Министерство науки и высшего образования Российской Федерации
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Российский экономический университет имени Г.В. Плеханова»
Московский приборостроительный техникум

Выпускная квалификационная работа
(Дипломная работа)

На тему: Разработка программного комплекса с предиктивной коррекцией ошибок управления (на примере ООО «Центр инновационных разработок ВАО»)

Дрюпина Андрея Александровича
Студент 4 курса группы П50-1-18

по специальности 09.02.07 «Информационные системы и программирование»
для присвоения квалификации: программист
Форма обучения: очная

Руководитель: _____ / Копылов Олег Валерьевич /
(подпись)
«____»_____2022 г.

Консультант: _____ / Попова Любовь Юрьевна /
(подпись)
«____»_____2022 г.

Студент: _____ / Дрюпин Андрей Александрович /
(подпись)
«____»_____2022 г.

Допущена к защите
Распоряжение от «_____»_____2022 г. № _____

2022

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Робототехника в современном мире является достаточно важной его частью. Это сфера, совершенствованию которой уделяется особое внимание ввиду пользы, которую она приносит человечеству. Промышленность, медицина, военно-промышленный комплекс, сельское хозяйство – лишь немногие примеры сфер, где робототехнические механизмы нашли обширное применение. Однако робототехническая отрасль, несмотря на все достижения, не является совершенной, и имеет массу проблем самого разного плана. Одной из таких проблем - скорость смены полезной нагрузки. Для большинства платформ вспомогательное оборудование не является универсальным и время разработки и установки зависит от его сложности и функциональности. Одна из целей выпускной квалификационной работы – спроектировать, разработать и запрогрограммировать универсальный интерфейс связи вспомогательного оборудования с ведущей платформой. Вторая цель выпускной квалификационной работы – разработка предиктивной системы коррекции ошибок управления. Эта система является вспомогательной для оператора платформы и призвана максимально снизить человеческий фактор.

Универсальные платформы используются для проведения широкого спектра работ на территориях, опасных для человека. Используя быстрозаменяемое вспомогательное оборудование, оператор робота может осуществлять: обследование подозрительных вещей, извлечение людей из-под завалов, радиационный и химический контроль, забор проб воды и грунта.

Было принято решение создать универсальный роботизированный аппаратно-программный комплекс для выполнения работ в условиях опасных и вредных факторов с возможностью быстрой замены полезной нагрузки.

Актуальность данной работы заключается в необходимости создания отечественной универсальной роботизированной платформы для использования его в гражданских (в том числе коммерческих) целях, требующих быстрой и легкой замены вспомогательной нагрузки на дистанционно-управляемых роботах. Проект особенно актуален для служб экстренного реагирования.

1. ОБЩАЯ ЧАСТЬ

1.1. Цель разработки

Создать универсальный роботизированный аппаратно-программный комплекс для выполнения работ в условиях опасных и вредных факторов (опасность обрушения, радиационная, биологическая или химическая угроза, высокие температуры) с возможностью быстрой замены полезной нагрузки.

1.2. Средства разработки

В качестве средств разработки указаны программы, представленные в таблице 1.

Таблица 1 – Программные средства

№	Тип средства	Название средства	Назначение
1	2	3	4
1	Текстовый редактор	Microsoft Word 2019 MSO (16.0.13530.20368)	Разработка документации, формирование отчетных документов
2	Инструментальное средство разработки программных решений	Visual Studio 2019 Community 16.8.4	Разработка десктоп приложения
3	Инструментальное средство разработки программных решений	Arduino IDE 1.8.10	Разработка встраиваемого приложения
4	Инструментальное средство разработки программных решений	Atmel Studio 4.19	Разработка встраиваемого приложения
5	Средство проектирования	Draw.io 2.0.9	Разработка схем для проектирования приложения
6	Средство логического анализа сигналов	Saleae Logic 2.3.45	Логический анализ сигналов
7	Операционная система	Microsoft Windows 10 Pro	Операционная система

В качестве средств вычислительной техники использовался персональный компьютер. Его характеристики представлены в таблице 2.

