

Министерство науки и высшего образования Российской Федерации  
Санкт-Петербургский политехнический университет Петра Великого  
Институт машиностроения, материалов и транспорта  
Высшая школа автоматизации и робототехники

Работа допущена к защите  
Директор ВШАиР

\_\_\_\_\_ А.Н. Волков

«\_\_\_» 2020 г.

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА  
БАКАЛАВРА**

**РАЗРАБОТКА СИСТЕМ РАСПОЗНАВАНИЯ РАСТЕНИЙ И  
УПРАВЛЕНИЯ ВОДЯНЫМИ КЛАПАНАМИ ДЛЯ  
РОБОТИЗИРОВАННОЙ СИСТЕМЫ ПО ВЫБОРОЧНОМУ  
ОПРЫСКИВАНИЮ ПОЛЕЙ ХИМИКАТАМИ**

по направлению подготовки 15.03.06 «Мехатроника и робототехника»  
направленность (профиль) 15.03.06\_04 «Автономные роботы»

Выполнил  
студент гр. 3331506/60401

Н.А. Сафонов

Руководитель

В.Н. Уланов

Консультант  
по нормоконтролю

С.Г. Чупров

Консультант

не предусмотрен

Санкт-Петербург

2020

**САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**ПЕТРА ВЕЛИКОГО**

**Институт машиностроения, материалов и транспорта**

**Высшая школа автоматизации и робототехники**

**УТВЕРЖДАЮ**

**Руководитель ОП**

О.Н. Мацко

« » 20 г.

**ЗАДАНИЕ**

**на выполнение выпускной квалификационной работы**

студенту Сафонову Никите Антоновичу, гр. 3331506/60401

фамилия, имя, отчество (при наличии), номер группы

1. Тема работы: Разработка систем распознавания растений и управления водяными клапанами для роботизированной системы по выборочному опрыскиванию полей химикатами
2. Срок сдачи студентом законченной работы: «18» июня 2020 г.
3. Исходные данные по работе: не предусмотрено
4. Содержание работы (перечень подлежащих разработке вопросов):  
Обзор существующих решений в области выборочного опрыскивания полей  
Проектирование архитектуры системы опрыскивания  
Разработка программного обеспечения для управляющих микроконтроллеров  
Создание и обучение нейросети для распознавания растений  
Изготовление колесного прототипа для отладки системы
5. Перечень графического материала (с указанием обязательных чертежей): не предусмотрено
6. Консультанты по работе: не предусмотрено
7. Дата выдачи задания «11» декабря 2019 г.

Руководитель ВКР \_\_\_\_\_  
(подпись)

**В.Н. Уланов**  
ициалы, фамилия

Задание принял к исполнению «11» декабря 2019 г.  
(дата)

Студент \_\_\_\_\_  
(подпись)

**Н.А.Сафонов**  
ициалы, фамилия

# РЕФЕРАТ

49с., 21 рис., 1 табл., 7 источников, 0 прил.

## КАМЕРА, НЕЙРОННАЯ СЕТЬ, FREERTOS, RS485, СЕЛЬСКОЕ ХОЗЯЙСТВО

В данной работе описывается разработка модулей распознавания растений и управления водяными клапанами. Данные модули используются в составе роботизированной системы для выборочного опрыскивания полей пестицидами.

Модуль распознавания растений состоит из набора камер и компьютера для обработки изображений. Распознавание осуществляется при помощи нейронной сети. Основные проблемы, решенные в рамках разработки данного модуля: обучение нейронной сети, преобразование выхода нейронной сети в последовательность команд на открытие клапанов.

Модуль управления водяными клапанами состоит из набора контроллеров, каждый из которых отвечает за 8 клапанов. Все контроллеры связаны между собой при помощи шины rs485. Основная проблема, решенные в рамках разработки данного модуля: разработать надежный протокол взаимодействия в шине rs485, обладающий низкими процентным соотношением служебных данных к полезным данным.

В рамках работы реализуется набор программ, реализующих алгоритм выборочного опрыскивания. Набор включает в себя программы для микроконтроллеров на языке C++, а также программу для главного компьютера на языке программирования Python.

Для отладки работы программы, а также в целях демонстрации MVP (минимальный работоспособный продукт) разрабатывается прототип, включающий в себя экземпляры каждого модуля системы и способный выполнять условную задачу, аналогичную выборочному опрыскиванию.

# ABSTRACT

49 pages, 21 figures, 1 table, 7 sources, 0 appendices.

CAMERA, NEURAL NETWORK, FREERTOS, RS485, AGRICULTURE

This work describes the development of module for plant recognition and module for the control of water valves. These modules are parts of the robotics system for selective spraying fields with pesticides.

Plant recognition module consists of a set of cameras and a computer for image processing. Image recognition is implemented via neural network. Key problems solved while developing this module are training of the neural network and transforming neural network's output into a sequence of commands for opening and closing valves.

Water valves control module consists of a set of controllers, each controlling 8 valves. All the controllers are connected via rs485 bus. Key problem solved while developing this module is development of a robust protocol for communication over the rs485 bus, that has low percentage of utility data in comparison with useful data.

In this work is being developed a set of programs that implement selective spraying algorithm. This set includes programs for microcontrollers in C++ language and a program for main computer in Python.

For debugging of the program and for demonstration MVP (Minimal Viable Product) it is being developed a prototype. This prototype incorporates instances of all modules of the system and is capable of solving modeled task similar to selective spraying.

# Оглавление

## ВВЕДЕНИЕ

7

<b>1 Синтез системы на основе существующих решений</b>	<b>8</b>
1.1 Исследование существующих решений . . . . .	8
1.2 Анализ наиболее перспективных образцов . . . . .	9
1.3 Синтез системы . . . . .	10
1.4 Теоретическое сравнение аналогов . . . . .	12
1.5 Выводы по главе 1 . . . . .	13
<b>2 Выбор компонентов и протоколов</b>	<b>15</b>
2.1 Выбор компьютера . . . . .	15
2.2 Выбор камер . . . . .	15
2.3 Выбор протокола взаимодействия между модулями . . . . .	16
2.4 Выбор протокола взаимодействия между главным компьютером и шиной . . . . .	17
2.5 Выбор контроллеров для модулей . . . . .	18
<b>3 Теоретическое изложение алгоритма</b>	<b>19</b>
3.1 Квантование поверхности опрыскивания . . . . .	19
3.2 Получение изображений и их предобработка . . . . .	19
3.3 Обоснование выбора нейронной сети в качестве детектора . . . . .	19
3.4 Формат выхода нейросети . . . . .	20
3.5 Архитектура нейросети . . . . .	20
3.6 Использование битовой маски для управления контроллерами . . . . .	21
3.7 Определение мгновенного положения прицепа . . . . .	22
3.8 Передача битовой маски в контроллеры клапанов . . . . .	22
<b>4 Разработка протокола передачи данных</b>	<b>24</b>
4.1 Архитектура сети . . . . .	24
4.2 Требования к протоколу . . . . .	24
4.3 Особенности реализации протокола для шины rs485 . . . . .	25
4.4 Служебные команды . . . . .	25

4.5	Механизм адресации . . . . .	26
4.6	Синхронизация системного времени . . . . .	27
4.7	Передача данных об открытии клапанов . . . . .	27
4.8	Очистка задания . . . . .	28
4.9	Передача данных о положении энкодеров . . . . .	28
<b>5</b>	<b>Этапы разработки прототипа</b>	<b>30</b>
5.1	Этап первый. Реализация протокола над rs-485. . . . .	30
5.1.1	Платы . . . . .	30
5.1.2	Протокол . . . . .	33
5.2	Этап второй. Изготовление колесного прототипа. . . . .	34
5.2.1	Проектирование . . . . .	34
5.2.2	Изготовление . . . . .	36
5.2.3	Тестирование . . . . .	37
5.3	Этап третий. Добавление камеры. . . . .	38
5.3.1	Синхронизация времени камеры . . . . .	38
5.3.2	Подключение и наладка . . . . .	41
5.3.3	Тестирование . . . . .	43
5.4	Этап четвертый. Добавление нейронной сети. . . . .	44
5.5	Выводы по главе 5 . . . . .	46
<b>ЗАКЛЮЧЕНИЕ</b>		<b>47</b>
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ</b>		<b>47</b>

# ВВЕДЕНИЕ

В сельском хозяйстве широко применяется опрыскивание полей различными веществами. Задачи, решаемые с помощью опрыскивания: полив, доставка удобрений, доставка инсектицидов (веществ, защищающих культурные растения от паразитов), доставка гербицидов (веществ, убивающих сорные растения и не повреждающих культурные). На данный момент вещества, в основном, разбрзгиваются равномерно по всей поверхности поля. Однако, для достижения требуемого эффекта, достаточно обрабатывать гербицидами только ростки сорных растений, а инсектицидами - только побеги культурных. Обработка растений данными химикатами производится на начальном этапе роста, при котором все зеленые побеги занимают всего 10% от всей площади поверхности поля. Поскольку химические вещества очень дорогие, а также, несмотря на то, что они безопасны для культурных растений, они все токсичны для окружающей среды, задача снижения расхода химикатов на опрыскивание является очень перспективной.

# 1 Синтез системы на основе существующих решений

## 1.1 Исследование существующих решений

На данный момент существует несколько компаний, разрабатывающих решения по выборочному опрыскиванию полей. Значительная часть подобных компаний находится на стадии прототипа. Среди наиболее перспективных решений - решение "See & Spray" от американской компании Blue river [6] (рис. 1). Кроме того, интересно решение "Smart Spraying" от компании Bosch [7] (рис. 2).



Рис. 1: Решение See & Spray



Рис. 2: Решение Smart Spraying

Прототипы и серийные образцы существующих компаний имеют большое количество оригинальных технических решений. По подсистеме, в которой применяются данные решения, их можно разделить на:

1. решения по поиску сорняков;
2. решения по уничтожению сорняков.

