arduino Mega 2560 формирует сигнал возврата в нулевую точку, по которому ЧПУ-контроллер выполняет перемещение рабочего органа в исходное положение. Такая последовательность позволяет безопасно остановить процесс проверки и исключить дальнейшее перемещение контактной матрицы по панелям при наличии неисправности.

Такой алгоритм так же обеспечивает синхронизацию механической и электронной частей комплекса. То есть, ЧПУ-контроллер не переходит к следующей плате до завершения технологических операций, а Arduino Mega 2560

не начинает проверку до подтверждения, что манипулятор прибыл в заданную координату.

2.7 Разработка алгоритма программирования и тестирования модуля

После получения сигнала от ЧПУ-контроллера о нахождении манипулятора в рабочей позиции Arduino Mega 2560 запускает цикл программирования/тестирования текущего электронного модуля. Данный цикл включает проверку правильности подключения, программирование модуля, тестирование его электрических параметров и функций, запись результата проверки и передачу разрешения ЧПУ-контроллеру на переход к следующей позиции.

Алгоритм работы управляющего МК построен поэтапно. Каждый этап выполняется только после успешного завершения предыдущего. Такой подход позволяет синхронизировать работу всех устройств комплекса и исключить ситуацию, при которой программирование или тестирование начнётся до полного подключения контактной игольчатой матрицы к проверяемому модулю.

После остановки манипулятора над очередной платой и опускания контактного устройства Arduino Mega 2560 выполняет проверку правильности позиционирования. Проверка осуществляется косвенным способом – по наличию напряжения на заданной контрольной точке проверяемого модуля. Если контактные иглы попали на соответствующие контактные площадки, то на входе управляющего МК появляется ожидаемый уровень напряжения. В этом случае подключение считается корректным, и комплекс переходит к программированию. Если напряжение отсутствует, фиксируется ошибка подключения или позиционирования, дальнейшая проверка данной платы не выполняется, а результат заносится в протокол. Фрагмент кода проверки правильности позиционирования представлен в приложении Б.3.

После подтверждения правильности подключения начинается этап программирования. На данном этапе управляющий МК переключает контактные линии игольчатой матрицы на программатор, подаёт питание на PICkit 5 и контролирует его состояние. Но, перед началом работы оператор устанавливает в программатор USB-накопитель с «прошивкой», соответствующей типу МК на проверяемых электронных модулях. Выбор накопителя выполняется по маршрутному листу или сопроводительной документации на партию плат. После включения питания программатор работает автономно, а запуск программирования и контроль его состояния выполняются управляющим МК.

На этапе программирования управляющее ПО Arduino Mega 2560 анализирует состояние входов, соответствующих светодиодной индикации программатора. По комбинации сигналов определяется текущее состояние PICkit 5: готовность к программированию, процесс программирования, ошибка или инициализация.

Если после включения питания программатор переходит в состояние готовности, Arduino Mega 2560 формирует сигнал запуска программирования. Запуск осуществляется через плату согласования PICkit 5 путём имитации нажатия штатной кнопки программатора. Для этого управляющий МК подаёт сигнал на оптопара, которое замыкает цепь кнопки запуска. Программатор воспринимает это как обычное нажатие кнопки и начинает процесс записи программы в МК проверяемого модуля.

Если программатор сразу выдаёт ошибку на этапе инициализации, управляющий МК выполняет повторный запуск. Такая ситуация может возникнуть, например, при отсутствии USB-накопителя с прошивкой, использовании неверного накопителя или неисправности программатора. Если после повторного запуска ошибка сохраняется, комплекс прекращает цикл проверки, возвращает манипулятор в нулевую точку и выдаёт оператору сообщение о неисправности программатора или необходимости проверки USB-накопителя с программой.

В процессе программирования Arduino Mega 2560 продолжает контролировать состояние PICkit 5 по сигналам светодиодной индикации. Если программирование завершается успешно, программатор переходит в состояние готовности, что воспринимается управляющим МК как успешное завершение этапа. Если в процессе программирования возникает ошибка, результат заносится в протокол проверки для данной платы, после чего комплекс переходит к дальнейшей обработке в соответствии с общим алгоритмом.

Питание программатора включается только на время выполнения операции программирования. После завершения программирования питание PICkit 5 отключается. Это необходимо для уменьшения влияния программатора на последующий процесс тестирования, так как часть контактных точек игольчатой матрицы используется как при программировании, так и при проверке электронного модуля. После отключения программатора коммутационные цепи переключают соответствующие линии на тестер, и комплекс переходит к следующему этапу технологического цикла. Фрагмент кода управления программированием представлен в приложении Б.4.

После завершения программирования происходит этап тестирования электронного модуля. Для этого Arduino Mega 2560 подаёт питание на тестер и проверяет наличие необходимого напряжения питания. Если питание тестера отсутствует, это воспринимается как системная неисправность, после чего комплекс прекращает выполнение текущего цикла и возвращает манипулятор в нулевую точку. Если питание присутствует, начинается выполнение проверки модуля.

Тестер выполняет проверку электронного модуля по заранее заданной методике. При этом управляющий МК не выполняет измерение всех электрических параметров напрямую, а контролирует ход проверки по состоянию светодиодной индикации тестера. По изменению состояния индикаторов Arduino Mega 2560 определяет переход между этапами проверки. Если последовательность изменения сигналов соответствует нормальному

прохождению теста, проверка продолжается. Если на одном из этапов возникает ошибка, МК фиксирует этап, на котором проверка была прервана, и заносит результат в протокол.

Так как, тестер был заимствован из состава технологического оборудования с ручным управлением, то для выполнения отдельных операций тестирования требуется имитация действий оператора. Поэтому в программе управляющего МК реализовано автоматическое формирование сигналов, соответствующих нажатию необходимых кнопок тестера. Это позволило использовать уже существующий специализированный тестер в составе автоматического комплекса без полной переработки его схемотехники.

Если тестирование завершилось успешно, в оперативной памяти Arduino Mega 2560 сохраняется информация о том, что текущая плата исправна. Также сбрасывается счётчик подряд идущих неисправных плат. Если тестирование завершилось ошибкой, в протокол заносится номер платы и соответствующий код неисправности. Номер платы определяется по текущему положению в последовательности проверки, а код ошибки формируется по этапу, на котором тестирование было прервано.

После фиксации результата тестирования питание тестера отключается. Это позволяет исключить лишнюю нагрузку на проверяемый модуль и подготовить контактные линии к переходу на следующую позицию. Далее управляющий МК анализирует результат проверки и принимает решение о продолжении работы комплекса.

В программе предусмотрена обработка аварийной ситуации, связанной с обнаружением нескольких неисправных плат подряд. Если в пределах одной панели фиксируются три подряд неисправные платы, комплекс прекращает дальнейшую проверку и возвращает манипулятор в нулевую точку. Такое решение необходимо для защиты модулей и контактной игольчатой матрицы, так как несколько последовательных ошибок могут свидетельствовать как о неисправности самих плат, так и о нарушении позиционирования, плохом

контакте или другой системной проблеме. При переходе ко второй панели счётчик подряд идущих неисправностей сбрасывается, поэтому неисправности, расположенные на разных панелях, не учитываются как одна непрерывная аварийная последовательность.

Если аварийных условий не обнаружено, Arduino Mega 2560 формирует сигнал разрешения дальнейшего движения. После получения этого сигнала ЧПУ-контроллер продолжает выполнение управляющей программы и перемещает контактное устройство к следующему электронному модулю. Далее цикл повторяется: манипулятор занимает новую позицию, опускает игольчатую матрицу, сообщает Arduino о готовности, после чего выполняются проверка подключения, программирование и тестирование следующей платы. Фрагмент кода управления тестированием представлен в приложении Б.5.

2.8 Реализация вывода результатов и режима повторной проверки

Для информационного взаимодействия с оператором в составе комплекса используется LCD-дисплей и кнопки управления. На дисплей выводится текущий этап работы комплекса: ожидание готовности манипулятора, проверка позиционирования и питания платы, программирование, тестирование, переход к следующей плате или возврат в нулевую точку. Это позволяет оператору контролировать ход процесса без подключения дополнительного оборудования.

После завершения проверки результаты выводятся на LCD-дисплей в виде протокола. Для удобства восприятия номера плат отображаются в координатной форме, аналогичной обозначениям в игре «морской бой»: A1, B1, C1, D1 и далее по расположению модулей на панели. Такой способ отображения позволяет оператору быстро найти неисправный модуль на групповой сборке.

Если неисправные модули отсутствуют, оператору выводится сообщение об успешном завершении проверки. Если в процессе проверки были обнаружены ошибки, на дисплее отображается список неисправных плат и соответствующие

коды ошибок. Переключение между строками протокола выполняется кнопками управления.

Результаты проверки хранятся в оперативной памяти Arduino Mega 2560. Для каждой позиции сохраняется номер проверяемой платы, признак исправности или неисправности, а также код ошибки. Кроме основного массива результатов формируется отдельный массив неисправных плат, который используется для повторной проверки.