Таблица 2 – Технические средства

№	Тип оборудование	Наименование оборудование
1	2	3
Ноутбук HP Envy x360 13-ar0010ur		
1	Размер экрана	13.3”
2	Разрешение экрана	1920x1080
3	Линейка процессора	AMD Ryzen 5 3500U
4	Количество ядер процессора	4
5	Оперативная память	8 ГБ
6	Тип видеокарты	встроенная
7	Видеокарта	AMD Radeon Vega 8
8	Конфигурация накопителей	SSD
9	Общий объем всех накопителей	128 ГБ

В качестве средств периферийной техники использовались устройства, приведённые в таблице 3.

Таблица 3 – Периферийные устройства

№	Наименование	Описание
1	2	2
1	Логический анализатор	Logic Pro 16
2	4-х канальный осциллограф	Hantek DSO4254C
3	Генератор сигналов	UNI-T UTG901C
4	AVR программатор	USBAsp
5	AVR JTAG Debugger	AVR ISP II

2. СПЕЦИАЛЬНАЯ ЧАСТЬ

2.1. Постановка задачи

Разработать встроенное приложение для управления универсальной роботизированной платформой-носителем с предиктивной системой коррекции ошибок управления (на примере ООО «Центр инновационных разработок ВАО»).

В рамках выпускной квалификационной работы ставится задача разработать и собрать образец автоматизированного универсального роботизированного комплекса, в который входит:

1) Универсальная платформа-носитель;
2) Универсальный интерфейс ввода/вывода для подключения полезной нагрузки.

2.2. Входные и выходные данные

Входные данные - поток данных в виде структурированного текста, представляющий собой строку из ASCII символов, конвертируемые программой в структуру данных;

Выходные данные – поток данных в виде структурированного текста, представляющий собой строку из ASCII символов, конвертируемые программой в структуру данных.

2.2.1. Подробные требования к проекту

В качестве требований проекта необходимо выполнить следующие задачи:

- Реализовать аппаратную часть платформы-носителя: основной вычислительный модуль, модуль радиосвязи, модуль получения пространственных и физических координат;
- Реализовать подключение к управляющему компьютеру через радиоканал для управления платформой-носителем на расстоянии, а также для получения телеметрии оператором;
- Реализовать SDK для сторонних разработчиков внешних модулей,

путем создания библиотеки с готовыми программными модулями управления, контроля и тестирования аппаратной части;

- Реализовать возможность передвижения платформы с задаваемой оператором программно-аппаратного комплекса скоростью и направлением;
- Реализовать возможность передачи телеметрии на управляющий компьютер;
- Реализовать определение текущих физических координат платформы;
- Реализовать определение текущих пространственных координат;
- Реализовать систему предиктивной коррекции ошибок управления для упрощения работы оператора с платформой, вывод подробной информации о всех узлах и модулях платформы-носителя и полезной нагрузки, динамического поддержания параметров движения, а также вывод предупредительных сообщений о некорректной работе программно-аппаратного комплекса;
- Реализовать программно-аппаратный интерфейс для подключения внешней полезной нагрузки к платформе-носителю по технологии Plug and Play («Подключи и работай»).

2.3. Внешняя спецификация

2.3.1. Описание задачи

Робототехнический комплекс состоит из универсальной роботизированной платформы – носителя полезной нагрузки с интегрированным интерфейсом для подключения внешних устройств (рабочего инструмента), манипулятора с уникальной системой контроля перемещения и программного обеспечения, осуществляющего управление робототехнической системой как единым целым. Предполагается организация техподдержки и сервиса на территории

России.