Поиск сорняков обычно ведется при помощи камеры, однако алгоритмы нахождения сорняков могут отличаться. В простейшем случае используется инфракрасная камера, а вид растения определяется по яркости пикселей. Данное решение можно редуцировать до простейшей пары ИК-светодиод и ИК-датчик. Более сложное решение использует классические алгоритмы наподобие каскадов Хаара для определения формы растений на изображении. Наиболее современные алгоритмы используют для тех же целей нейронные сети.

Решения по уничтожению сорняков подразделяются на механические и химические. Механические включают в себя корчеватели различных конфигураций,

вырывающие сорняки из земли. Химические решения представляют собой массивы разбрызгивателей, выборочно включаемые над сорняками.

Несмотря на большое разнообразие экспериментальных прототипов различных компаний, серийные продукты вышеописанных Blue river и Bosch имеют много общего. Данные компании используют исключительно химический способ уничтожения сорняков, поскольку подобные конструкции надежнее, чем механические, а также обладают более высокой производительностью. Также для повышения производительности массив разбрызгивателей делается как можно более широким. Что касается решений по поиску сорняков, то обе компании используют для этих целей камеры с нейросетями.

## 1.2 Анализ наиболее перспективных образцов

В связи с тем, что все существующие решения являются коммерческими продуктами, найти статьи по их устройству в открытом доступе не представляется возможным. Остается только путь реверс-инжиниринга.

Существующие системы представляют собой прицеп, либо обвес на трактор, который состоит из ряда модулей:

1. Бак с химикатами
2. Рама
3. Гидросистема
4. Сенсорная система
5. Система управления

Рассмотрим подробнее каждый модуль.

Бак рассчитан примерно на 500 - 2000 литров. Заполнять такой объем воздухом под давлением опасно, поэтому обычно в баке поддерживается атмосферное давление, а давление в гидросистеме поддерживается насосом и небольшим расширительным бачком. Насос может питаться либо от двигателя трактора через специальный вал, либо от тракторного аккумулятора.

Рама может быть складной, как в случае с Bosch, так и жесткой, как у Blue river.

Гидросистема состоит из системы нагнетания давления - насос и расширительный бачок - и управляемых форсунок. Форсунки представляют собой сопла

особой формы, надетые на электромеханические клапаны.

Сенсорная система представлена камерами, расположенными перед линией форсунок - на расстоянии 1-2 метра.

Система управления состоит из компьютера, а также интерфейса управления, проведенного в кабину водителя.

### 1.3 Синтез системы

Ориентируясь на существующие промышленные системы выборочного опрыскивания и учитывая их достоинства и недостатки разработаем проект собственной системы.

Система управления будет представлять собой компьютер, существующий в единственном экземпляре. Этот компьютер также будет производить нейросетевые вычисления.

Система управления клапанами представляет собой ряд модулей, каждый из которых управляет ограниченным количеством клапанов (оптимальным было выбрано число 8).

Система определения положения состоит из модулей контроля энкодеров, каждый из которых установлен в непосредственной близости от колеса, к которому подключен энкодер.

Система фотографирования земли представляет собой ряд камер, чьи зоны обзора имеют незначительные пересечения (10-20%), что необходимо для предотвращения появления слепых зон.

Раму предлагается сделать модульной. Выбор размеров модулей обусловлен такими параметрами как:

1. Габариты в сложенном состоянии, а также при транспортировке в разобранном состоянии

2. Ширина грядок (поскольку каждый модуль имеет опорное колесо, а колеса должны ехать по междурядью)

Поскольку ширина грядок тех с/х культур, на которых предполагается использовать систему, составляет 35 и 70 см, размер одного модуля должен быть кратен 70 см. Оптimalен размер в 2.1 м. Предполагаемая максимальная ширина прицепа составляет 40 метров, то есть 20 модулей. Предполагаемая ширина

рабочей комплектации составляет от 10 до 20 метров (5-10 модулей).

На один модуль рамы будет приходиться одна камера, а также три модуля управления клапанами - по одному на каждый 70-ти сантиметровый отрезок. Кроме того, на каждое колесо будет приходиться один модуль контроля энкодера.

Схема одного модуля рамы приведена на рисунке 3. Данные модули объединяются в единую систему так, как показано на рисунке 4.

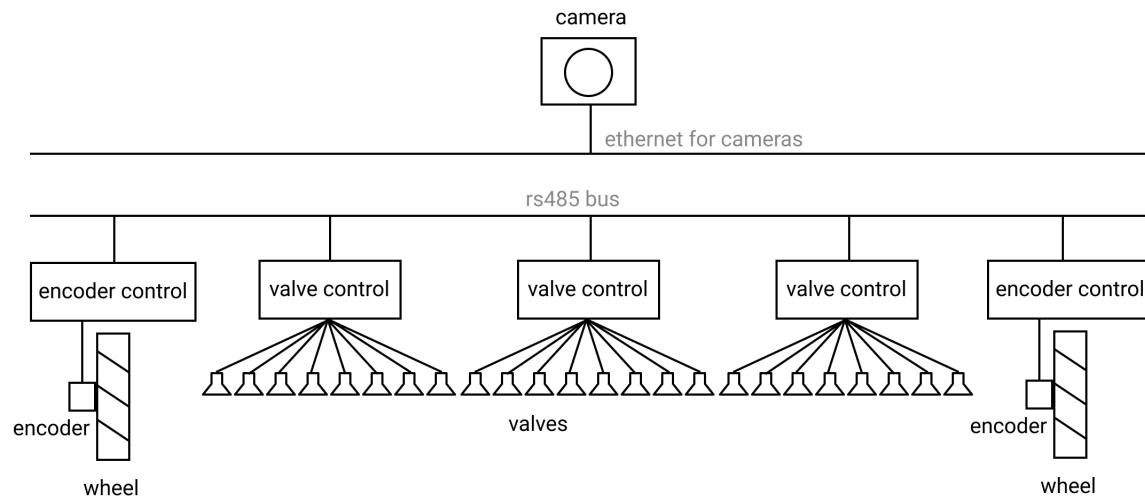


Рис. 3: Схема одного модуля

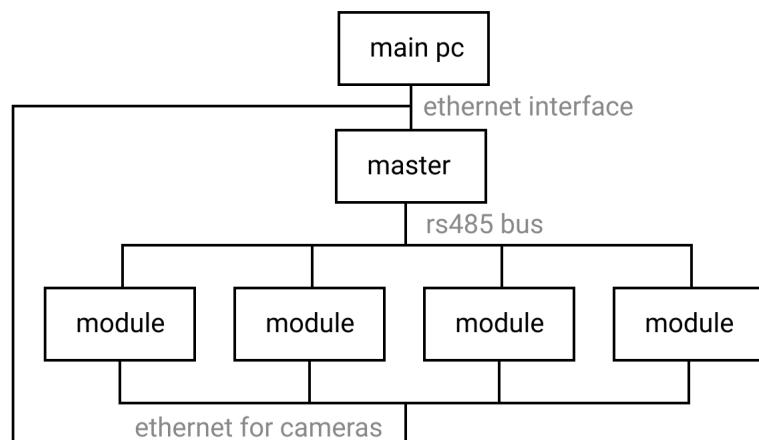


Рис. 4: Схема соединения модулей

Поскольку разработка системы целиком - это очень объемная тема, в данной работе будет описан лишь начальный этап разработки, заключающийся в разработке электронной и информационной частей системы, не затрагивая механическую.

## 1.4 Теоретическое сравнение аналогов

Для конкурентоспособности разрабатываемого решения необходимо удостовериться, что его качества сопоставимы с лидерами рынка. Сравним предлагаемое решение и продуктом "See & Spray" от компании Blue river и "Smart Spraying" от компании Bosch.

Определимся с параметрами, по которым будем вести сравнение:

1. Производительность
2. Точность
3. Удобство транспортировки
4. Нацеленность на российский рынок

Производительность можно вычислить как произведение ширины опрыскивателя на скорость трактора.

Расчет для предлагаемой в статье системы:  $20m \cdot 12km/h \cdot 0.1 = 24ha/h$ , где 12 км/ч - стандартная скорость трактора.

Расчет для "See & Spray":  $8m \cdot 4km/h \cdot 0.1 = 3.2ha/h$

Расчет для "Smart Spraying":  $16m \cdot 10km/h \cdot 0.1 = 16ha/h$

Точность можно оценить через минимальную длину обрабатываемого участка. Важно отметить, что наше решение и решение компании Blue River нацелены на точное опрыскивание единичных сорняков одним гербицидом, в то время как задачей системы Smart Spraying является выявление типа сорняков и применение к ним нужного типа гербицида. При этом система Smart Spraying неспособна доставить гербицид к единичному растению, а распыляет его на группы растений размером от 1 метра. Таким образом, Smart Spraying более эффективна на тех полях, где сорняки растут плотными группами, так как только в этом случае будет отмечаться экономия гербицида.

Наша система является прицепом на трактор. При этом штанги опрыскивателя могут складываться, а сама конструкция имеет небольшой вес. Совокупность этих факторов позволяет легко решить вопрос транспортировки.

Система See & Spray также является прицепом, однако представляет собой монолитную конструкцию. По этой причине, ее невозможно транспортировать по дорогам в собранном виде.

Система Smart Spraying является трактором, оснащенным специальным оборудованием, поэтому способна передвигаться самостоительно. Штанги опрыскивателя складные, так что проблемы с транспортировкой по дорогам отсутствуют.

Система See & Spray в первую очередь нацелена на рынок США, а Smart Spraying - на рынок Евросоюза. Таким образом, только наша система разрабатывается с упором на российский рынок. Однако, поскольку наша система еще не вышла в производство, поставим ей в данной графе ±.

Сведем результаты в таблицу:

Таблица 1: Сравнение аналогов

Характеристика	Наша система	"See & Spray"	"Smart Spraying"
Производительность, Га/час	24	3.2	16
Точность, см	5	10	100
Удобство транспортировки	+	-	+
Национальность на российский рынок	±	-	-

## 1.5 Выводы по главе 1

Был проведен обзор существующих решений, и на его основании были выделены наиболее перспективные конструкции существующих систем.