Режим повторной проверки предназначен для исключения ложных отказов, которые могут возникать из-за нестабильного контакта игл с контактными площадками. После завершения основной проверки оператор может выбрать повторную проверку неисправных модулей. При этом панели остаются установленными на рабочей поверхности, а ЧПУ-контроллер повторно проходит по координатам модулей. Однако Arduino Mega 2560 запускает программирование и тестирование только для тех плат, которые были записаны в массив неисправных. Для остальных позиций технологический цикл пропускается. После выполнения повторной проверки на LCD-дисплей выводится новый результат.

Такой режим позволяет уменьшить вероятность ошибочного отбраковывания модулей, связанного не с реальной неисправностью платы, а с временным нарушением контакта между иглами и контактными площадками.

3 Результаты разработки и оценка эффективности комплекса

3.1 Основные результаты разработки

В результате выполнения дипломной работы был разработан, собран и введен в эксплуатацию опытный образец роботизированного АПК автоматического программирования и тестирования электронных модулей на печатных платах. Разработанный комплекс применен для работы с групповыми сборками плат и обеспечивает последовательное автоматическое подключение контактного тестирующего устройства к каждому электронному модулю, выполнение программирования, тестирования и выдачу результата проверки оператору.

Внедрение комплекса, позволило заменить ручное программирование и последовательную ручную проверку отдельных модулей автоматизированным технологическим процессом, при котором участие оператора ограничивается установкой панелей с платами на рабочую поверхность, запуском процесса проверки и последующим анализом полученных результатов.

Также была подготовлена инструкция по эксплуатации комплекса. В ней приведено назначение аппарата, описаны его основные блоки, порядок подготовки к работе, запуск программирования и тестирования, действия оператора при завершении проверки, а также расшифровка кодов ошибок. При внедрении инструкция прошла корректировку по результатам пробных проверок непосредственными исполнителями работ (операторами).

В ходе разработки были получены следующие основные конкретные результаты:

- разработан и собран опытный образец АПК, представленный на рисунке 30;
- разработана плата согласования для автоматического запуска и контроля состояния программатора PICkit 5;
- разработано ПО управляющего МК Arduino Mega 2560;

- подготовлены управляющие программы на G- и M-кодах для ЧПУ-контроллера SMC4-4-16A16B;
- реализован информационный обмен между ЧПУ-контроллером и управляющим МК;
- обеспечена автоматическая последовательность операций программирования и тестирования электронных модулей;
- реализован вывод результатов проверки на LCD-дисплей;
- реализован режим повторной проверки модулей, по которым были зафиксированы ошибки;
- разработана инструкция по эксплуатации комплекса;
- выполнена опытная проверка партии электронных модулей.



Рисунок 30 – Роботизированный комплекс автоматического программирования и тестирования электронных модулей на печатных платах

3.2 Результаты опытной проверки партии плат

Для оценки работоспособности разработанного комплекса была выполнена опытная проверка партии электронных модулей. Всего в ходе опытной эксплуатации было проверено 654 платы. Проверка выполнялась с использованием разработанного АПК в автоматическом режиме. Результаты опытной проверки представлены в таблице 2.

Таблица 2 – Результат опытной проверки электронных модулей

Показатель	Количество электронных модулей	Количество ошибок
Первичная проверка автоматическим комплексом	654	17
Вторичная проверка автоматическим комплексом	17	5
Ручная проверка	80	5

Для дополнительного контроля правильности работы комплекса часть плат была перепроверена на ручном пульте. Всего вручную было перепроверено 80 плат, выбранных случайным образом. Данная перепроверка выполнялась не потому, что все эти платы были признаны неисправными автоматом, а для подтверждения корректности работы комплекса и сопоставления результатов автоматической и ручной проверки.

По результатам опытной проверки были зафиксированы платы, по которым автоматический комплекс первоначально выдал сообщение об ошибке, однако, благодаря режиму повторной проверки, неисправность некоторых электронных модулей не подтвердилась. Такие случаи могут быть связаны с временным нарушением контакта между подпружиненными иглами и контактными площадками, особенностями позиционирования. Наличие данного режима позволяет снизить вероятность ошибочного отбраковывания исправных модулей.

Также были выявлены платы, по которым ошибка была зафиксирована как автоматическим комплексом, так и ручным пультом. Это подтверждает, что комплекс способен выявлять реальные неисправности электронных модулей и фиксировать их в результатах проверки.

То, что комплекс может иногда выдавать ложные заключения о неисправности, не является существенным недостатком. Так как для производства важнее не пропустить брак. Тем более, что количество ложных определений не велико. А вот исключение «человеческого фактора», напротив, практически исключает возможность пропуска брака.

По результатам опытной проверки можно сделать вывод, что разработанный комплекс обеспечивает выполнение заданного технологического процесс и позволяет проводить автоматическое программирование и тестирование электронных модулей в составе групповых сборок.

3.3 Оценка эффективности комплекса

В данном разделе рассматривается эффективность применения комплекса с точки зрения снижения трудоёмкости операций, то есть сокращения времени, затрачиваемого на проверку партии плат, и устранение так называемого «человеческого фактора».

При ручном способе программирование и тестирование одного электронного модуля занимают в среднем от 4 до 5 минут. Это время включает подключение платы к программатору, запуск программирования, переподключение к специализированному тестирующему устройству, выполнение проверки, анализ результата и переход к следующему модулю. При серийном производстве такие операции многократно повторяются, поэтому даже небольшое сокращение времени на одну плату приводит к заметному снижению общей трудоёмкости.

При использовании разработанного автоматизированного комплекса время программирования и тестирования одного модуля составляет около 55 секунд. Если учитывать полный цикл работы комплекса, включая перемещение

контактного тестирующего устройства между платами, опускание и подъём игольчатой матрицы, а также возврат манипулятора в исходное положение, проверка группы из 40 плат занимает в среднем 41-42 минуты.

Для сравнения, при ручном способе проверка 40 плат занимает от 160 до 200 минут. Следовательно, применение автоматизированного комплекса позволяет сократить время проверки одной такой группы примерно на 2-2,5 часа. В пересчёте на один электронный модуль экономия времени составляет около 3-4 минут. Сравнение времени ручной и автоматической проверки представлено в таблице 3.

Таблица 3 – Сравнение времени ручной и автоматической проверки

Способ проверки	Время на один модуль, мин.	Время на группу из 40 модулей, мин.
Ручной	4-5	160-200
Автоматический	≈1	≈41-42

Из приведённых данных видно, что разработанный комплекс позволяет значительно уменьшить время выполнения технологических операций. Особенно заметный эффект достигается при проверке групповых сборок, так как оператору не требуется вручную подключать и проверять каждую плату по отдельности. Его участие ограничивается установкой панели, запуском процесса и последующим просмотром результатов.

При ориентировочном годовом объёме проверки 3000–3500 модулей суммарное сокращение времени становится ещё более заметным. При ручном способе проверка такого количества плат заняла бы примерно от 200 до 292 часов в год. При использовании автоматизированного комплекса, с учётом среднего времени проверки около одной минуты на модуль, на тот же объём потребуется примерно от 52 до 61 часа.

Таким образом, годовая экономия времени при применении разработанного комплекса составляет примерно от 148 до 231 часа. Это время ранее затрачивалось бы оператором на выполнение однотипных операций ручного программирования, подключения, проверки и фиксации результатов.

Кроме сокращения времени, применение комплекса обеспечивает и другие положительные эффекты. Автоматизация снижает влияние человеческого фактора, повышает повторяемость условий проверки, уменьшает вероятность ошибок при фиксации результатов и позволяет выполнять повторную проверку только тех модулей, по которым были получены сообщения о неисправности.

Кроме того, сто процентное выявление ошибок до дальнейшей сборки изделия, дает еще одну дополнительную экономию времени. Так как исключается разборка забракованных конечных изделий и их повторная сборка.

Таким образом, разработанный АПК обеспечивает существенное сокращение времени программирования и тестирования электронных модулей. Это подтверждает целесообразность его применения в условиях серийного производства, где проверка большого количества однотипных плат является регулярной и трудоёмкой операцией.

3.4 Перспективы дальнейшего развития

Разработанный опытный образец подтвердил возможность автоматизации программирования и тестирования электронных модулей на печатных платах. При этом дальнейшее развитие комплекса может быть направлено на повышение универсальности, расширение области применения и улучшение средств обработки результатов.

Одним из основных направлений дальнейшей модернизации является создание сменных контактных тестирующих устройств под разные типы электронных модулей. В текущей реализации контактное тестирующее устройство адаптировано под конкретный тип проверяемых плат. Для применения комплекса с другой номенклатурой изделий потребуется разработка сменных устройств, включающих соответствующую игольчатую матрицу, элементы коммутации, программатор и тестирующее оборудование. Такой подход позволит использовать один и тот же координатный манипулятор и систему управления для проверки разных типов электронных модулей.

Также перспективным направлением является расширение системы хранения и обработки результатов проверки. В текущей реализации результаты выводятся на LCD-дисплей и используются оператором для дальнейшей работы с проверенной партией. В дальнейшем целесообразно реализовать запись результатов проверки в файл, передачу данных на ПК или сохранение информации в базе данных. Это позволит формировать электронный архив результатов контроля, упростить анализ статистики отказов и обеспечить более удобное документирование результатов испытаний.