Основные технические параметры:

- Масса – 20 кг;
- Габаритные размеры – 400x250x110 мм
- Грузоподъёмность – 15 кг;
- Максимальная скорость – 5 км/час;
- Максимальная дальность управления на прямой видимости – 1 км;
- Тип разъема порта ввода/вывода – D-sub 15;
- Максимально выдаваемая мощность порта ввода/вывода – 80 Вт;
- Максимальная скорость порта ввода/вывода в командном режиме – 2 Мб/сек;
- Модель связи порта ввода/вывода – Мастер/ведомый - Мастер/ведомый;
- Вид механического крепления полезной нагрузки – универсальные рельсы;
- Время автономной работы на одном элементе питания – 40 минут;
- Максимально возможное количество встраиваемых элементов питания – 6 шт;
- Предполагаемая цена разрабатываемой платформы-носителя – 400 тыс. руб.

Разработчик встраиваемого приложения скачивает приложение для станции управления, SDK для платформы и подключает скачанные библиотеки к своему проекту. Далее он создает новый объект класса платформы, содержащий все необходимые функции для работы с аппаратной частью. В самом начале необходимо провести

инициализацию объекта, введя в аргументы метода функциональное название платформы и уникальный ключ, и провести инициализацию метода-обработчика команд с персонального компьютера. После первоначальной инициализации разработчик приложения пишет свою управляющую программу, используя методы класса платформы.

Работа с универсальным интерфейсом осуществляется в двух режимах: ручном и командном. Ручной режим позволяет управляющей программе конфигурировать любой контакт интерфейса. В командном режиме передача осуществляется при помощи командных пакетов и пакетов с данными, в полудуплексном режиме на скорости до 2 Мб/сек.

2.3.2. Входные и выходные данные

В Таблице 4 представлены входные данные, получаемые с компьютера.

Таблица 4 – Входные данные

№	Наименование	Тип	Ограничения	Формат ввода	Описание
1	2	3	4	5	6
1	move	[A-Z, a-z] {1,1}	Char	Пакет, передаваемый с компьютера на платформу	Направление движения
2	Speed	[0-9, ,] {1, 100}	Int		Скорость в процентах
3	Value	[A-Z, a-z] {1,1}	Char		Тип разгона
4	azimutloc	[0-9, ,] {1, 100}	Int		Азимут движения
5	Gpio1	[0-9 , ,] {1, 100}	Double		значение gpio1 masterLink (masterLink - универсальный интерфейс подключения устройств)
6	Gpio2	[0-9 , ,] {1, 100}	Double		значение gpio2 masterLink (masterLink - универсальный интерфейс подключения

№	Наименование	Тип	Ограничения	Формат ввода	Описание
1	2	3	4	5	6
					устройств)
7	Gpio3	[0-9 , ,] {1, 100}	Double		значение gpio3 masterLink (masterLink - универсальный интерфейс подключения устройств)
8	Gpio4	[0-9 , ,] {1, 100}	Double		значение gpio4 masterLink (masterLink - универсальный интерфейс подключения устройств)
9	Systemstatus	[0-9 , ,] {1, 100}	Int		Статус системы
10	data	[A-Z, a-z] {1,256}	String		Дополнительные данные

В Таблице 5 представлены выходные данные, отправляемые на компьютер.

Таблица 5 – Выходные данные

№	Наименование	Тип	Ограничения	Формат ввода	Описание
1	2	3	4	5	6
1	Move	Char	[A-Z, a-z] {1,1}	Пакет, передаваемый с платформы на компьютер	Направление движения
2	speed	Int	[0-9, ,] {1, 100}		Скорость в процентах
3	value	Char	[A-Z, a-z] {1,1}		Тип разгона
4	Lcurr	Double	[0-9 , ,] {1, 100}		Значение тока левого мотора
5	Rcurr	Double	[0-9 , ,] {1, 100}		Значение тока правого мотора
6	Accx	Double	[0-9 , ,] {1, 100}		Значение акселерометра по оси x
7	Accy	Double	[0-9 , ,] {1, 100}		Значение акселерометра по оси y
8	Accz	Double	[0-9 , ,] {1, 100}		Значение акселерометра по