Исследование данных конструкций позволило разработать функциональную схему конструкции, которая будет разрабатываться в рамках данной работы. Поскольку разработка системы целиком - это очень объемная тема, в данной работе будет описан лишь начальный этап разработки, заключающийся в разработке электронной и информационной частей системы, не затрагивая механическую.

Из соображений конкурентоспособности были подобраны целевые параметры разрабатываемой системы, позволяющие ей работать с производительностью, превосходящей существующие аналоги.

## 2 Выбор компонентов и протоколов

### 2.1 Выбор компьютера

Для большого объема нейросетевых вычислений эффективно использовать видеокарты с поддержкой CUDA от компании Nvidia. Выбор видеокарты по мощности зависит от количества механических модулей. По этой причине целесообразно использовать обычную материнскую плату, на которую устанавливаются компоненты в зависимости от размеров прицепа, нежели использовать встраиваемые системы от Nvidia, такие как Jetson и Xavier.

Библиотеки для использования CUDA работают только под архитектурами x86 (32 и 64 бита) и POWER9. Поддерживаются ОС Windows 10, MacOS, Ubuntu 16.04 и Ubuntu 18.04. Мы будем использовать Ubuntu 18.04 и 64-битный процессор. Важно заметить, что, поскольку Ubuntu не является операционной системой реального времени (RTOS), необходимо также предусмотреть вспомогательный компьютер, чьей основной задачей будет обработка событий в реальном времени.

Наиболее совершенный на данный момент инструмент для создания нейронных сетей - это библиотека Tensorflow. Ее мы и будем использовать для разработки нашей системы.

Будем писать программу обработки видео с камер на Python3, при необходимости оптимизируя узкие места при помощи Cython и C Api.

### 2.2 Выбор камер

Поскольку камеры должны быть установлены на значительном расстоянии (до 20 м) от центра прицепа, где располагается главный компьютер, необходимо использовать надежный способ передачи данных на таком расстоянии. Оптимальным решением будет использовать IP-камеры, подключаемые по кабелю Ethernet. Для упрощения проводки желательно использовать питание камер по технологии PoE.

Для избавления от необходимости производить перспективное преобразование кадра (вычислительно затратная операция), будем устанавливать камеры строго вертикально.

Для того, чтобы избежать потери вычислительных мощностей на избавление

от нелинейных искажений от линзы, будем использовать узкоугольный ( $70^\circ$  по горизонтали) объектив, а также обрезать края изображения (что потребует увеличения областей пересечения кадров с соседних камер).

Требования к качеству изображения - порядка 10 пикселей на 1 см. При ширине зоны ответственности одной камеры в 2.1 метр и пересечении в 40% ширина зоны составит  $2.1 \cdot 1.4 = 2.94 \approx 3$  метра. Тогда при разрешении Full HD на 1 см будет приходиться  $1920/300 = 6.4$  пикселя. Этого вполне достаточно для целей распознавания растений.

Частота снимков подбирается из соображения ненулевого пересечения рабочих зон соседних кадров для максимальной скорости движения. Например, для скорости 15 км/ч (4.17 м/с) и высоты кадра 1 м частота составит не менее 5 кадров в секунду. Такие требования выполнит почти любая современная камера.

Для того, чтобы изображение не было смазанным, необходимо, чтобы время экспозиции было минимально. Для скорости 15 км/ч время экспозиции должно быть порядка 1 мс. Для того, чтобы на такой малой выдержке получались хорошие кадры, необходимо, чтобы матрица обладала очень высокой чувствительностью.

Поскольку камера будет работать в уличных условиях, необходимо, чтобы она поддерживала HDR. В противном случае кадр может быть засвечен от солнечного блика.

Кроме того, камера должна хорошо работать ночью в условиях искусственной подсветки, поскольку большая часть работы по опрыскиванию производится ночью из-за меньшей скорости испарения разбрзганной жидкости.

Для корпуса камеры требования по герметичности - не менее IP66, в идеале - IP68.

**Вывод:** по результатам сравнения камер было установлено, что оптимальный вариант по соотношению цена/качество - это уличные камеры от компании Hikvision из линейки Dark Fighter.

## 2.3 Выбор протокола взаимодействия между модулями

Поскольку прицеп представляет из себя объект с линейной топологией, логично объединить все контроллеры, расположенные на нем, в сеть с топологией

"шина". Протокол Ethernet поддерживает обмен данными по одной шине, однако он плохо подходит для обмена короткими сообщениями (минимальная длина пакета - 64 байта), а также имеет много лишних возможностей, которые не используются в нашем проекте. Поэтому, было принято решение использовать более простой протокол RS-485.

Поскольку протокол RS-485 полудуплексный, то при большом количестве устройств домен коллизий станет слишком велик для нормальной передачи данных. Поэтому, было принято решение сделать одно устройство мастером, который всегда будет инициировать передачу данных.

На шину возможно посадить большое количество разнотипных устройств. В первую очередь, это будут и контроллеры клапанов и энкодеров. Также в дальнейшем к шине можно присоединить контроллеры габаритных огней, подсветки, очистки объективов камер и другие устройства.

## 2.4 Выбор протокола взаимодействия между главным компьютером и шиной

Главный компьютер соединяется с шиной RS-485 через Мастера.

Первоначально планировалось передавать данные между главным компьютером и мастером при помощи протокола RS-232, однако, как оказалось, драйверы COM-порта в Ubuntu (и, судя по всему, также и в остальных ОС) слишком медленные. Это связано с низкой востребованностью данного интерфейса в наше время. Поэтому, было решено использовать протокол Ethernet для физической связи.

Поверх Ethernet было решено использовать UDP, так как он проще в реализации, более предсказуемый, а также позволяет реализовать свой контроль доставки пакетов, который будет оптимизирован под нужды приложения, в отличие от TCP, где контроль пакетов работает из коробки, но при этом добавляет неконтролируемые задержки и другие проблемы.

Поскольку на Мастере сходятся все информационные потоки, было решено сделать его также главным обработчиком событий в реальном времени. Таким образом, появилась необходимость поднять на Мастере RTOS. Было решено использовать для этих целей FreeRTOS.

Таким образом, главный компьютер обменивается с Мастером асинхронными

сообщениями, адресованными устройствам в шине, а все синхронные задачи (отправка периодических сообщений в шину, формирование очереди пакетов и их последовательная отправка и т. д.) становятся зоной ответственности Мастера.

## 2.5 Выбор контроллеров для модулей

1. Для мастера было решено выбрать микроконтроллер stm32f103c8t6 (архитектура ARM, реализация - отладочная плата "blue pill"). Мощностей данного контроллера достаточно для установки на него FreeRTOS, а также библиотеки UIP, реализующей стек TCP/IP для взаимодействия с компьютером.

1\*. Поскольку на плате stm32 нет встроенного контроллера Ethernet, к нему по SPI подключается готовый модуль Ethernet на базе чипа enc28j60.

2. Контроллеры клапанов изначально были реализованы на базе Arduino Mega Pro 2560 (архитектура AVR), однако планируется также заменить их на stm32f103c8t6.

3. Контроллеры клапанов изначально были реализованы на базе Arduino Nano (архитектура AVR), однако планируется также заменить их на stm32f103c8t6.

### **3 Теоретическое изложение алгоритма**

#### **3.1 Квантование поверхности опрыскивания**

В силу конструктивных особенностей системы опрыскивания минимальной единицей опрыскивания является прямоугольник, ширина которого равна ширине форсунки, а длина - произведению скорости движения на длительность открытого состояния клапана. Таким образом, всю поверхность опрыскивания можно представить в виде прямоугольной сетки.

При этом ширина ячеек является фиксированной в силу конструкции, в то время как длина может меняться. Поскольку состояние ячейки бинарно (либо форсунка была открыта и ячейку обработали, либо нет), то погрешность наложения битовой маски по оси x (параллельной направлению движения) не может быть меньше половины длины ячейки. Поэтому, необходимо минимизировать длину ячейки для повышения точности. Предварительно примем длину ячейки равной 5 см.

#### **3.2 Получение изображений и их предобратка**

Поскольку ранее мы расположили камеру строго горизонтально, вся предобработка сводится к вырезанию из изображения рабочего прямоугольника. При этом вертикальные границы вырезаемого прямоугольника выравниваются по краям форсунок, а горизонтальные - по целому числу длин ячеек относительно линии форсунок.

#### **3.3 Обоснование выбора нейронной сети в качестве детектора**

В данный момент в задаче выборочного опрыскивания применяются различные подходы к детектированию регионов интереса, нуждающихся в опрыскивании.

В случае, когда растения выросли и составляют сплошной покров, широко применяются ультразвуковые датчики, позволяющие держать разбрзгиватели на нужном расстоянии от вершин колосьев. Данный метод неприменим в случае

предлагаемой в статье системы, поскольку описываемая система специализирована под опрыскивание растений на ранних стадиях развития.

Существует подход, при котором опрыскиваются все растения, но участки, свободные от растительности, не опрыскиваются в целях экономии пестицида. В таком случае часто используют инфракрасные датчики, поскольку растения становятся контрастными и легко различимыми в инфракрасном спектре. Данный подход также не годится, так как не позволяет отличить культурные растения от сорняков.

Наиболее гибкий подход к определению сорняков и культурных растений - это использование нейронных сетей в качестве классификаторов. Значительное превосходство нейросетей над другими классификаторами, такими как каскад Хаара, наличие достаточных вычислительных мощностей для работы в реальном времени, а также наличие широкого спектра готовых архитектур делает нейросети наиболее мощным инструментом для решения задачи детектирования объектов. Так, один из конкурентов, компания Blue River использует в своем продукте нейронные сети для классификации растений.