Дополнительным направлением развития является применение измерительных линеек или других датчиков обратной связи по координатам перемещения. В текущем варианте перемещение выполняется ЧПУ-контроллером по заранее заданной программе. При использовании измерительных линеек управляющий контроллер сможет получать дополнительную информацию о фактическом перемещении манипулятора. Это позволит расширить возможности контроля положения рабочего органа, повысить информативность системы управления и использовать полученные данные для дополнительной диагностики работы механической части комплекса, исключить возможные сбои.

Ещё одним направлением модернизации является переход от Arduino Mega 2560 к управляющему МК семейства STM32. Применение STM32 позволит повысить быстродействие системы управления, увеличить объём доступной памяти, расширить количество интерфейсов обмена данными и упростить дальнейшую интеграцию комплекса с внешними устройствами. Кроме того, использование STM32 создаёт задел для реализации более сложных алгоритмов обработки данных, ведения электронного протокола проверки и обмена информацией с ПК.

Таким образом, дальнейшее развитие комплекса может включать:

- разработку сменных контактных тестирующих устройств под разные типы электронных модулей;
- расширение комплекса под другую номенклатуру выпускаемых изделий;

- запись результатов проверки в файл, на ПК или в базу данных;
- использование измерительных линеек для получения информации о фактическом перемещении манипулятора;
- переход на управляющий МК семейства STM32.

Реализация указанных направлений позволит перейти от опытного образца, предназначенного для проверки конкретного типа плат, к более универсальному АПК, пригодному для применения с различными электронными модулями в условиях серийного производства.

					ПГУ 1.090301.09.001 ПЗ	Лист
						61
Изм.	Лист	№ докум.	Подпись	Дата		

4 Безопасность жизнедеятельности

В данном разделе рассматриваются вопросы обеспечения безопасности жизнедеятельности при разработке и эксплуатации АПК для программирования и тестирования электронных модулей на печатных платах. Разрабатываемый комплекс включает ЧПУ-контроллер, управляющий МК, программатор, тестер, источники питания, драйверы шаговых двигателей и подвижный трёхкоординатный механизм. При этом ПК используется на этапах разработки программного обеспечения, подготовки управляющих программ, настройки и обслуживания комплекса. Поэтому требования безопасности должны учитывать как работу разработчика за ПК, так и эксплуатацию электрооборудования и движущихся частей комплекса.

4.1 Характеристика потенциальных рисков при работе с комплексом

В соответствии с классификацией ГОСТ 12.0.003-2015 к опасным и вредным производственным факторам при работе с комплексом могут относиться физические, психофизиологические и организационные факторы [12]. Наиболее существенными для рассматриваемого комплекса являются возможность поражения электрическим током, наличие движущихся частей манипулятора, повышенная зрительная нагрузка при работе с ПК и LCD-дисплеем, монотонность выполняемых операций.

К электрическим факторам относятся наличие цепей питания от сети 220 В переменного тока, источников постоянного напряжения 5, 12 и 24 В, а также вероятность короткого замыкания при повреждении изоляции, неправильном подключении разъёмов или попадании посторонних предметов в зону электрических соединений. Несмотря на то, что значительная часть управляющих цепей является низковольтной, источники питания и драйверы шаговых двигателей требуют соблюдения правил электробезопасности.

К механическим факторам относятся перемещение рабочего органа по координатам X, Y и Z, опускание контактной игольчатой матрицы на плату, а также наличие подпружиненных контактных игл. При нарушении правил эксплуатации возможно защемление пальцев, повреждение проверяемых плат или травмирование оператора острыми элементами контактного устройства.

4.2 Требования безопасности при эксплуатации комплекса

К работе с комплексом допускается персонал, ознакомленный с назначением устройства, инструкцией по эксплуатации, порядком запуска проверки и действиями при возникновении нештатных ситуаций. Перед началом работы оператор должен убедиться в исправности кабелей, разъёмов, органов управления, светозвуковой индикации, а также в отсутствии посторонних предметов в рабочей зоне манипулятора.

Панели с электронными модулями должны устанавливаться только на предусмотренные фиксирующие элементы. Запуск проверки допускается выполнять после того, как панель заняла правильное положение, а рабочая область свободна от проводов и других предметов. Во время перемещения манипулятора запрещается помещать руки в рабочую зону, перемещать проверяемые панели, касаться игольчатой матрицы и вмешиваться в работу привода.

При появлении запаха гари, дыма, нехарактерного шума, самопроизвольного движения, сбоя позиционирования или повторяющихся ошибок проверки работа комплекса должна быть остановлена. После остановки необходимо отключить питание и сообщить ответственному лицу о неисправности. Повторный запуск допускается только после устранения причины нештатной ситуации.

4.3 Организация рабочего места оператора

Рабочее место оператора должно обеспечивать удобное наблюдение за ходом проверки, доступ к органам управления и возможность безопасной установки панелей с электронными модулями. ЧПУ-контроллер, блоки питания и

					ПГУ 1.090301.09.001 ПЗ	Лист
						63
Изм.	Лист	№ докум.	Подпись	Дата		

другие элементы комплекса должны располагаться так, чтобы кабели не находились в проходах, не подвергались натяжению и не мешали действиям оператора.

4.4 Электробезопасность

Электробезопасность при эксплуатации комплекса должна обеспечиваться в соответствии с требованиями ГОСТ 12.1.019-2017, устанавливающего общие требования и виды защиты персонала от воздействия электрического тока, электрической дуги и электромагнитного поля [13]. Все элементы комплекса, питающиеся от сети 220 В, должны иметь исправную изоляцию, защитные кожухи и подключаться к сети с защитным заземлением.

Подключение и отключение разъёмов, замена программатора, тестера, контактного устройства или проверяемой панели должны выполняться только при отключённом питании соответствующих цепей. Запрещается эксплуатировать комплекс при повреждении кабелей, отсутствии защитных крышек, следах перегрева, неустойчивом контакте разъёмов или попадании жидкости на оборудование.

Провода питания и сигнальные линии должны быть закреплены так, чтобы исключить их механическое повреждение при движении манипулятора. Обслуживание источников питания, драйверов шаговых двигателей и других электрических узлов должно выполняться обученным персоналом. Оператор должен знать порядок отключения питания комплекса и основные действия при поражении электрическим током.

4.5 Пожарная безопасность

Пожарная безопасность комплекса должна обеспечиваться системой предотвращения пожара и организационно-техническими мероприятиями, что соответствует общим положениям ГОСТ 12.1.004-91 [14]. Основными причинами пожароопасных ситуаций могут быть короткое замыкание, перегрев источников питания и драйверов шаговых двигателей, перегрузка электрической сети,

					ПГУ 1.090301.09.001 ПЗ	Лист
						64
Изм.	Лист	№ докум.	Подпись	Дата		

использование неисправных удлинителей или накопление пыли внутри оборудования.

Для снижения риска возгорания необходимо использовать исправные источники питания и коммутационную аппаратуру, не допускать перегрузки розеток, не закрывать вентиляционные отверстия и не размещать горючие материалы рядом с блоками питания, драйверами и другими нагревающимися элементами. Комплекс должен располагаться на безопасном расстоянии от отопительных приборов и легковоспламеняющихся материалов.

В помещении должны быть предусмотрены первичные средства пожаротушения, свободные проходы и доступ к средствам отключения электропитания. Персонал должен знать расположение огнетушителей, порядок эвакуации и действия при обнаружении признаков возгорания. После окончания работы рекомендуется отключать питание комплекса, если дальнейшая работа не планируется.

4.6 Санитарно-гигиенические требования

В помещении необходимо поддерживать нормируемые параметры микроклимата, обеспечивать вентиляцию, регулярную уборку пыли и достаточную освещённость рабочей зоны.

Освещённость рабочего места должна быть достаточной для уверенного наблюдения за LCD-дисплеем, органами управления и проверяемыми платами. Для работы с электронными модулями особенно важно исключить недостаточную освещённость зоны установки панелей и зоны контактной игольчатой матрицы.

Температурный режим в помещении рекомендуется поддерживать на уровне, обеспечивающем комфортную работу оператора и стабильную работу оборудования. Необходимо исключать перегрев источников питания и электронных узлов, обеспечивать свободную циркуляцию воздуха около корпуса комплекса и своевременно удалять пыль с рабочих поверхностей. Уровень шума от работающих приводов и вентиляции не должен создавать помех для нормальной работы оператора.

					ПГУ 1.090301.09.001 ПЗ	Лист
						65
Изм.	Лист	№ докум.	Подпись	Дата		

Соблюдение перечисленных требований позволяет снизить вероятность травмирования персонала, уменьшить влияние вредных факторов, обеспечить безопасную эксплуатацию комплекса и повысить надёжность выполнения операций автоматического программирования и тестирования электронных модулей.

					ПГУ 1.090301.09.001 ПЗ	Лист
						66
Изм.	Лист	№ докум.	Подпись	Дата		

Заключение

В ходе выполнения дипломной работы были изучены способы автоматизации контроля электропараметров и функционирования электронных модулей на печатных платах, а также рассмотрены существующие решения для проверки плат в условиях серийного производства.