№	Наименование	Тип	Ограничения	Формат ввода	Описание
1	2	3	4	5	6
					оси z
9	Gyrox	Double	[0-9 , ,] {1, 100}		Значение гироскопа по оси x
10	Gyroy	Double	[0-9 , ,] {1, 100}		Значение гироскопа по оси y
11	Gyroz	Double	[0-9 , ,] {1, 100}		Значение гироскопа по оси z
12	Magx	Double	[0-9 , ,] {1, 100}		Значение магнетометра по оси x
13	Magy	Double	[0-9 , ,] {1, 100}		Значение магнетометра по оси y
14	Magz	Double	[0-9 , ,] {1, 100}		Значение магнетометра по оси z
15	Lan	String	[A-Z, a-z] {1,256}		Значение широты
16	Lon	String	[A-Z, a-z] {1,256}		Значение долготы
17	vbat	Double	[0-9 , ,] {1, 100}		Значение напряжения батареи в вольтах
18	Exitd	int	[0-9] {1, 7}		Идентификатор внешнего устройства
19	exitstatus	int	[0-9] {1, 1}		Статус внешнего устройства

2.3.3. Методы

При разработке приложения для управления роботизированной платформы-носителем использовались следующие методы:

- Объектно-ориентированное программирование: работа с классами, методами и структурами;
- Данные, передаваемые платформой на компьютер представлены на рисунке 1.
- Данные, передаваемые компьютером на платформу представлены на рисунке 2.

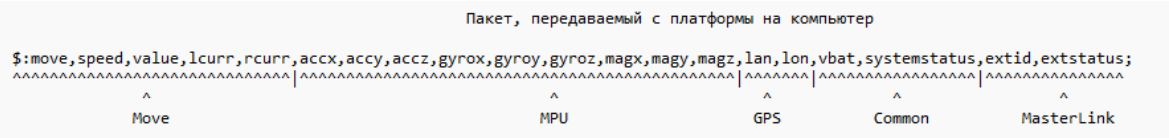

Рисунок 1 – Пакет Platform to PC

Аргументы пакета:

move - направление движения, char

speed - скорость в процентах, int

value - тип разгона, char

lcurr - значение тока левого мотора, double

rcurr - значение тока правого мотора, double

accx - значение акселерометра по оси x, double

accy - значение акселерометра по оси y, double

accz - значение акселерометра по оси z, double

gyrox - значение гироскопа по оси x, double

gyroy - значение гироскопа по оси y, double

gyroz - значение гироскопа по оси z, double

magx - значение магнетометра по оси x, double

magy - значение магнетометра по оси y, double

magz - значение магнетометра по оси z, double

lan - широта, string

lon - долгота, string

vbat - напряжение батареи в вольтах, double

extid - идентификатор внешнего устройства

extstatus - статус внешнего устройства, int

В получаемом пакете находятся значения с конкретных датчиков изменения которых, непосредственно отображаются в приложении. Например, если значения «move» - направление движения платформы будет меняться, это означает что платформа меняет своё направление движения.

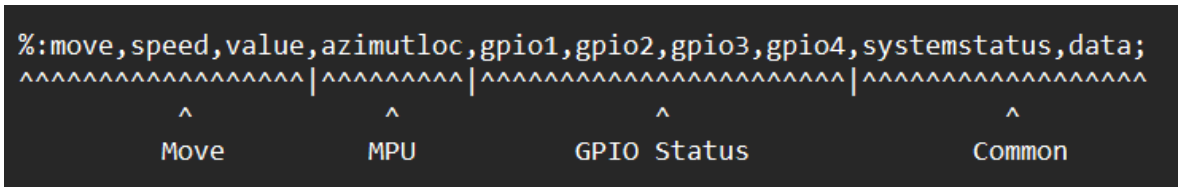

Рисунок 2 - Пакет PC to Platform

Аргументы пакета:

move - направление движения, char

speed - скорость в процентах, int

value - тип разгона, char

azimutloc - азимут движения, int

gpio1 - значение gpio1 masterLink, double

gpio2 - значение gpio2 masterLink, double

gpio3 - значение gpio3 masterLink, double

gpio4 - значение gpio4 masterLink, double

systemstatus - статус системы, int

data - Дополнительные данные, String

2.3.4. Тесты

После разработки приложения был отобран следующие виды тестирования:

Функциональное тестирование:

- Тестирование методом черного ящика;
- Метод «Входной двери» (Бинарное тестирование yes/no).