### **3.4 Формат выхода нейросети**

Результатом обработки одного кадра нейросетью должна быть битовая маска, соответствующая участку сетки, снятому камерой. То есть, если ширина рабочей области соответствует 24 форсункам, а высота - 100 см при длине ячейки 5 см, то в результате должна получиться битовая маска 24x20.

### **3.5 Архитектура нейросети**

В данной задаче нейросеть должна решать задачу сегментации с понижением разрешения битовой маски относительно размера входного изображения. Среди возможных архитектур: U-Net [1], SegNet, Mask-RCNN [2] и YOLO [3] с кастомной сегментирующей головкой. В принципе, нейросети, заточенные под object detection, также показывают лучшие результаты в задачах сегментации по сравнению с классическими сетями энкодер-декодер (U-Net, SegNet). Однако, они требуют значительно больших вычислительных мощностей и труднее обучаются.

Поскольку имеет место понижение размерности входного изображения, разумно использовать архитектуру, аналогичную SegNet, но без декодера. Другими словами, просто набор сверточных слоев и MaxPooling слоев. Данная архитектура является наиболее простой, однако, вполне способна решать поставленную задачу.

Для качественной сегментации требуется большой датасет. Тем не менее, используя имеющийся датасет из 10 фотографий мне удалось получить качество распознавания, достаточное, чтобы отличать как минимум растения от земли. Даже это уже ведет к сокращению площади опрыскивания в 7 раз. Что же касается конечной задачи - умения отличать ряд культурных растений и сорняков друг от друга - то здесь понадобится датасет на 10-100к изображений.

### 3.6 Использование битовой маски для управления контроллерами

Для того, чтобы опрыскать участок, снятый камерой, необходимо перевести битовую маску в команды на открытие-закрытие форсунок. Все сводится к тому, чтобы в нужный момент времени выставлять нужный паттерн открытия на форсунках. Так как камеры вынесены вперед относительно форсунок, между моментом снятия кадра и моментом, когда линия форсунок пересекает первую строчку битовой маски, соответствующей данному кадру, проходит некоторое время. Для расстояния 1 метр и скорости 15 км/ч эта задержка составляет  $1/(15/3.6) = 0.24$  секунды. За это время программе нужно успеть распознать изображение, а также передать его в соответствующие контроллеры клапанов (как было описано ранее, каждая камера соответствует 3 контроллерам клапанов). Если времени на распознавание недостаточно, можно увеличить расстояние от камеры до форсунок, однако нежелательно делать его больше, чем 2 метра, так как в таком случае конструкция станет слишком громоздкой.

Поскольку момент переключения - это тот момент, когда линия форсунок находится на границе раздела ячеек, необходимо реализовать какой-то механизм определения мгновенного положения прицепа. Зная положение прицепа в момент, когда был сделан снимок, и текущее положение, мы сможем определить, на какой строчке битовой маски находится линия форсунок в данный момент.

### **3.7 Определение мгновенного положения прицепа**

Существует способ измерения положения через оптический поток. Приведу подробное пояснение, почему данный метод не подходит в данной задаче. Во-первых, он требует более высокого числа кадров в секунду (так как пересечение между кадрами должно быть не менее 50%) и дополнительных вычислительных мощностей. Во-вторых, этот метод довольно грубый (одно измерение на 50 см при пересечении кадров 50%). В-третьих, из-за того, данный подход требует вычислений на компьютере (то есть под управлением ОС общего назначения, а не RTOS), задержка получения точной координаты контроллером клапанов составляет десятки миллисекунд.

Вместо этого было принято решение использовать контроллеры энкодеров, которые с частотой порядка 10-100 Гц посыпают в шину rs485 значение энкодеров (на самом деле они посыпают данные не сами, а только после запроса от Мастера). Качественные характеристики данного решения на порядки выше:

1. Разрешение энкодеров - 1-2 см на деление.
2. Задержка передачи данных - менее 1 мс.
3. Ресурсы компьютера при этом не используются.

### **3.8 Передача битовой маски в контроллеры клапанов**

Передача данных в контроллеры клапанов происходит следующим образом:

1. Регулярно происходит синхронизация системного времени между отправителем и получателем данных (то есть контроллер клапана должен знать системное время главного компьютера, а обратное не требуется).
2. При получении кадра с камеры, компьютер сохраняет момент его снятия.
3. Компьютер отправляет битовую маску (то есть набор состояний клапанов, смещенных друг от друга на длину ячейки) и момент ее снятия.
4. Контроллер по метке времени достает из памяти положение прицепа, которое было в тот момент времени.
5. Зная абсолютное положение прицепа в момент снятия битовой маски и относительное смещение для каждой строки битовой маски, контроллер кладет соответствующую строку в циклический массив заданий, связанный с абсолютным положением прицепа.

6. Когда положение прицепа меняется, контроллер передвигает по массиву заданий указатель на текущее положение и достает из массива текущее задание (то есть строку битовой маски), после чего применяет это задание (выставляет биты открытия/закрытия на пинах, управляющих клапанами).

## **4 Разработка протокола передачи данных**

### **4.1 Архитектура сети**

Как было описано ранее, на физическом уровне система состоит из двух подсетей:

1. Сеть Ethernet для связи между главным компьютером и Мастером.
2. Шина RS-485 для связи контроллеров с Мастером, а также контроллеров между собой (несмотря на то, что передачу инициирует Мастер, все сообщения "слышат" все участники шины).

Основные направления передачи данных:

1. От компьютера Мастеру и обратно
2. От компьютера контроллеру клапанов
3. От контроллера энкодера контроллеру клапанов
4. От Мастера всем устройствам (периодические сообщения)

### **4.2 Требования к протоколу**

1. Реализовать служебные команды: проверка соединения (ping запрос), сведения об устройстве (там, где необходимо идентифицировать подключенное устройство), передача отладочной информации.
2. Поскольку среди направлений передачи существуют передачи к конкретным экземплярам однотипных устройств, необходимо реализовать механизм адресации.
3. Реализовать механизм синхронизации системного времени.
4. Реализовать команду для передачи данных об открытии клапанов (битовой маски) в контроллер клапанов
5. Реализовать команду очистки задания на открытие клапанов (в целях экстренной остановки работы, например, при потере соединения с главным компьютером)
6. Реализовать механизм передачи данных о положении от контроллеров энкодеров в контроллеры клапанов

## 4.3 Особенности реализации протокола для шины rs485

Шина rs485 является полудуплексной. Поэтому, во избежание коллизий, сеанс передачи всегда начинает мастер. При этом некоторые команды бывают трех типов:

1. Не требующие ответа или подтверждения о получении (обычно отправляются на широковещательный адрес).

2. Требующие ответа или подтверждения о получении

3. Команды, вызывающие цепочку сообщений от разных устройств. Для каждой такой команды должна быть приведена гарантия того, что данная цепочка не имеет циклов и, следовательно, закончится за конечное время.

Формат команды на уровне пакета выглядит так: <packet header><packet body><crc>, где packet header - заголовок пакета - 3 байта; packet body - тело пакета; crc - контрольная сумма от всего пакета - 1 байт.

Как видно, пакет не содержит поля "длина пакета". Вместо этого, длина пакета определяется по периоду молчания после отправки последнего байта пакета (около 2-х длительностей байта).

Некоторые пакеты не имеют тела. Таким образом, минимальная длина пакета - 4 байта.

Максимальная длина пакета ограничена 128 байтами для того, чтобы длинные пакеты не занимали общий канал связи слишком долго.

Формат заголовка пакета: <address to><address from><cmd>, где address to - адрес получателя - 1 байт; address from - адрес отправителя - 1 байт; cmd - команда - 1 байт.

## 4.4 Служебные команды

### Команда User message

Назначение: передача отладочных и статусных сообщений.

Формат команды: <address to><address from><U><...><crc>

Тип команды: является командой ответа на запрос

### Команда Ping

Назначение: проверка связи с устройством

Формат команды: <address to><address from><P><crc>

Тип команды: требует ответа

Формат ответа: <address to><address from><U><ok><crc>

### Команда Device type

Назначение: определить тип устройства, зная его адрес

Формат команды: <address to><address from><D><crc>

Тип команды: требует ответа

Формат ответа: <address to><address from><U><device type><crc> где device type - тип устройства - 1 байт. Например, для контроллера клапанов тип устройства равен "v а для контроллера энкодера - "e"

## 4.5 Механизм адресации

Так как задание каждого устройства адреса на этапе прошивки в виде уникального серийного номера - слишком негибкий способ, была предложена следующая альтернатива: наряду с проводами питания и протокола RS-485 каждые 2 соседних устройства в шине соединяются дополнительным проводом наподобие SS в протоколе SPI. Таким образом, каждое устройство имеет дополнительных 2 пина: address pin left (APL) и address pin right (APR), причем пин APL связан с пином APR левого устройства и наоборот.

Операция динамического присвоения адресов выглядит следующим образом:

1. Мастер выставляет на пин APL логическую единицу.

2. Мастер отправляет команду "присвоить адрес 127".

3. Устройство, на чьем APL установлена логическая единица, запоминает свой новый адрес - 127, после чего выставляет на свой пин APL логическую единицу и отправляет сообщение "присвоить адрес 126".

4. Так происходит до тех пор, пока все устройства слева от Мастера не получат свои адреса. Таким образом, адреса влево от Мастера идут в порядке убывания (127->126->125->...).

5. Мастер выставляет на пин APR логическую единицу.

6. Мастер отправляет команду "присвоить адрес 128".

7. Устройство, на чьем APR установлена логическая единица, запоминает свой новый адрес - 128, после чего выставляет на свой пин APR логическую единицу и отправляет сообщение "присвоить адрес 129".

8. Так происходит до тех пор, пока все устройства справа от Мастера не получат свои адреса. Таким образом, адреса вправо от Мастера идут в порядке возрастания ( $128->129->130->\dots$ ). Если же посмотреть на все устройства в шине, то их адреса слева направо будут идти в порядке возрастания подряд ( $\dots->125->126->127->128->129->130->\dots$ ).