В процессе работы был разработан и собран опытный образец роботизированного АПК автоматического программирования и тестирования электронных модулей. В состав комплекса вошли трёхкоординатный манипулятор, ЧПУ-контроллер SMC4-4-16A16B, управляющий МК Arduino Mega 2560, программатор PICkit 5, специализированный тестер и игольчатая матрица.

Была разработана плата согласования для автоматического запуска программатора и считывания его состояния. Также было написано ПО для Arduino Mega 2560 и подготовлены управляющие программы на G- и M-кодах для ЧПУ-контроллера.

Работоспособность комплекса была подтверждена в ходе опытной проверки партии электронных модулей. Всего в автоматическом режиме было проверено 654 модуля. По результатам проверки установлено, что комплекс способен выполнять программирование, тестирование, фиксировать неисправные платы и выводить результаты оператору.

Применение разработанного комплекса позволяет сократить время проверки электронных модулей и снизить трудоёмкость операций. При годовом объёме 3000-3500 модулей экономия времени составляет примерно 148-231 час в год.

В дальнейшем комплекс можно развивать путём создания сменных контактных тестирующих устройств под разные типы плат, записи результатов проверки в файл или базу данных, применения измерительных линеек для контроля перемещения манипулятора, а также перехода на управляющий МК семейства STM32.

Список использованных источников

1. DipTrace tutorial: [Электронный ресурс]. – URL: <https://diptrace.com/books/tutorial.pdf> (дата обращения 15.04.2026).
2. diptrace.com.: [Электронный ресурс]. – URL: <https://diptrace.com/rus/download/older-versions/> (дата обращения 15.04.2026).
3. arduino.cc.: [Электронный ресурс]. – URL: <https://www.arduino.cc/en/software/> (дата обращения 01.05.2026).
4. LinuxCNC G-code: reference [Электронный ресурс]. – URL: <https://linuxcnc.org/docs/html/gcode/g-code.html> (дата обращения 26.04.2026).
5. LinuxCNC M-code: reference [Электронный ресурс]. – URL: <https://linuxcnc.org/docs/html/gcode/m-code.html> (дата обращения 26.04.2026).
6. SMC4-4-16A16B Four-Axis Motion Controller: Instruction Manual [Электронный ресурс]. – URL: https://cdn.hackaday.io/files/1588196794912096/DTM_850_controller_manual.pdf (дата обращения 14.04.2026).
7. MPLAB PICkit 5 In-Circuit Debugger: User's Guide [Электронный ресурс]. – URL: <https://www.microchip.com/en-us/development-tool/PG164150> (дата обращения 03.05.2026).
8. Mega 2560 Rev3: [Электронный ресурс]. – URL: <https://docs.arduino.cc/hardware/mega-2560/> (дата обращения 01.05.2026).
9. LP2950, LP2951, NCV2951 voltage regulator: data sheet [Электронный ресурс]. – URL: <https://static.chipdip.ru/lib/994/DOC031994832.pdf> (дата обращения 04.05.2026).
10. HSSR-S1A05L-2: data sheet [Электронный ресурс]. – URL: <https://static.chipdip.ru/lib/493/DOC048493593.pdf> (дата обращения 04.05.2026).
11. MCP601/602/603/604 Operational Amplifier Family: data sheet [Электронный ресурс]. – URL: <https://static.chipdip.ru/lib/922/DOC012922799.pdf> (дата обращения 04.05.2026).

12. ГОСТ 12.0.003-2015 Система стандартов безопасности труда. Опасные и вредные производственные факторы. Классификация: [Электронный ресурс]. – URL: <https://docs.cntd.ru/document/1200136071> (дата обращения 20.05.2026).

13. ГОСТ 12.1.019-2017 Система стандартов безопасности труда. Электробезопасность. Общие требования и номенклатура видов защиты: [Электронный ресурс]. – URL: <https://docs.cntd.ru/document/1200161238> (дата обращения 20.05.2026).

14. ГОСТ 12.1.004-91 Система стандартов безопасности труда. Пожарная безопасность. Общие требования: [Электронный ресурс]. – URL: <https://docs.cntd.ru/document/9051953> (дата обращения 20.05.2026).

					ПГУ 1.090301.09.001 ПЗ	Лист
						69
Изм.	Лист	№ докум.	Подпись	Дата		

Приложение А

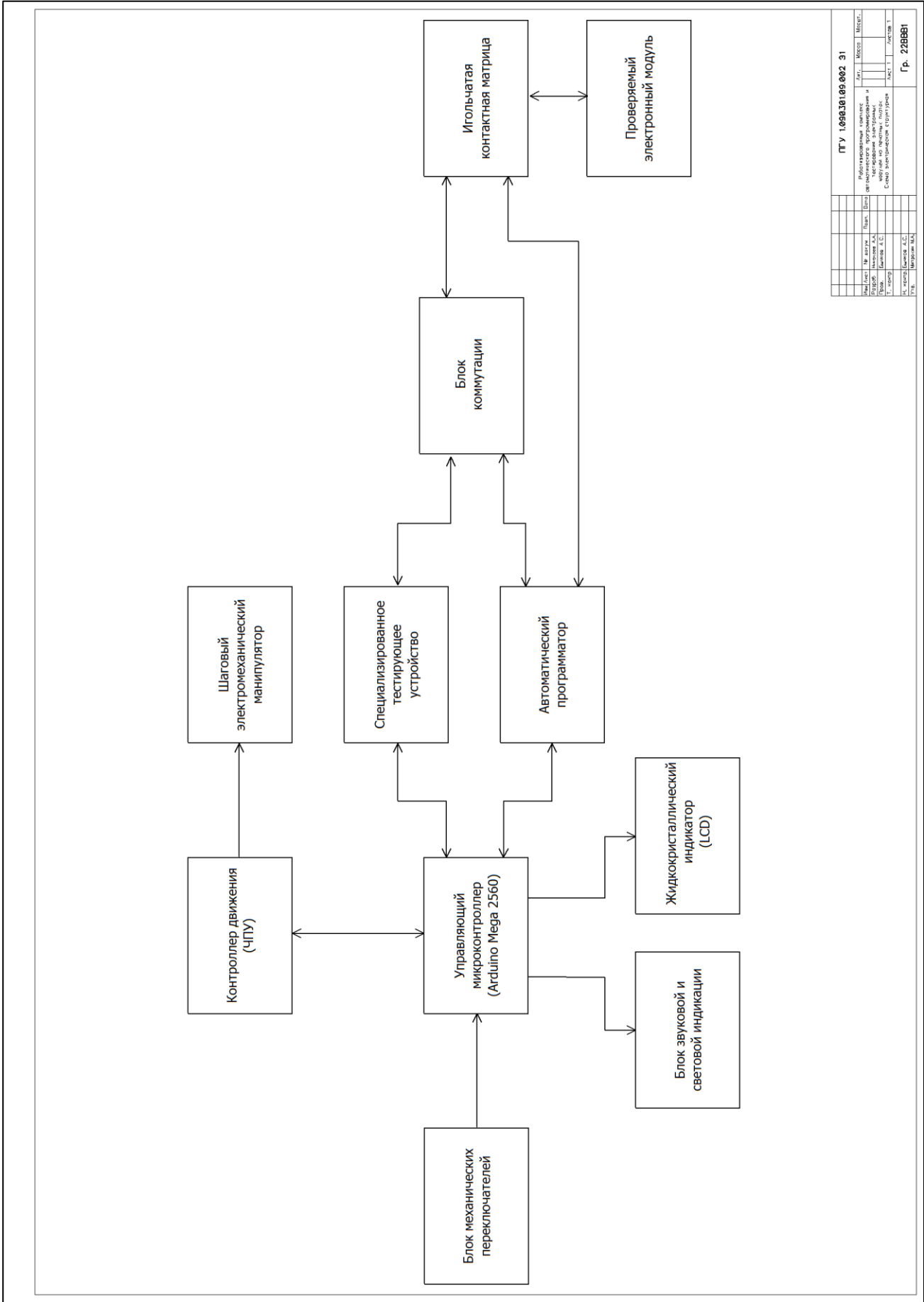
(обязательное)

Роботизированный комплекс автоматического программирования и тестирования
электронных модулей на печатных платах