Тестирование производительности:

- Нагрузочное тестирование;
- Стресс тестирование.

Далее определившись с видами тестирования, был произведен ряд тестовых испытаний:

Тестирование методом черного ящика было применено при тестировании работоспособности функционала встроенного приложения для передачи команд на станцию управления.

Метод «Входной двери» применялся при тестировании работы приложения со станцией управления.

Нагрузочное тестирование - тестирование предназначено для проверки работоспособности системы при стандартных нагрузках и для определения максимально возможного пика, при котором система работает правильно.

Стресс тестирование - Тестирование предназначено для проверки работоспособности системы при нестандартных нагрузках и для определения максимально возможного пика, при котором система работает правильно. Так же предназначено для выявления результатов, при которых система переходит в нерабочее состояние.

Тестовые сценарий и результаты тестирования указаны в приложении Б «Результаты и сценарий тестовых испытаний».

2.4. Проектирование

2.4.1. Схема архитектуры программы

Схема архитектуры программы представлена на рисунке 3.

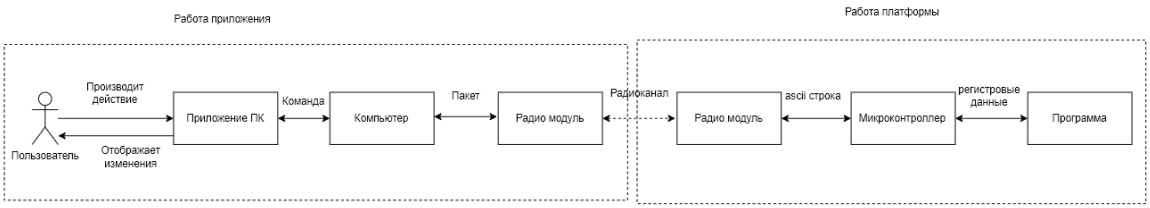

Рисунок 3 – Схема архитектуры программы

«Производит действие» - работа с приложением при помощи компьютерной мыши или клавиатуры;

«Команда» - работа приложения с ПК на аппаратном системном уровне;

«Пакет» - работа ПК с радио модулем на аппаратном уровне периферии;

«Радиоканал» - обмен байтами по радиоканалу на частоте 443.920 МГц.

«ascii строка» - передача данных на уровне представления;

«Регистровые данные» - работа программы с микроконтроллером на уровне представления регистров;

«Отображает изменения» - пользователь видит изменения данных.

2.4.2. Структурная схема программы

Структурная схема программы представлена на рисунке 4.

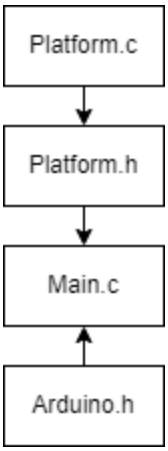

Рисунок 4– Структурная схема приложения

Описание модулей представлено в таблице 6.

Таблица 6 – Описание модулей

№	Название модуля	Описание модуля
1	Main.c	Точка входа
2	Platform.h	Заголовочный файл библиотеки управления
3	Platform.c	Файл логики библиотеки управления
4	Arduino.h	Заголовочный файл вспомогательных функций платформы Ардуино

2.4.3. Функциональная схема

Функциональная схема программы представлена на рисунке 5.

Рисунок 5 - Функциональная схема приложения

2.4.4. Аппаратная схема

Аппаратная структура платформы представлена на рисунке 6.