После того, как все адреса распределены, Мастер может, прочитав все операции присвоения адреса, легко построить таблицу подключения устройств. При этом часть адресов зарезервированы под особые нужды:

1. адрес 0 - начальный адрес устройства после перезагрузки
2. адрес 1 - адрес Мастера
3. адрес 255 - широковещательный адрес

#### **Команда Address assign**

Назначение: динамическое присвоение адреса.

Формат команды: <BROADCAST ADDRESS><address from><A><address to assign><crc> где address to assign - адрес, который нужно присвоить - 1 байт.

Тип команды: команда, вызывающая цепочку сообщений

## **4.6 Синхронизация системного времени**

#### **Команда Sync timestamp**

Назначение: передать свое системное время

Формат команды: <BROADCAST ADDRESS><address from><S><timestamp><crc> где timestamp - системное время в микросекундах - 4 байта (переполняется каждые 71.5 минут).

Тип команды: широковещательная, не требует ответа

## **4.7 Передача данных об открытии клапанов**

#### **Команда Set valves future data**

Назначение: передать задание на открытие клапанов

Формат команды: <address to><address from><V><timestamp><relative pos><valves states><crc> где timestamp - время снятия фотографии в микросекундах - 4 байта; relative pos - смещение линии первого задания относительно линии клапанов в длинах ячейки - 1 байт; valves states - задания значений клапанов - каждый байт

соответствует 8 битам положений 8 клапанов в определенный момент - от 1 байта до максимальной длины пакета

Тип команды: требует ответа

Формат ответа: <address to><address from><U><ok><crc>

## 4.8 Очистка задания

### **Команда Clear**

Назначение: очистить задание на одном устройстве

Формат команды: <address to><address from><C><crc>

Тип команды: требует ответа

Формат ответа: <address to><address from><U><ok><crc>

### **Команда Clear broadcast**

Назначение: очистить задание на всех устройствах

Формат команды: <BROADCAST ADDRESS><address from><C><crc>

Тип команды: широковещательная, не требует ответа

## 4.9 Передача данных о положении энкодеров

### **Команда Encoder start**

Назначение: инициация процесса отчета контроллеров энкодеров.

Формат команды: <BROADCAST ADDRESS><address from><e><crc>

Тип команды: команда, вызывающая цепочку сообщений

### **Команда Encoder**

Назначение: отчет контроллера энкодера.

Формат команды: <BROADCAST ADDRESS><address from><E><encoder pos><crc> где encoder pos - положение энкодера в данный момент времени - 1 байт.

Тип команды: команда, вызывающая цепочку сообщений

### **Последовательность команд**

1. Мастер отправляет команду **Encoder start**.
2. Контроллер энкодера, чей адрес наименьший в сети (заметим, что крайние модули на шине - всегда контроллеры энкодеров), отправляет команду **Encoder**.

3. Контроллер энкодера, чей адрес на 3 больше, чем адрес предыдущего отправителя (заметим, что каждый четвертый модуль - это контроллер энкодера), отправляет команду **Encoder**.

4. Шаг 3 повторяется, пока контроллеры энкодеров не закончатся.

## 5 Этапы разработки прототипа

### 5.1 Этап первый. Реализация протокола над rs-485.

#### 5.1.1 Платы

Для реализации протокола необходимо было разработать и реализовать несколько плат: мастер, контроллер энкодеров и контроллер клапанов. В идеале все платы должны быть основаны на отладочной плате stm32f103 в целях единообразности, однако понимание этого пришло только после того, как были изготовлены первые прототипы.

Платы мастера и контроллеров клапанов были изготовлены на базе arduino mega, а плата контроллера энкодеров - на базе arduino nano.

После реализации первых прототипов оказалось, что только плата мастера обладает недостаточной вычислительной мощностью, в следствие чего она была переработана с заменой центрального микроконтроллера на stm32f103. Остальные платы были оставлены без изменений.

Рассмотрим каждую плату более подробно.

На Рис. 5 изображена плата мастера. Она состоит из микроконтроллера stm32f103, модуля rs485 и модуля Ethernet. Также плата имеет два отладочных светодиода и два "адресных" разъема, используемых для присвоения адресов в цепи устройств (более подробно этот механизм описан в предыдущем НИР-е).

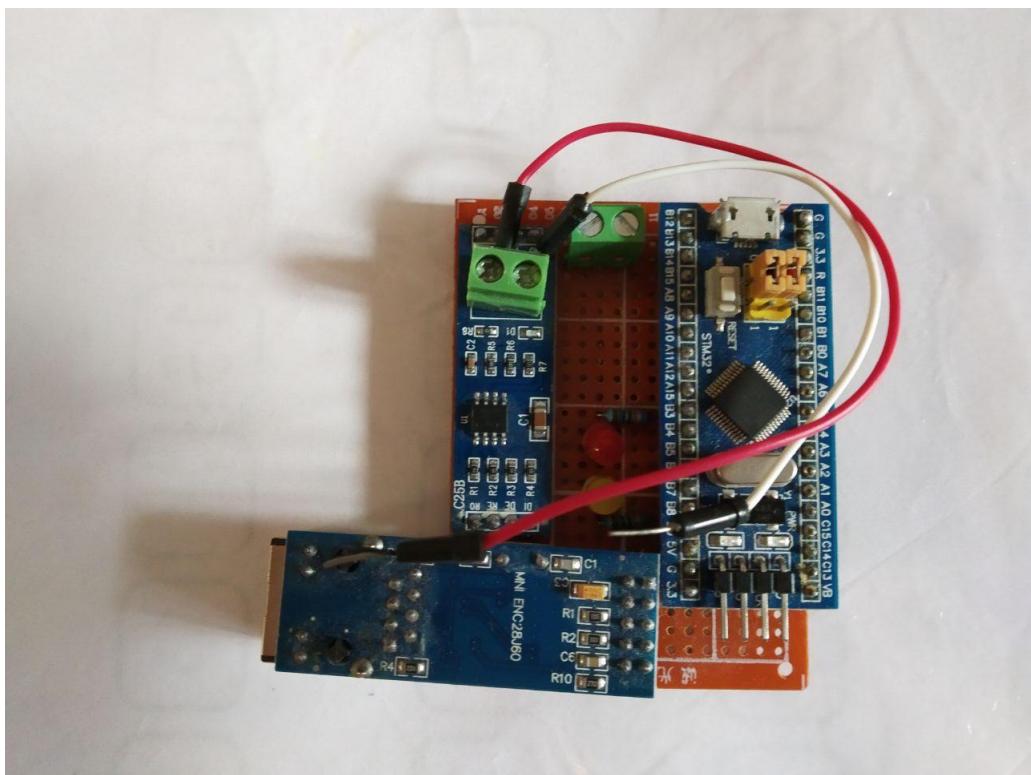


Рис. 5: Плата мастера

На Рис. 6 изображена плата контроллера энкодера. Она также имеет два адресных пина и два отладочных светодиода. Модуль rs485 идентичен используемому в плате мастера. Плата управляется микроконтроллером arduino nano. Данная плата имеет два разъема для подключения инкрементальных энкодеров.

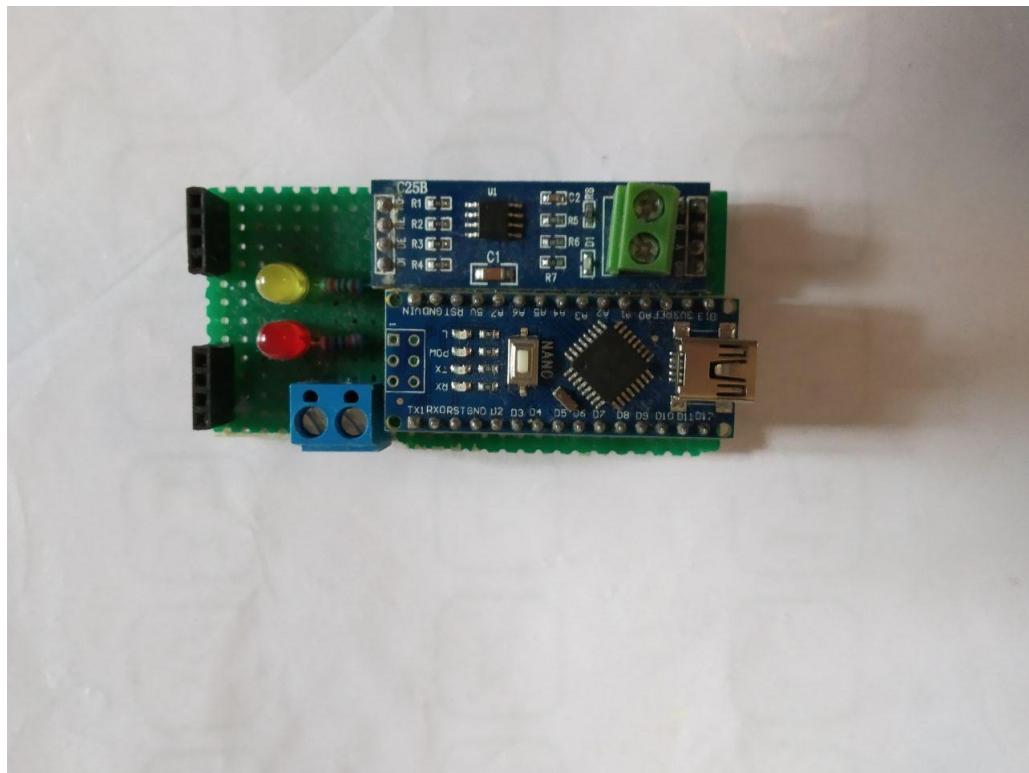


Рис. 6: Плата контроллера энкодеров

На Рис. 7 изображена плата контроллера клапанов. Стандартными элементами платы являются два отладочных светодиода, адресные пины и модуль rs485. Плата управляется микроконтроллером arduino pro mega. Для управления восемью клапанами на плате установлены восемь силовых ключей. Каждый ключ снабжен индикаторным светодиодом, позволяющим определить, открыт ли соответствующий канал. Силовые ключи управляются внешним питанием, подводимым к клеммному разъему справа внизу.