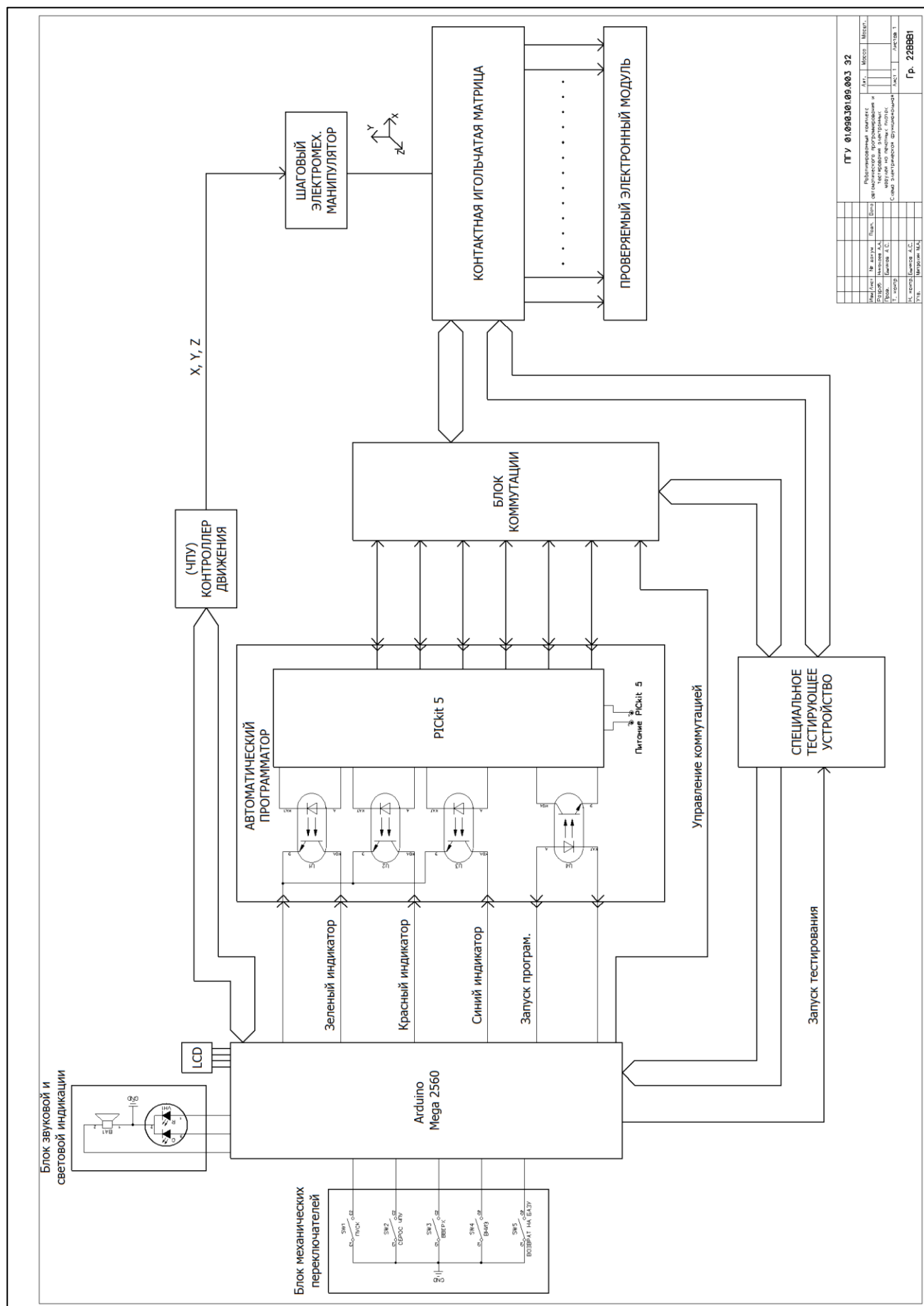
Схемы электрические

					ПГУ 1.090301.09.001 ПЗ	Лист
						70
Изм.	Лист	№ докум.	Подпись	Дата		

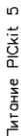
Приложение А.1 – Схема электрическая структурная комплекса



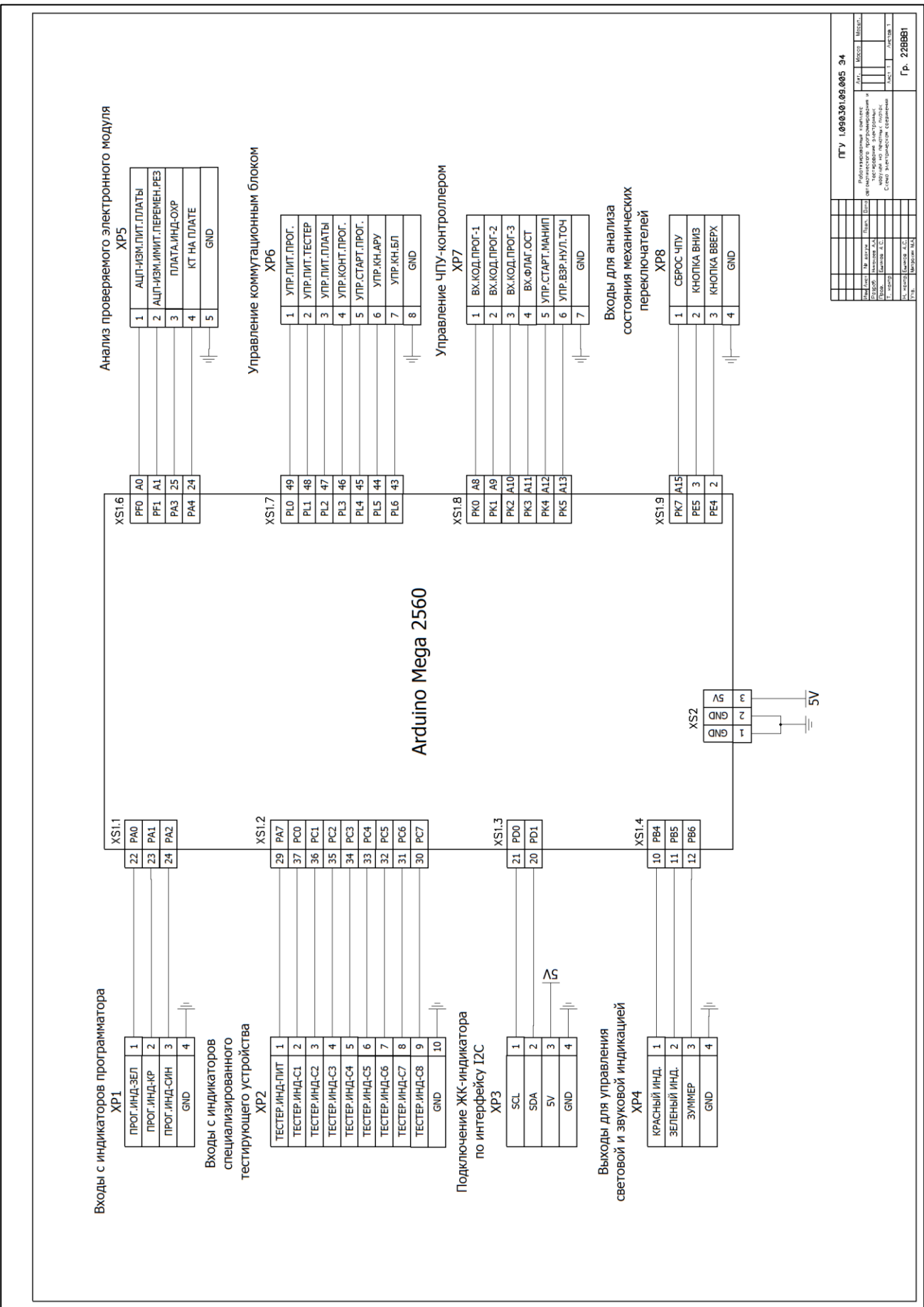
Приложение А.2 – Схема электрическая функциональная комплекса



СОГЛАСОВАНИЯ



Приложение А.4 – Схема электрическая соединений



Приложение Б

(обязательное)

Роботизированный комплекс автоматического программирования и тестирования
электронных модулей на печатных платах

Листинг фрагментов программы

					ПГУ 1.090301.09.001 ПЗ	Лист
						75
Изм.	Лист	№ докум.	Подпись	Дата		

Приложение Б.1 – Основной алгоритм программы

```
void loop() {
    /* Функции, выполняющиеся постоянно независимо от текущего этапа
    работы манипулятора */
    readPowerAndVarRes(&ReadingAdcData);
    analyzeProgrammator();
    analyzeLedTestingBoard();
    returnToHome();
    returnZeroPointPushed(&RETURN_ZERO);
    checkChangeStageForMonitoring(&MainFlags);
    sendBasicState(&MainFlags);
    analyzeCode();
    /* Этапы работы манипулятора */
    switch (MainFlags.stageOfWork)
    {
        case WAIT_READY_MANIPULATOR:
        {
            MAINSTAGE_waitReadyManipulator(&MainFlags);
            break;
        }
        case CHECK_POSITION:
        {
            MAINSTAGE_checkPosition(&MainFlags);
            break;
        }
        case PROGRAMMER:
        {
            MAINSTAGE_Programming(&MainFlags, &StateProgrammator);
            break;
        }
        case TESTING:
        {
            MAINSTAGE_Testing(&MainFlags);
            break;
        }
        case CHECK_BRANCH:
        {
            if(!MAINSTAGE_checkEndTestAndProg(&MainFlags,
            &StateProgrammator))
            {
                MainFlags.stageOfWork = GOTO_NEXTPLATE;
                MainFlags.subStageOfWork = START_TO_NEXT;
            }
            break;
        }
        case GOTO_NEXTPLATE:
        {
            MAINSTAGE_goToNextPlate(&MainFlags);
            break;
        }
    }
}
```

```

    }
    case GOTO_ZEROPOINT:
    {
        MAINSTAGE_goToZeroPoint(&MainFlags,&StateProgrammator);
        break;
    }
    default;;
}
}