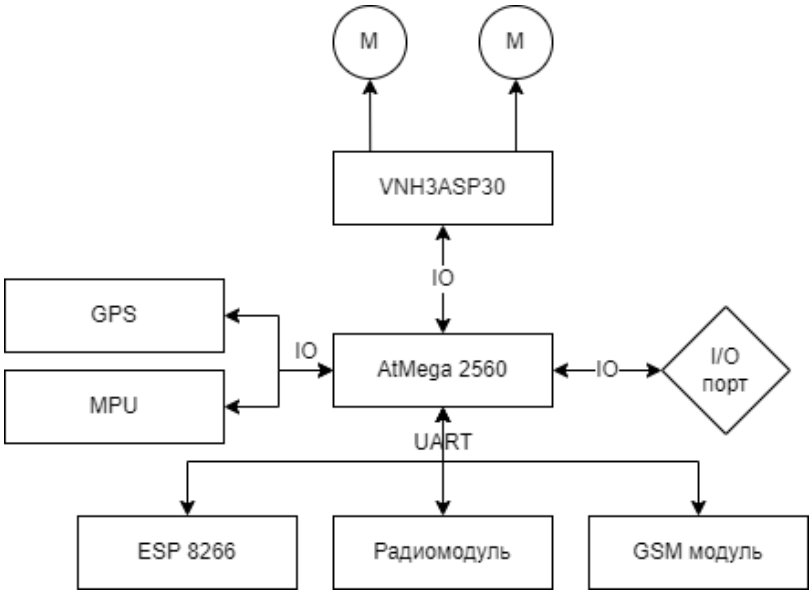

Рисунок 6 - Аппаратная схема

Описание компонентов:

Atmega2560 – AVR Микроконтроллер Atmel серии Mega;

M – Тяговый мотор;

VNH3ASP30 – Сдвоенный драйвер моторов;

GPS – GNSS модуль точных геокоординат;

MPU – Модуль отслеживания пространственных перемещений;

ESP 8266 – Модуль Wi-Fi;

Радиомодуль – Приемопередающий модуль для связи с ПК посредством защищенного радиоканала;

GSM модуль - Приемопередающий модуль для связи с ПК посредством передачи пакетов через станции подвижной связи (сотовая связь);

I/O Порт – Порт интерфейса MasterLink для связи с полезной нагрузкой.

2.5. Результаты работы программы

Так как у встраиваемого приложения нет интерфейса, на рисунке 7 приведены основные управляющие сигналы, полученные при помощи логического анализатора.

Рисунок 7- Основные сигналы платформы

Описание сигналов:

- Канал 0 – Исходящие данные от платформы
- Канал 1 – Входящий сигнал
- Канал 2 – Тактовый сигнал драйвера мотора
- Канал 3 – Отладочный сигнал векторов прерываний UART
- Время обработки полученного сигнала микроконтроллером: 2,1186 ms;
- Время работы прерывания UART: 9,999687 ms;
- Длительность отладочных сигналов между пакетами: 16 ns;

3. ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ

3.1. Инструментальные средства разработки

Для разработки приложения использовалась среда разработки Arduino IDE, которая позволяет писать код на языке C++ и Wiring с встроенным компилятором под архитектуру AVR. Преимуществами данной среды разработки является удобный графический интерфейс, быстрая компиляция программного кода и возможность быстрого редактирования программного кода. Архитектура AVR была выбрана из-за простоты разработки под нее, легкой поддержки, а также из-за встроенной богатой периферии.

Для работы с сигналами использовался программно-аппаратный комплекс Saleae Logic PRO. Это один из немногих комплексов для работы с сигналами. Его преимуществами является информативный графический интерфейс, большая база предустановленных библиотек для анализа цифровых сигналов, большое количество вводных каналов и скорость работы близкая к 1 ГГц.