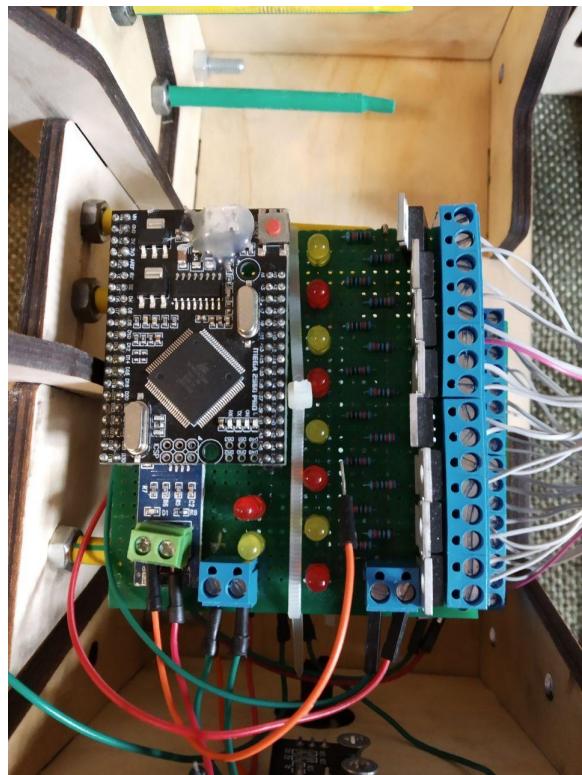


Рис. 7: Плата контроллера клапанов

### 5.1.2 Протокол

Протокол был реализован в виде трех программных файлов - для каждого типа плат. Платы одного типа содержат одинаковые программы.

Для отладки протокола была собрана схема, в которой все платы были подключены в шину rs485, а пины адреса образовывали цепь, по которой передавался адрес от мастера.



Рис. 8: Тестовый стенд

Данный тестовый стенд позволил отладить процедуры:

- начального присвоения адреса устройствам,
- отправки значений "клапанов" с компьютера в контроллер
- отправки значений энкодеров в контроллеры клапанов и в компьютер.

Однако, в связи с тем, что стенд оставался неподвижным, он не позволял проверить задержку передачи данных и синхронность работы модулей. Для дальнейших экспериментов был необходим колесный прототип.

## 5.2 Этап второй. Изготовление колесного прототипа.

### 5.2.1 Проектирование

Прототип был спроектирован при помощи SolidWorks (Рис. 9).

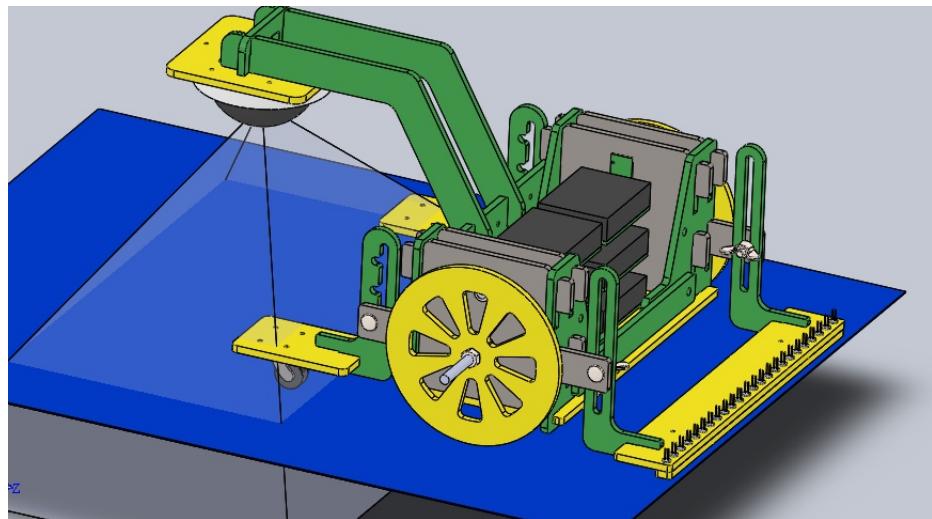


Рис. 9: Модель в SolidWorks

Большая часть конструкционных элементов прототипа планировалось вырезать лазером из листовой 4-мм фанеры. Для сборки применены болты м3 и саморезы. В местах быстроразборных соединений использованы мебельные винты (плоская шляпка, резьба заканчивается квадратным стержнем) с барашковыми гайками.

Модуль колес состоит из пары больших колес, имеющих три исполнения: 15, 20 и 25 см диаметром соответственно, а также пары маленьких мебельных самоустанавливающихся колес. Для обеспечения горизонтальности корпуса платформы, к которым крепятся маленькие колеса, регулируются по высоте при помощи выемок, расположенных на трех фиксированных высотах. После изготовления прототипа было принято решение остановиться на колесах 20 см, как на наиболее удобных.

В качестве осей больших колес выступают шпильки м8, закрепленные на радиальноупорных подшипниках. Помимо колес на осях закреплены зубчатые колеса, жестко связанные с колесами. Зубчатые колеса входят в зацепление с шестернями с передаточным отношением 3:1. Шестерни закреплены на осях инкрементальных энкодеров (Рис. 10). Данный энкодер имеет 20 импульсов на оборот, следовательно, при использовании колес диаметром 20 см на один импульс энкодера будет приходиться  $\frac{20\pi}{20 \cdot 3} \approx 1$  см.



Рис. 10: Модуль энкодера

Внутри корпуса прототипа отведено место под электронику. Электроника крепится к горизонтальным шпилькам м4, выступающим из передней стенки наподобие полок.

Сзади располагается гребенка под светодиоды, которая будет имитировать ряд форсунок. Гребенка может непрерывно регулироваться по высоте при помощи направляющих, фиксируемых мебельными винтами с баращковыми гайками.

Спереди располагается Г-образная рама, к которой должна крепиться камера. Закрепленная таким образом, камера способна фотографировать паттерн на пути движения тележки. Рама сделана съемной в целях удобства транспортирования прототипа.

### 5.2.2 Изготовление

Деревянные компоненты были вырезаны на лазерном резаке в ФабЛабе Политеха. Собранный корпус представлен на Рис. 11



Рис. 11: Деревянный каркас

После сборки на прототип были установлены ранее изготовленные платы. В гребенку были вмонтированы шестнадцать светодиодов. Светодиоды были подключены к двум контроллерам клапанов. Энкодеры были подключены к плате контроллера энкодеров.

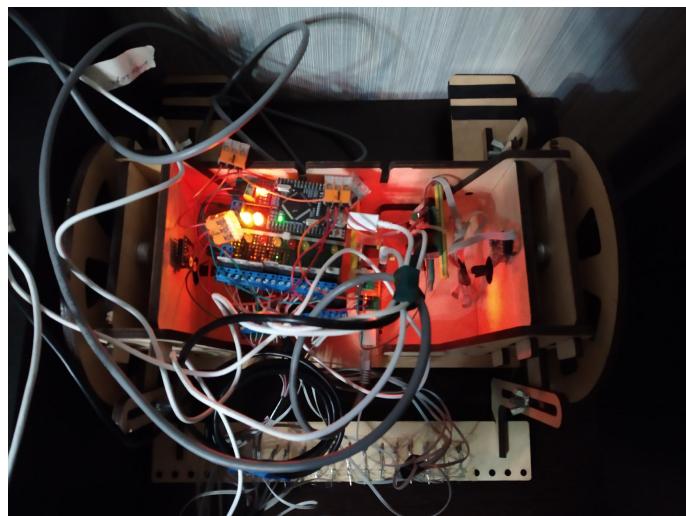


Рис. 12: Прототип со смонтированной электроникой

### 5.2.3 Тестирование

Для тестирования электроники была написана программа для компьютера, отправляющая в плату мастера фиксированный паттерн в реальном времени.

Волочение тележки по полу приводило к изменению показаний энкодеров, и система реагировала на это изменением положений открытия клапанов, что можно было считать по изменению паттерна включения светодиодов.

Для тестирования был выбран крестообразный паттерн. Точность воспроизведения паттерна в пространстве-времени можно определить при помощи съемки с большой выдержкой, либо при помощи записи видео с последующим совмещением кадров. Был выбран второй метод.

На Рис. 13 изображен результат совмещения кадров при съемке прототипа во время воспроизведения паттерна. Прототип двигался слева направо, следовательно, горизонтальную ось на Рис. 13 можно считать осью времени.

Линия светодиодов расположена вертикально и состоит из 16 светодиодов. За нижние 8 светодиодов отвечает одна плата, за верхние - другая. Как можно видеть из Рис. 13, обе платы отрабатывают синхронно.

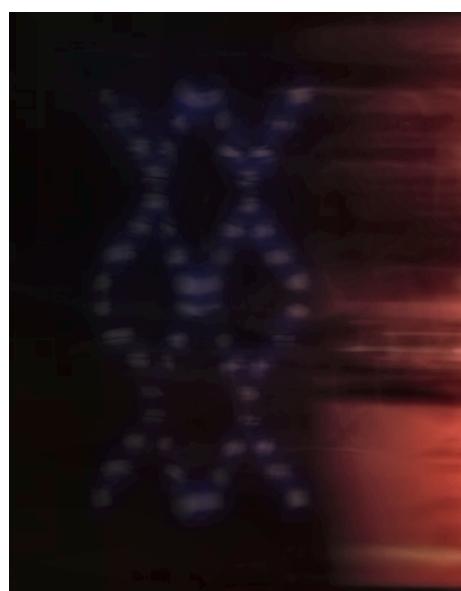


Рис. 13: Паттерн 1 - косая линия

### 5.3 Этап третий. Добавление камеры.

#### 5.3.1 Синхронизация времени камеры

##### Теория

Задача данного этапа - синхронизировать данные с камеры с выходами контроллеров клапанов. Поскольку синхронизация времени компьютера и времени контроллеров уже была реализована на прошлом этапе, остается только синхронизировать время камеры со временем компьютера.