```

					ПГУ 1.090301.09.001 ПЗ	Лист
						77
Изм.	Лист	№ докум.	Подпись	Дата		

Приложение Б.2 – Функции взаимодействия с ЧПУ-контроллером

```

/* Функция переезда на следующую плату */
void MAINSTAGE_goToNextPlate (_MAIN_FLAGS *flags) {
    switch (flags->subStageOfWork)
    {
        case START_TO_NEXT:
        {
            PORTK = writePin(PINK,MAN_START,1);
            mainDelay(300,&DelayMs);
            flags->subStageOfWork = WAIT_BUTTON_RESET_NEXT;
            break;
        }
        case WAIT_BUTTON_RESET_NEXT:
        {
            if (!DelayMs.startWaitFlag)
            {
                PORTK = writePin(PINK,MAN_START,0);
                if (flags->startingCheckCodeProgFlag)
                    flags->subStageOfWork = CHECK_CODE;
                else
                    flags->subStageOfWork = CHECK_STAGE_MAN;
                flags->stageOfWork= WAIT_READY_MANIPULATOR;
            }
            break;
        }
        default;;
    }
}

/* Функция возврата в ноль*/
void MAINSTAGE_goToZeroPoint (_MAIN_FLAGS *flags, _STATE_PROGRAMATOR
*programator) {
    switch (flags->subStageOfWork)
    {
        case RESET:
        {
            if (programator->powerProgramOk)
            {
                PORTL = writePin(PINL,POWER_PROG,0);
                programator->powerProgramOk = 0;
            }
            PORTK = writePin(PINK,MAN_RESET,0);
            mainDelay(500,&DelayMs);
            flags->subStageOfWork = WAIT_BUTTON_RESET_RESET;
            break;
        }
        case WAIT_BUTTON_RESET_RESET:
        {
            if (!DelayMs.startWaitFlag)
            {
                PORTK = writePin(PINK,MAN_RESET,1);
            }
        }
    }
}

```

```

        mainDelay(500,&DelayMs);
        flags->subStageOfWork = START_TO_ZERO;
    }
    break;
}
case START_TO_ZERO:
{
    if (!DelayMs.startWaitFlag)
    {
        PORTK = writePin(PINK,MAN_ZEROPOINT,0);
        mainDelay(500,&DelayMs);
        flags->subStageOfWork = WAIT_BUTTON_RESET_ZERO;
    }
    break;
}
case WAIT_BUTTON_RESET_ZERO:
{
    if (!DelayMs.startWaitFlag)
    {
        PORTK = writePin(PINK,MAN_ZEROPOINT,1);
        flags->stageOfWork = WAIT_READY_MANIPULATOR;
        flags->subStageOfWork = 0;
    }
    break;
}
default;;
}
}

/* Функция ожидания готовности манипулятора */
void MAINSTAGE_waitReadyManipulator(_MAIN_FLAGS *flags) {
    switch(flags->subStageOfWork)
    {
        case CHECK_CODE:
        {
            uint8_t code = flags->readingCode;
            if(flags->startingCheckCodeProgFlag)
            {
                if (flags->readingCode != 0b00000011)
                {
                    Serial.print("Код: ");
                    Serial.println(code, BIN);
                    Serial.print("Количество проверяемых плат - ");
                    switch (code)
                    {
                        case 0:
                        {
                            Serial.println("40");
                            break;
                        }
                        case 1:
                        {

```

```

        Serial.println("20");
        break;
    }
    case 2:
    {
        Serial.println("1");
        break;
    }
    default:;
}
flags->codeTypeOrNumberPlate = code;
flags->startingCheckCodeProgFlag = 0;
flags->subStageOfWork = CHECK_STAGE_MAN;
flags->startProcess = 1;
flagLCD.printTesting = 1;
}
}
break;
}
case CHECK_STAGE_MAN:
{
    flags -> stopTest = 0;
    if(!readPin(PINK, MAN_ARRIVED))
    {
        flags->countCheckingPlate++;
        /* Сброс количества подряд неисправных плат при переходе на
следующую панель */
        if (flags->codeTypeOrNumberPlate == 0)
        {
            if (flags->countCheckingPlate == 21) flags-
>countBreaksPlate = 0;
        }
        flags->subStageOfWork = SELECT_MODE;
    }
    mainDelay(300,&DelayMs);
    break;
}
case SELECT_MODE:
{
    if(!DelayMs.startWaitFlag)
    {
        if(flags -> retesting)
        {
            if(flags->countCheckingPlate == RETEST_PLATE[iter.i]+1)
            {
                flags->stageOfWork = CHECK_POSITION;
                flags->subStageOfWork = SET_POWER_TESTINGPLATE;
                iter.i++;
            }
            else
            {
                flags->normOrBreakPlateFlag = 1;
            }
        }
    }
}

```

```

        flags->currentNumPlate = flags->countCheckingPlate+1;
        flags->countBreaksPlate = 0;
        flags->stageOfWork = CHECK_BRANCH;
    }
}
else
{
    flags->stageOfWork = CHECK_POSITION;
    flags->subStageOfWork = SET_POWER_TESTINGPLATE;
}
}
break;
}
default;;
}
}

```

					ПГУ 1.090301.09.001 ПЗ	Лист
						81
Изм.	Лист	№ докум.	Подпись	Дата		

Приложение Б.3 – Функция проверки правильности позиционирования

```

/* Функция проверки правильности позиции манипулятора */
void MAINSTAGE_checkPosition (_MAIN_FLAGS *flags) {
    switch (flags->subStageOfWork)
    {
        case SET_POWER_TESTINGPLATE:
        {
            /* Подача питания платы тестера -----*/
            PORTL = writePin(PINL, POWER_TESTER, 1);
            /* -----*/
            mainDelay(2000, &DelayMs);
            flags->subStageOfWork = CHECK_CP5;
            break;
        }
        case CHECK_CP5:
        {
            if (!DelayMs.startWaitFlag)
            {
                uint8_t checkOk = 0;
                /* Проверка флага наличия питания на проверяемой плате,
                поскольку напряжение измеряется постоянно в loop */
                if (ReadingAdcData.powerPlateOk)
                    checkOk = 1;
                if (checkOk)
                    flags->subStageOfWork = CHECK_VARRES;
                else
                {
                    flags->errorPosition = 1;
                    flags->normOrBreakPlateFlag = 0;
                    flags->currentNumPlate = flags->countCheckingPlate+1;
                    if (checkBreakPlate(flags))
                    {
                        flags->stageOfWork = GOTO_ZEROPOINT;
                        flags->subStageOfWork = RESET;
                    }
                    else
                    {
                        flags->stageOfWork = CHECK_BRANCH;
                    }

                    /* Снятие питания с платы тестирования -----*/
                    PORTL = writePin(PINL, POWER_TESTER, 0);
                    /* -----*/
                }
            }
            break;
        }
        case CHECK_VARRES:
        {
            uint8_t checkOk = 0;
            /* Измерение напряжения на переменном резисторе */

```

```

        if ((ReadingAdcData.varResValue >=
215)&&(ReadingAdcData.varResValue <= 250))
            checkOk = 1;
        if (checkOk)
        {
            flags->subStageOfWork = RESET_POWER_TESTINGPLATE;
        }
        else
        {
            flags->errorPosition = 1;
            flags->normOrBreakPlateFlag = 0;
            flags->currentNumPlate = flags->countCheckingPlate+1;
            if (checkBreakPlate(flags))
            {
                flags->stageOfWork = GOTO_ZEROPOINT;
                flags->subStageOfWork = RESET;
            }
            else
            {
                flags->stageOfWork = CHECK_BRANCH;
            }
            /* Снятие питания с платы тестирования -----*/
            PORTL = writePin(PINL, POWER_TESTER, 0);
            /* -----*/
        }
        break;
    }
    case RESET_POWER_TESTINGPLATE:
    {
        /* Снятие питания с платы тестирования -----*/
        PORTL = writePin(PINL, POWER_TESTER, 0);
        /* -----*/
        flags->stageOfWork = PROGRAMMER;
        flags->subStageOfWork = CONNECT_CONTACTS;
        mainDelay(1000, &DelayMs);
        break;
    }
    default:;
}
}

```

Приложение Б.4 – Функция программирования платы

```

/* Функция программирования платы */
void MAINSTAGE_Programming (_MAIN_FLAGS *flags, _STATE_PROGRAMATOR
*programator) {
    switch (flags->subStageOfWork)
    {
        case CONNECT_CONTACTS:
        {
            /* Подключение контактов программатора -----*/
            PORTL = writePin(PINL, CONTACTS_PROG, 1);
            /* -----*/
            flags->subStageOfWork = SET_POWER_PROG;
            break;
        }
        case SET_POWER_PROG:
        {
            if (!DelayMs.startWaitFlag)
            {
                /* Подача питания на программатор -----*/
                if (!programator->powerProgramOk)
                {
                    PORTL = writePin(PINL, POWER_PROG, 1);
                    programator->powerProgramOk = 1;
                }
                /* -----*/
                flags->subStageOfWork = CONNECT_POWER_PLATE;
                mainDelay(1000, &DelayMs);
            }
            break;
        }
        case CONNECT_POWER_PLATE:
        {
            if (!DelayMs.startWaitFlag)
            {
                /* Подача питания на тестируемую плату -----*/
                PORTL = writePin(PINL, POWER_TESTING_BOARD, 1);
                /* -----*/
                flags->subStageOfWork = WAIT_READY_PROG;
                mainDelay(1000, &DelayMs);
            }
            break;
        }
        case WAIT_READY_PROG:
        {
            if (!DelayMs.startWaitFlag)
            {
                if (programator->stateProgram == NORMAL)
                {
                    RELAY_ON;
                    flags->subStageOfWork = START_PROGRAM_PROCESS;
                }
            }
        }
    }
}