Для работы с документацией использовался текстовый процессор, предназначенный для создания, просмотра и редактирования текстовых документов, с локальным применением простейших форм таблично-матричных алгоритмов. Выпускается корпорацией Microsoft в составе пакета Microsoft Office. Используемой версией является Microsoft Office Word 2016 для Windows. Его преимуществами являются большая функциональность, поддержка мультиоперационности, а также удобный интерфейс.

Для создания схем использовался векторный графический редактор «Draw.io», редактор диаграмм и блок-схем для различных операционных систем. Его преимуществом является Web-расположение, без необходимости установки.

В качестве операционной системы использовалась операционная

система для персональных компьютеров и рабочих станций, разработанная корпорацией Microsoft в рамках семейства Windows NT. Она является распространенной и проработанной.

3.2. Отладка программы

Основными ошибками были ошибки переполнения входного буфера UART, получение неточных координат GNSS, получение неправильных данных с датчиков.

Исправление данных ошибок проводилось путем сокращения количества отправляемых с рабочей станции оператора команд, ошибки GNSS путем программной перезагрузки модуля, а неточные данные с датчиков компенсировались при помощи различных фильтров значений.

3.3. Характеристика программы

Характеристика программы представлена в таблице 7.

Таблица 7 – Характеристика программы

№	Название модуля	Описание модуля	Размер модуля	Кол-во строк
1	PlatformMain.cpp	Модуль основной программы	1,2 кб	36
2	Platform.h	Заголовочный файл библиотеки Platform	4,6 кб	147
3	Platform.cpp	Модуль логики библиотеки Platform	23,9 кб	628
4	Arduino.h	Заголовочный файл библиотеки Arduino	7,2 кб	260
5	Display.cpp	Модуль программы полезной нагрузки «Дисплей»	3,2 кб	116

ЗАКЛЮЧЕНИЕ

В изложенной выпускной квалификационной работе стояла задача упростить работу в условиях опасных для человека, снизить вероятность получения травм, снизить вероятность получения вредных факторов, способных навредить здоровью человека на предприятиях. Опасность обрушения, радиационная, биологическая или химическая угроза, высокие температуры и т.д. все этими проблемы нам предстояло решить благодаря созданию универсальной платформы-носителя с возможностью быстрой замены полезной нагрузки.

На этапе анализа необходимо было продумать, как создать устройство, при помощи которого можно обезопасить нахождение на опасных для человека предприятиях. Далее был проанализирован рынок на наличие аналогов. Прямых аналогов предстоящей разработки не нашлось, а косвенные аналоги имели существенные недостатки, например такие как большая масса и габариты, отсутствие возможности замены полезной нагрузки «на ходу», отсутствие автономных режимов управления и гораздо большую стоимость.

После выявления необходимых требований и анализа поставленных задач предстояло спроектировать аппаратную часть и интерфейс подключения внешних устройств к платформе. Проектирование затрудняла задача автоматической перенастройки платформы внешним устройством под себя и необходимость использовать промышленные стандарты соединений, что успешно решилось использованием разъема D-SUB15 и разработкой встроенного программного модуля-программатора.

Разработка встроенной управляющей программы производилась на языке C++ с вставками ассемблерного кода для увеличения производительности и уменьшения конечного размера программы. Также были использованы аппаратные реализации протоколов UART, SPI, I2C

и блоков арифметики для ускорения работы программы и реализации вычислительного конвейера. Далее были написаны основные функции для взаимодействия с платформой по радиоканалу.

После разработки аппаратная и программная части были протестированы различными способами, включая тесты сигналов, анализ временных задержек, помехоустойчивость радиоканала, устойчивость к механическим воздействиям и устойчивость к ошибкам управления.

Выполненная задача отвечает поставленным методам, основными преимуществами программно-аппаратного комплекса являются: быстрая реакция на команды управления, интуитивное управление, позволяющее работать в визуальном режиме, подробная телеметрия позволяющая работать в полувизуальном режиме, а также работа с любыми типами полезной нагрузки. Из недостатков: отсутствие возможности работы в инструментальном режиме, отсутствие возможности работы по управляющей программе без перенастройки платформы, а также требуются минимальные навыки работы с промышленными роботами.