Известно, что цифровая камера снимает кадры с фиксированной частотой - 25

fps, однако передача кадров по сети занимает некоторое время и кадры приходят с разной задержкой.

Таким образом, для того, чтобы знать точное время съемки любого кадра, достаточно знать точное время съемки одного опорного кадра и порядковый номер целевого кадра.

Чтобы достичь этого, необходимо подать в камеру стимул, для которого известно его точное время появления, а затем распознать этот стимул на изображении.

## Подготовка

Для этих целей была создана плата, управляющая 14-разрядным линейным индикатором. Индикатор показывал время в бинарном виде: крайний правый разряд отвечал за миллисекунды, крайний левый - за восьмерки секунд. Таким образом, время переполнения таймера составляло примерно 16 секунд.

Имея серию кадров с камеры, где счетчик попадал в камеру, можно было распознать числа, изображенные на счетчике, и для каждого кадра получить его метку времени. Для повышения точности измерения следует усреднить метки с нескольких кадров, зная, каково время между кадрами.

После того, как будет получена метка кадра в относительном времени счетчика, зная время начала отсчета счетчика можно будет перевести ее в метку реального времени.

Теперь, для любой последующей непрерывной серии кадров можно будет определить ее метку времени. Данное ограничение означает, что каждый раз в начале программы камеру необходимо заново синхронизировать. Следовательно, калибровочный индикатор нужно закрепить на корпусе в поле зрения камеры.

## Эксперимент

Для постановки эксперимента была написана программа для платы, запускающая отсчет индикатора с нуля по сигналу с СОМ-порта, а также скрипт на python для управления платой и захвата видео с камеры DS-I102.

На Рис. 14 изображен способ снятия данных с таймера. Таймер состоит из 16 светодиодов. Крайние светодиоды всегда горят. Положение таймера в кадре неподвижно. Центры крайних светодиодов в кадре отмечается вручную, положение остальных светодиодов определяется линейной интерполяцией. В каждой точке производится подсчет яркости окрестности, отфильтрованной по Гауссу. Если яр-

кость выше порога, то светодиод считается включенным, иначе - выключенным.

Включенные светодиоды на изображении обозначены красными точками, выключенные - синими.



Рис. 14: Способ распознавания значения таймера

После записи видео с камеры стало ясно, что последние 4 разряда не являются значимыми из-за ограничения по выдержке камеры. Оптимальной выдержкой камеры являлась выдержка в 10 мс, следовательно, за время выдержки в разрядах с периодом 2,4,8 мс успевал пройти полный период включения-выключения, а в разряде с периодом 16 мс - половина периода. Таким образом, первый значимый разряд имел период 32 мс и точность определения метки времени кадра составляла  $\pm 16$  мс.

Оказалось, что метод определения метки времени по таймеру в кадре довольно стабилен. На графике зависимости времени от номера кадра (Рис. 15) видно, что этот график - прямая линия, как и должно быть в случае, если частота кадров равномерна. По углу наклона кривой на участке 40-140 определим, что за 100 кадров прошло 4000 мс, то есть  $fps = 100/4 = 25$ , что соответствует действительности.

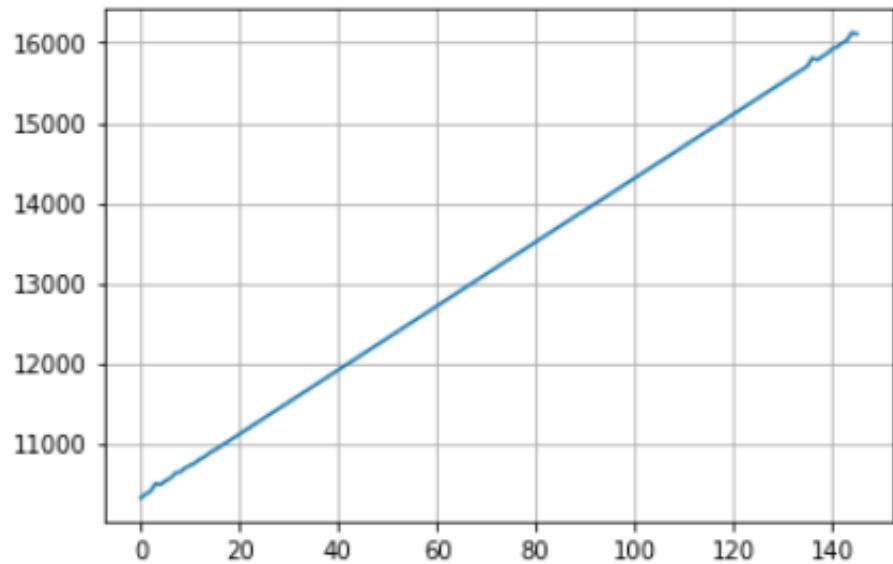


Рис. 15: График зависимости времени от номера кадра

### 5.3.2 Подключение и наладка

По окончании экспериментов с синхронизации камера была установлена на прототип. Камера была подключена при помощи двух кабелей - питание 12 В и кабель Ethernet. В связи с увеличением количества проводов, ведущих к прототипу они были соединены в жгут.

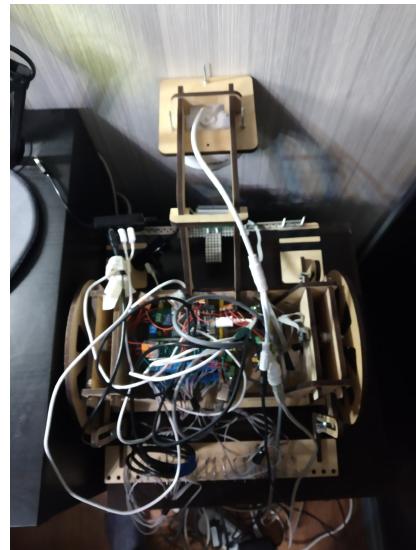


Рис. 16: Камера установлена на прототипе

Под камерой в области ее видимости был смонтирован индикатор.

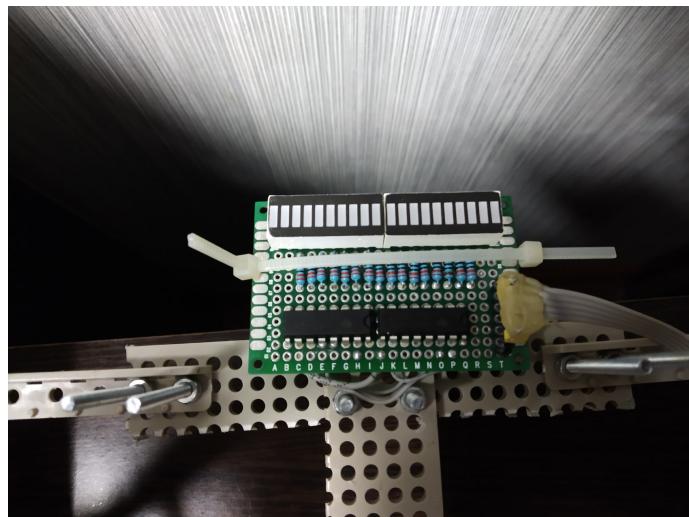


Рис. 17: Индикатор установлен на прототипе



Рис. 18: Индикатор в поле зрения камеры

Для синхронизации индикатора со временем системы к плате управления индикатором был подключен конвертер rs-485, и модуль индикатора был включен в общую шину rs-485 в пассивном режиме. Модуль прослушивал синхросообщения с меткой времени мастера, что позволило синхронизировать счетчик со временем мастера.

Кроме того, синхронизация времени компьютера и мастера, которая ранее работала только в сторону мастера, была доработана до двунаправленной. Это позволило на компьютере конвертировать метку времени со счетчика в систему отсчета компьютера.

Поскольку распознавание крайних светодиодов на индикаторе работало нестабильно, было решено оставить крайние светодиоды всегда включенными, а следующие от края - всегда выключенными. Под сам счетчик оставалось  $16 - 2 \cdot 2 = 12$  разрядов. Это позволило стабильно отличать на изображении опорные светодиоды от всех остальных.

Так как период минимально различимого разряда составлял 32 мс, то 12 разрядов переполнялись раз в  $32 \cdot 2^{12} = 131000ms = 131sec \approx 2.18min$ , что более чем достаточно для того, чтобы отследить задержку камеры, которая не превышает 0.3 секунды.

### 5.3.3 Тестирование

Для того, чтобы проверить не только временную, но и пространственную синхронизацию камеры, нужно провести следующий эксперимент: на светлый пол при помощи изоленты наносится черный паттерн. Данный паттерн легко считывается камерой и преобразуется в команды на включение светодиодов.

Если вся система работает синхронно, то в результате должно получиться, что светодиоды загораются только тогда, когда находятся ровно над черными областями. Если будет иметь место временная задержка, то паттерн включения светодиодов будет соответствовать паттерну на полу, но будет смещен относительно него.

Для тестирования был изготовлено поле с черно-белым паттерном.

Путем настройки параметров системы удалось добиться точного совпадения между паттерном на поле и паттерном включения светодиодов на скорости порядка 10 см/с. Теоретически, прототип будет стабильно работать и при большей скорости движения, однако разогнать его до больших скоростей руками не удалось.