```

Изм.	Лист	№ докум.	Подпись	Дата

```

else if (programator->stateProgram == FAILURE)
{
    /* Неисправность программатора, возврат манипулятора на
нулевую точку */
    flags->errorProgrammatorOrFlash = 1;
    flags->normOrBreakPlateFlag = 0;
    flags->currentNumPlate = flags->countCheckingPlate+1;
    /* Отключение питания программатора и тестируемой платы,
отключение
    * контактов от программатора
    */
    PORTL = writePin(PINL, POWER_TESTING_BOARD, 0);
    PORTL = writePin(PINL, POWER_PROG, 0);
    programator->powerProgramOk = 0;
    PORTL = writePin(PINL, CONTACTS_PROG, 0);
    /* ----- */
    flags->restartProgrammator = 0;
    flags->startWriteFinalProtocol = 1;
    flags->stageOfWork = GOTO_ZEROPOINT;
    flags->subStageOfWork = RESET;
}
}
break;
}
case START_PROGRAM_PROCESS:
{
    if (programator->stateProgram == PROGRAMMING)
    {
        mainDelayProgramming(&DelaysMsProg);
        flags->subStageOfWork = WAIT_END_PROGRAM;
    }
    else if (programator->stateProgram == FAILURE)
    {
        if (!flags->restartProgrammator)
        {
            flags->restartProgrammator = 1;
            PORTL = writePin(PINL, POWER_TESTING_BOARD, 0);
            PORTL = writePin(PINL, POWER_PROG, 0);
            programator->powerProgramOk = 0;
            mainDelay(3000, &DelayMs);
            flags->subStageOfWork = SET_POWER_PROG;
        }
        else if (flags->restartProgrammator)
        {
            flags->subStageOfWork = END_STEP_PROGRAMMER;
        }
    }
    break;
}
case WAIT_END_PROGRAM:
{
    if (programator->stateProgram == NORMAL)

```

```

    {
        DelaysMsProg.startWaitFlagProg = 0;
        flags->restartProgrammator = 0;
        flags->subStageOfWork = END_STEP_PROGRAMMER;
    }
    else if (programator->stateProgram == FAILURE ||
DelaysMsProg.countProg >= 200)
    {
        if (DelaysMsProg.startWaitFlagProg)
        {
            DelaysMsProg.startWaitFlagProg = 0;
            flags->restartProgrammator = 1;
        }

        if (!flags->restartProgrammator)
        {
            flags->restartProgrammator = 1;
            PORTL = writePin(PINL, POWER_TESTING_BOARD, 0);
            PORTL = writePin(PINL, POWER_PROG, 0);
            programator->powerProgramOk = 0;
            mainDelay(3000, &DelayMs);
            flags->subStageOfWork = SET_POWER_PROG;
        }
        else if (flags->restartProgrammator)
        {
            flags->subStageOfWork = END_STEP_PROGRAMMER;
        }
    }
    break;
}
case END_STEP_PROGRAMMER:
{
    /* Отключение контактов программатора и питания тестируемой
платы */
    PORTL = writePin(PINL, POWER_TESTING_BOARD, 0);
    PORTL = writePin(PINL, CONTACTS_PROG, 0);
    if (flags->restartProgrammator)
    {
        flags->errorProgrammingPlate = 1;
        flags->restartProgrammator = 0;
        /* Запись в протокол номера неисправной платы */
        flags->normOrBreakPlateFlag = 0;
        flags->currentNumPlate = flags->countCheckingPlate+1;
        if (checkBreakPlate(flags))
        {
            flags->stageOfWork = GOTO_ZEROPOINT;
            flags->subStageOfWork = RESET;
        }
        else
        {
            flags->stageOfWork = CHECK_BRANCH;
        }
    }
}

```

```

/* Отключение питания программатора при неисправности -----
-----*/
if (programator->powerProgramOk)
{
    PORTL = writePin(PINL, POWER_PROG, 0);
    programator->powerProgramOk = 0;
}
/* -----
-----*/
}
else
{
    if (programator->powerProgramOk)
    {
        PORTL = writePin(PINL, POWER_PROG, 0);
        programator->powerProgramOk = 0;
    }
    flags->stageOfWork = TESTING;
    flags->subStageOfWork = SET_POWER_TESTING_BOARD;
    mainDelay(500, &DelayMs);
}
break;
}
default;;
}
}

```

Приложение Б.5 – Функция тестирования платы

```

/* Функция для проверки смены этапа тестирования.
   Применяется в функции MAINSTAGE_Testing() */
void checkingChangeLedStepOfTest (_MAIN_FLAGS *flags, _INPUT
*ledCurrent, _INPUT *ledLast){
    if (!flags->waitChangeStateLedTest)
    {
        if (!ledCurrent->state)
            flags->waitChangeStateLedTest = 1;
        else
        {
            if (!ledLast->state)
                flags->subStageOfWork = FAULT_TEST;
        }
    }
    else
    {
        if (ledCurrent->state)
        {
            flags->waitChangeStateLedTest = 0;
            flags->subStageOfWork++;
        }
    }
}

/* Основная функция тестирования платы */
void MAINSTAGE_Testing (_MAIN_FLAGS *flags) {
    switch (flags->subStageOfWork)
    {
        case SET_POWER_TESTING_BOARD:
        {
            if (!DelayMs.startWaitFlag)
            {
                /* Подача питания платы тестера -----*/
                PORTL = writePin(PINL, POWER_TESTER, 1);
                /*-----*/
                mainDelay(1000, &DelayMs);
                flags->subStageOfWork = CHECK_POWER_TESTING_BOARD;
            }
            break;
        }
        case CHECK_POWER_TESTING_BOARD:
        {
            if (!DelayMs.startWaitFlag)
            {
                if (TP_POWER.state)
                    flags->subStageOfWork = WAIT_LED_C8;
                else
                {
                    flags->subStageOfWork = FAULT_TEST;
                    flags->errorTestingBoard = 1;
                }
            }
        }
    }
}

```

```

    }
}
break;
}
case WAIT_LED_C8:
{
    if (!flags->waitChangeStateLedTest)
    {
        if (!TP_LED[C8].state)
            flags->waitChangeStateLedTest = 1;
    }
    else
    {
        if (TP_LED[C8].state)
        {
            flags->waitChangeStateLedTest = 0;
            flags->subStageOfWork = CTRL_C1;
            mainDelay(1000, &DelayMs);
        }
    }
    break;
}
case CTRL_C1:
{
    if (!DelayMs.startWaitFlag)
    {
        if (!TP_LED[C1].state)
        {
            flags->subStageOfWork = SET_ARU;
            ARU_ON;
        }
        else
        {
            if (!TP_LED[C8].state)
            {
                flags->subStageOfWork = FAULT_TEST;
                flags->numOfErrorStageTesting = C8+1;
            }
        }
    }
    break;
}
case SET_ARU:
{
    if (!flags->waitChangeStateLedTest)
    {
        if (TP_LED[C1].state)
        {
            flags->waitChangeStateLedTest = 1;
            mainDelay(1500, &DelayMs);
        }
    }
}

```

```

else
{
    if (!TP_LED[C1].state)
    {
        if (!DelayMs.startWaitFlag)
        {
            flags->waitChangeStateLedTest = 0;
            flags->subStageOfWork = SET_BL;
            BL_ON;
        }
    }
}
break;
}
case SET_BL:
{
    if (TP_LED[C1].state)
    {
        flags->subStageOfWork = CTRL_C2;
    }
    break;
}
case CTRL_C2:
{
    checkingChangeLedStepOfTest(flags, &TP_LED[C2], &TP_LED[C1]);
    if (flags->subStageOfWork == FAULT_TEST)
        flags->numOfErrorStageTesting = C1+1;
    break;
}
case CTRL_C3:
{
    checkingChangeLedStepOfTest(flags, &TP_LED[C3], &TP_LED[C2]);
    if (flags->subStageOfWork == FAULT_TEST)
        flags->numOfErrorStageTesting = C2+1;
    break;
}
case CTRL_C4:
{
    checkingChangeLedStepOfTest(flags, &TP_LED[C4], &TP_LED[C3]);
    if (flags->subStageOfWork == FAULT_TEST)
        flags->numOfErrorStageTesting = C3+1;
    break;
}
case CTRL_C5:
{
    checkingChangeLedStepOfTest(flags, &TP_LED[C5], &TP_LED[C4]);
    if (flags->subStageOfWork == FAULT_TEST)
        flags->numOfErrorStageTesting = C4+1;
    break;
}
case CTRL_C6:
{