СПИСОК ИСПОЛЬЗУЕМЫХ МАТЕРИАЛОВ

1) ГОСТ Р 7.0.5-2008 БИБЛИОГРАФИЧЕСКАЯ ССЫЛКА. Общие требования и правила составления.

2) Arduino Library List [Электронный ресурс] URL: https://www.arduinolibraries.info/ (дата обращения 06.02.2022).

3) Avr-libc: Example using the two-wire interface (TWI) [Электронный ресурс] URL: https://www.nongnu.org/avr-libc/user-manual/group__twi__demo.html (дата обращения 04.02.2022).

4) TWI on ATMEGA2560 | AVR Freaks [Электронный ресурс] URL: https://www.avrfreaks.net/forum/twi-atmega2560 (дата обращения 06.02.2022).

5) Электронная свободная энциклопедия [Электронный ресурс] URL: https://ru.wikipedia.org/wiki/I%C2%B2C (дата обращения 07.02.2022).

6) atmega microcontroller registers setting for i2C | AVR Freaks [Электронный ресурс] URL: https://www.avrfreaks.net/forum/atmega-microcontroller-registers-setting-i2cы (дата обращения 15.02.2022).

7) Atmega2560/i2c.h at master · JoshAshby/Atmega2560 [Электронный ресурс] URL: https://github.com/JoshAshby/Atmega2560 (дата обращения 02.03.2022).

8) ATmega2560: работа с UART [Электронный ресурс] URL: http://microsin.net/programming/avr/atmega2560-working-with-uart.html (дата обращения 02.03.2022).

9) Serial Communication using UART on ATMEGA2560 [Электронный ресурс] URL: https://www.avrfreaks.net/forum/serial-communication-using-uart-atmega2560 (дата обращения 03.03.2022).

10) Atmel ATmega640/V-1280/V-1281/V-2560/V-2561/V [Электронный ресурс] URL: https://ww1.microchip.com/downloads/en/devicedoc/atmel-2549-8-bit-avr-

microcontroller-atmega640-1280-1281-2560-2561_datasheet.pdf (дата обращения 03.03.2022).

11) ATmega328PA [Электронный ресурс] URL: https://files.amperka.ru/datasheets/ATmega328.pdf (дата обращения 03.03.2022).

12) SPI on Arduino Mega not working in Assembler [Электронный ресурс] URL: https://www.avrfreaks.net/forum/spi-arduino-mega-not-working-assembler (дата обращения 04.03.2022).

13) avr-sbi-for-spi-ports-h [Электронный ресурс] URL: https://stackoverflow.com/questions/40807274/avr-sbi-for-spi-ports-h-atmega2560 (дата обращения 06.03.2022).

14) ATmega2560 as an SPI slave [Электронный ресурс] URL: https://rpc.gehennom.org/2013/09/atmega2560-as-an-spi-slave/ (дата обращения 08.03.2022).

15) Справка по Ассемблеру для Atmel AVR [Электронный ресурс] URL: https://dfe.karelia.ru/koi/posob/avrlab/avrasm-rus.htm (дата обращения 08.03.2022).

16) Первая программа для AVR микроконтроллера на Ассемблере [Электронный ресурс] URL: https://ph0en1x.net/79-avr-asm-first-program-for-microcontroller.html (дата обращения 10.03.2022).

17) Примеры на Ассемблере для микроконтроллеров Atmel AVR [Электронный ресурс] URL: https://smartep.ru/index.php?page=avr_asm_examples (дата обращения 16.03.2022).

18) <avr/interrupt.h>: Interrupts [Электронный ресурс] URL: https://www.nongnu.org/avr-libc/user-manual/group__avr__interrupts.html (дата обращения 16.03.2022).

19) AVR External Interrupts [Электронный ресурс] URL: https://exploreembedded.com/wiki/AVR_External_Interrupts (дата

обращения 16.03.2022).