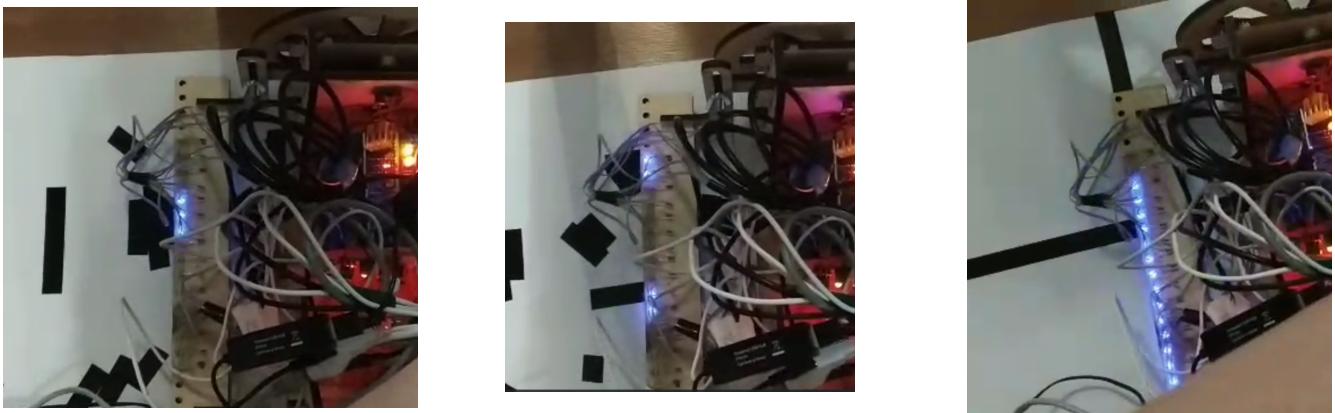


Рис. 19: Робот проходит по полю

#### 5.4 Этап четвертый. Добавление нейронной сети.

Четвертый этап для реализации требовал цветное покрытие, имитирующее поверхность настоящего поля. К сожалению, в связи с коронавирусом, а также отсутствием финансирования проекта, покрытие не было изготовлено, и протестировать прототип на нем не удалось.

Тем не менее, нейросеть для распознавания растений была разработана и протестирована на тестовой выборке изображений.

Для нейросети была выбрана архитектура U-Net, причем использовался только энкодер без декодера. Структура нейросети представлена на Рис. 20.

Model: "model_1"		
Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[None, None, None, 3]	0
conv2d (Conv2D)	(None, None, None, 16)	448
max_pooling2d (MaxPooling2D)	(None, None, None, 16)	0
conv2d_1 (Conv2D)	(None, None, None, 32)	4640
max_pooling2d_1 (MaxPooling2D)	(None, None, None, 32)	0
conv2d_2 (Conv2D)	(None, None, None, 64)	18496
max_pooling2d_2 (MaxPooling2D)	(None, None, None, 64)	0
conv2d_3 (Conv2D)	(None, None, None, 128)	73856
max_pooling2d_3 (MaxPooling2D)	(None, None, None, 128)	0
conv2d_4 (Conv2D)	(None, None, None, 256)	295168
conv2d_6 (Conv2D)	(None, None, None, 128)	295040
conv2d_7 (Conv2D)	(None, None, None, 3)	3459
<hr/>		
Total params: 691,107		
Trainable params: 691,107		
Non-trainable params: 0		

Рис. 20: Структура нейросети

Нейросеть обучалась на 10 изображениях разрешением 480x256 пикселей, которые были аугментированы случайными сдвигами, поворотами и искажениями.

Несмотря на столь скромный датасет, попиксельная точность предсказаний нейросети равна 96%.

Разумеется, этого недостаточно, ведь валидационная выборка довольно похожа на тренировочную выборку. Поскольку датасет очень мал, данная нейросеть обладает очень низкой способностью к генерализации.

Однако, результаты обучения показывают, что архитектура нейросети в целом выбрана верно.

Для применения данной нейросети на практике, данную архитектуру нужно обучить значительно большем датасете, включающем в себя не менее 500 изображений, а желательно около 5000.

Результаты работы нейросети можно видеть на Рис. 21. Слева представлено исходное изображение, справа - ground truth маска, а по центру - выход нейросети. Синим цветом обозначены культурные растения, зеленым - сорные.

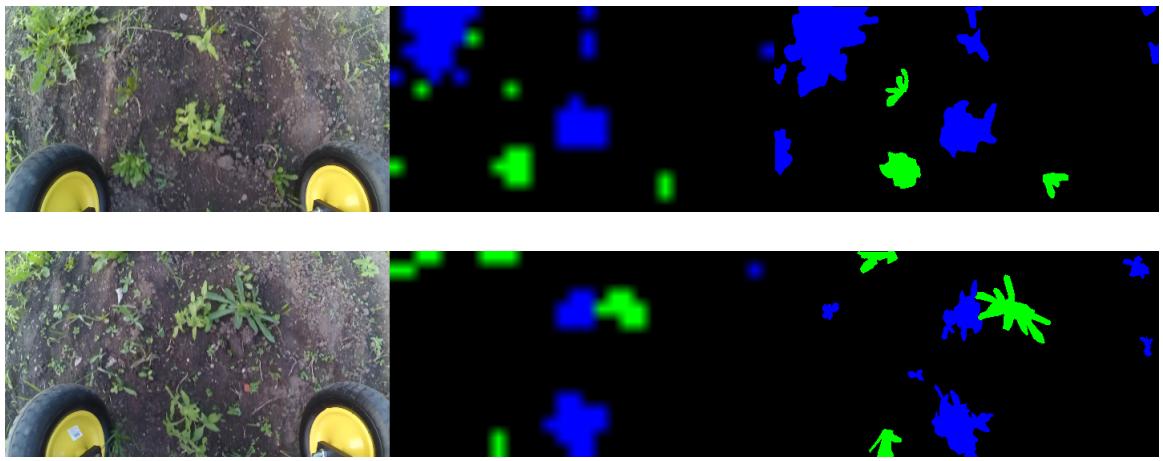


Рис. 21: Результаты распознавания растений

## 5.5 Выводы по главе 5

В данной главе был описан процесс реализации ранее разработанных алгоритмов на практике, а также реализация теории в физическом прототипе. Результатом данной работы явилось изготовление прототипа, способного имитировать опрыскивание растений путем подсвечивания черных маркеров на белом поле.

Достигнутый результат свидетельствует о жизнеспособности разработанного алгоритма, однако ни в коем случае не является окончательным. Исходя из данного результата можно очертить направление дальнейшей работы.

Следующим этапом исследований будет проведение эксперимента с распознаванием изображений растений по камере с физическим прототипом.

После этого нужно изготовить новые версии электронных компонентов, учтя ошибки при разработке текущих версий. Также нужно изготовить их в большем количестве, чтобы создать прототип как минимум из десятка плат.

В следующей итерации прототипа нужно увеличить скорость движения, а также поискать способы уменьшения задержки камеры, например, использовать аналоговую камеру вместо цифровой.

# ЗАКЛЮЧЕНИЕ

В данной научно исследовательской работе были рассмотрены два функциональных узла системы выборочного опрыскивания полей химикатами. Были описаны требования к данным узлам, функциональная схема, а также подробный алгоритм их работы. На основании теоретических расчетов был изготовлен и за-программирован прототип.

В результате проведенной работы были решены следующие задачи:

- реализован алгоритм преобразования выхода нейронной сети в команды управления клапанами;
- реализован протокол передачи данных между главным компьютером и контроллерами клапанов по шине rs485.
- разработаны электронные платы управления разбрызгиванием гербицидов, а также реализовано программное обеспечение для них.
- создан физический прототип для отладки работы системы.
- реализован алгоритм синхронизации времени камеры со временем всей системы.
- реализована программа распознавания растений на основе нейронной сети;

В результате сравнения с аналогами было выяснено, что на существующие на данный момент решения не ориентированы на российский рынок. Кроме того, было выяснено, что разрабатываемое нами решение не уступает по характеристикам зарубежным аналогам. Таким образом, описанное решение по выборочному опрыскиванию полей позволит значительно повысить эффективность сельского хозяйства в России, а также сократить содержание вредных веществ в продуктах.

Результатом проделанной работы явился минимально работоспособный прототип, позволяющий положительно судить о жизнеспособности проекта. Проект в целом нельзя считать завершенным, дальнейшая разработка предполагает как минимум еще одну итерацию прототипирования перед созданием промышленного образца.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. U-Net: Convolutional Networks for Biomedical Image Segmentation  
[Электронный ресурс]. - Режим доступа:  
<https://arxiv.org/abs/1505.04597>  
(дата обращения: 05.12.2019)
2. Mask R-CNN  
[Электронный ресурс]. - Режим доступа:  
<https://arxiv.org/abs/1703.06870>  
(дата обращения: 05.12.2019)
3. You Only Look Once: Unified, Real-Time Object Detection  
[Электронный ресурс]. - Режим доступа:  
<https://arxiv.org/abs/1506.02640>  
(дата обращения: 05.12.2019)
4. Formalizing FreeRTOS: First Steps  
[Электронный ресурс]. - Режим доступа:  
[https://www.researchgate.net/publication/221136366\\_Formalizing\\_FreeRTOS\\_First\\_Steps](https://www.researchgate.net/publication/221136366_Formalizing_FreeRTOS_First_Steps)  
(дата обращения: 05.12.2019)
5. Design and Implementation of Modbus Slave Based on ARM Platform and FreeRTOS Environment  
[Электронный ресурс]. - Режим доступа:  
[https://www.researchgate.net/publication/284724085\\_Design\\_and\\_Implementation\\_of\\_Modbus\\_Slave\\_Based\\_on\\_ARM\\_Platform\\_and\\_FreeRTOS\\_Environment](https://www.researchgate.net/publication/284724085_Design_and_Implementation_of_Modbus_Slave_Based_on_ARM_Platform_and_FreeRTOS_Environment)  
(дата обращения: 05.12.2019)
6. Blue River "See & Spray" web site  
[Электронный ресурс]. - Режим доступа:  
<http://www.bluerivertechnology.com/>  
(дата обращения: 05.12.2019)

7. Bosch "Smart spraying" web site  
[Электронный ресурс]. - Режим доступа:  
[https://www.bosch.com/research/know-how/success-stories/  
smart-spraying-precision-herbicide-application-on-weeds/](https://www.bosch.com/research/know-how/success-stories/smart-spraying-precision-herbicide-application-on-weeds/)  
(дата обращения: 05.12.2019)