```

```

        checkingChangeLedStepOfTest(flags, &TP_LED[C6], &TP_LED[C5]);
        if (flags->subStageOfWork == FAULT_TEST)
            flags->numOfErrorStageTesting = C5+1;
        break;
    }
case CTRL_C7:
{
    checkingChangeLedStepOfTest(flags, &TP_LED[C7], &TP_LED[C6]);
    if (flags->subStageOfWork == FAULT_TEST)
        flags->numOfErrorStageTesting = C6+1;
    break;
}
case CTRL_C8:
{
    if (!TP_LED[C8].state)
    {
        flags->waitChangeStateLedTest = 0;
        flags->subStageOfWork = END_TESTING;
        /* Запись в протокол номер платы и то, что она исправна */
        flags->normOrBreakPlateFlag = 1;
        flags->currentNumPlate = flags->countCheckingPlate+1;
        /* Сброс счётчика неисправных подряд плат */
        flags->countBreaksPlate = 0;
    }
    else if (!TP_LED[C7].state)
    {
        flags->subStageOfWork = FAULT_TEST;
        flags->numOfErrorStageTesting = C7+1;
    }
    break;
}
case FAULT_TEST:
{
    flags->errorTest = 1;
    if (flags->errorTestingBoard)
    {
        flags->stageOfWork = GOTO_ZEROPOINT;
        flags->subStageOfWork = RESET;
    }
    else
    {
        /* Запись в протокол номера неисправной платы */
        flags->normOrBreakPlateFlag = 0;
        flags->currentNumPlate = flags->countCheckingPlate+1;
        if (checkBreakPlate(flags))
        {
            flags->stageOfWork = GOTO_ZEROPOINT;
            flags->subStageOfWork = RESET;
        }
    }
}
}

```

```

case END_TESTING:
{
    /* Отключаем питание с платы тестера -----*/
    PORTL = writePin(PINL, POWER_TESTER, 0);
    /*-----*/
    if (flags->stageOfWork != GOTO_ZEROPOINT)
    {
        flags->stageOfWork = CHECK_BRANCH;
    }
    break;
}
default;;
}
}

```

					ПГУ 1.090301.09.001 ПЗ	Лист
						92
Изм.	Лист	№ докум.	Подпись	Дата		

Приложение В

(обязательное)

Роботизированный комплекс автоматического программирования и тестирования
электронных модулей на печатных платах

Презентация

					ПГУ 1.090301.09.001 ПЗ	Лист
						93
Изм.	Лист	№ докум.	Подпись	Дата		

РОБОТИЗИРОВАННЫЙ КОМПЛЕКС АВТОМАТИЧЕСКОГО ПРОГРАММИРОВАНИЯ И ТЕСТИРОВАНИЯ ЭЛЕКТРОННЫХ МОДУЛЕЙ НА ПЕЧАТНЫХ ПЛАТАХ

Автор работы: студент гр. 22ВВВ1 Николаев Александр Алексеевич

Научный руководитель: к.т.н., доцент каф. ВТ Бычков Андрей Станиславович

Цель работы — разработка роботизированного автоматического комплекса для программирования и тестирования электронных модулей на печатных платах, обеспечивающего последовательное подключение к модулям в групповой панели, выполнение программирования, тестирования, фиксацию ошибок и вывод результатов оператору.

Задачи:

- Выполнить анализ существующих технологических решений
- Обосновать актуальность разработки роботизированного комплекса
- Выбрать аппаратные средства комплекса
- Разработать структуру комплекса
- Разработать программное обеспечение управления
- Провести опытную проверку комплекса и оценить его эффективность

АНАЛИЗ АНАЛОГОВ

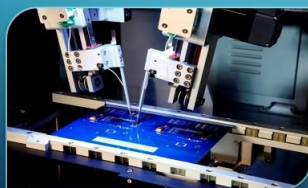
ЛЕТАЮЩИЕ ПРОБНИКИ

Достоинства:

- Подходят для проверки сложных плат
- Могут гибко перенастраиваться

Недостатки:

- Требуется точная калибровка
- Невысокая скорость проверки



ПОЛУАВТОМАТИЧЕСКИЙ КОМПЛЕКС

Достоинства:

- Быстрое подключение к электронным модулям
- Не требует подачи тестовых сигналов

Недостатки:

- Оператор устанавливает и снимает каждую плату поочередно
- Малое контактное поле

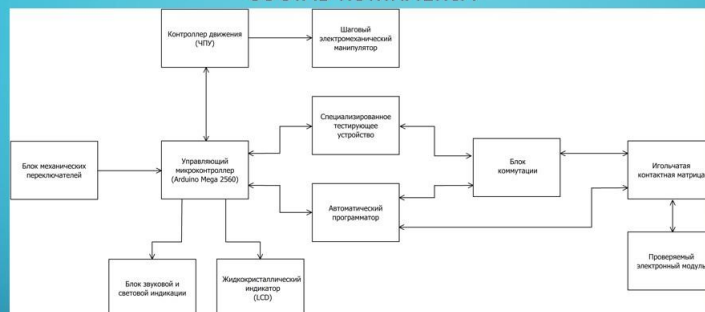


АКТУАЛЬНОСТЬ

- Ручная проверка занимает много времени
- Вероятность ошибок оператора
- Неудобство проверки групповых сборок поштучно



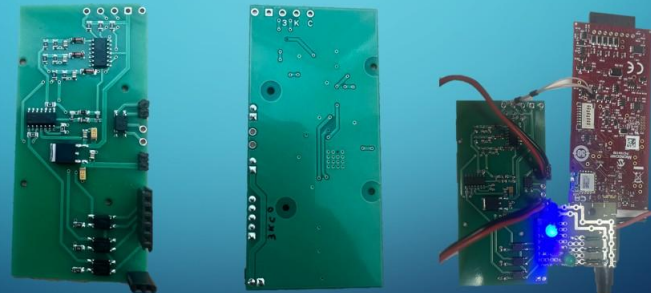
СОСТАВ КОМПЛЕКСА



ПРИНЦИП РАБОТЫ КОМПЛЕКСА

- 1) Оператор устанавливает групповые панели и запускает процесс
- 2) Перемещение рабочего органа к первой проверяемой плате
- 3) Проверка правильности позиционирования
- 4) Программирование проверяемого модуля
- 5) Тестирование проверяемого модуля
- 6) Сохранение результата
- 7) Переход к следующей плате

ПЛАТА СОГЛАСОВАНИЯ ДЛЯ ПРОГРАММАТОРА PICKIT 5



The image displays three views of a custom PCB designed for a PICkit 5 programmer. The left view shows the top of the board with various electronic components including a PICkit 5 module, a USB cable, and several integrated circuits. The middle view shows the bottom of the board, which is mostly empty with some surface markings. The right view shows the board assembled with a PICkit 5 module and a USB cable, with a blue LED indicator lit.



РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ



РЕЗУЛЬТАТ ОПЫТНОЙ ПРОВЕРКИ И ЭФФЕКТИВНОСТЬ РАБОТЫ КОМПЛЕКСА

Опытная проверка

Показатель	Количество электронных модулей	Количество ошибок
Первичная проверка автоматическим комплексом	654	17
Вторичная проверка автоматически м комплексом	17	5
Ручная проверка	80	5

Эффективность

Способ проверки	Время на один модуль, мин.	Время на группу из 40 модулей, мин.
Ручной	4-5	160-200
Автомат	~1	~41-42

РЕЗУЛЬТАТ ОПЫТНОЙ ПРОВЕРКИ И ЭФФЕКТИВНОСТЬ РАБОТЫ КОМПЛЕКСА

Опытная проверка

Показатель	Количество электронных модулей	Количество ошибок
Первичная проверка автоматическим комплексом	654	17
Вторичная проверка автоматически м комплексом	17	5
Ручная проверка	80	5

Эффективность

Способ проверки	Время на один модуль, мин.	Время на группу из 40 модулей, мин.
Ручной	4-5	160-200
Автомат	~1	~41-42

РЕЗУЛЬТАТ ОПЫТНОЙ ПРОВЕРКИ И ЭФФЕКТИВНОСТЬ РАБОТЫ КОМПЛЕКСА

Опытная проверка

Показатель	Количество электронных модулей	Количество ошибок
Первичная проверка автоматическим комплексом	654	17
Вторичная проверка автоматически м комплексом	17	5
Ручная проверка	80	5

Эффективность

Способ проверки	Время на один модуль, мин.	Время на группу из 40 модулей, мин.
Ручной	4-5	160-200
Автомат	~1	~41-42

ПЕРСПЕКТИВЫ ДАЛЬНЕЙШЕГО РАЗВИТИЯ

- Разработка сменных контактных тестирующих устройств под разные типы электронных модулей
- Расширение комплекса под другую номенклатуру выпускаемых изделий
- Запись результатов проверки в файл, на ПК или в базу данных
- Использование измерительных линеек для получения информации о фактическом перемещении манипулятора
- Переход на управляющий МК семейства STM32

СПАСИБО ЗА ВНИМАНИЕ!