

Государственное бюджетное общеобразовательное учреждение города
Москвы

«Школа №2103»

Документация по командному кейсу №1.
“Автоматизированный генератор псевдослучайных чисел”
МОСКОВСКОЙ ПРЕДПРОФЕССИОНАЛЬНОЙ

ОЛИМПИАДЫ ШКОЛЬНИКОВ

“ГРУТ”

Работу выполнила
команда “Грут”



Ученики 11 «Т» класса ГБОУ
Школы №2103

Усачёв Даниил Сергеевич
Бриллиантов Михаил Константинович
Наседкин Тимофей Филиппович
Сагдуллин Юрий Романович
Владимир Владимирович Шафранский

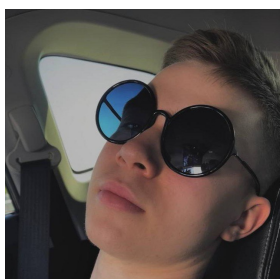
Научный руководитель
инженер
Ткаченко Артём Алексеевич

Москва, 2023 г.

Оглавление

Актуальность и проблематика	4
Цель 	5
Задачи 	6
Общее описание	7
Общая структурная схема	8
Функционал	9
Диаграмма Use Case	10
Диаграмма StateMachine	11
Диаграмма Sequence	13
Фото узлов	14
Кинематика	15
3Д модель	16
Текстовое описание	17
Таблица электрокомпонентов	18
Монтажная схема	20
Принципиальная схема	21
Таблица подключений	22
Raspberry Pi 3B	22
Arduino Nano	22
Текстовое описание и оформление	24
Блок-схемы	25
Код	25
Ввод переменных	26
Список использованных источников	30

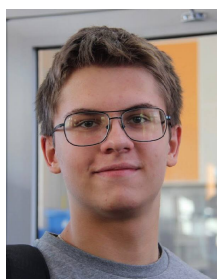
Наша команда



Бриллиантов Михаил
‘Программист’



Усачев Даниил
‘Программист’ и ‘Капитан’



Наседкин Тимофей
‘Документалист’



Сагдуллин Юрий
‘Конструктор’



Шафранский Владимир
‘Электронщик’

Наш логотип.



Введение

Основной задачей нашего проекта является разработка автоматизированного генератора псевдослучайных чисел. Источники настоящих случайных чисел найти крайне трудно. Физические шумы, такие, как детекторы событий ионизирующей радиации, дробовой шум в резисторе или космическое излучение, могут быть такими источниками.

В то же время случайные числа, получаемые из физического источника, могут использоваться в качестве порождающего элемента для программных генераторов псевдослучайных чисел. Такие генераторы применяются в игровых автоматах, лотереях, а также в настольных играх.

Мы создали устройство, которое способно определить и запомнить полученные с игровых кубиков числа. Мы считаем, что наш проект будет полезен как в промышленных сферах(в игровых автоматах, лотереях), так и в повседневной жизни.

Нас заинтересовал этот кейс использованием системы компьютерного зрения. Мы считаем, что при помощи этой системы упростится считывание данных с игровых кубиков в любой настольной игре. Наше устройство облегчит совершение бросаний игровых кубиков и также обеспечит честность выпадения очков и разнообразит любую настольную игру.

Актуальность и проблематика

Источники настоящих случайных чисел найти крайне трудно. Физические шумы, такие, как детекторы событий ионизирующей радиации, дробовой шум в резисторе или космическое излучение, могут быть такими источниками. В то же время случайные числа, получаемые из физического источника, могут использоваться в качестве порождающего элемента для программных генераторов псевдослучайных чисел. Такие комбинированные генераторы применяются в криптографии, лотереях, игровых автоматах и даже в настольных играх.

Для устранения фальсификации процесс перемешивания должен быть независимым и автоматизированным. Решения данной проблемы - компьютерное зрение. Сегодня данной технология получила широкое распространение в различных сферах жизни, таких как: распознавание по биометрии, перевод текста, разработка автопилота.

Наша команда озадачилась проблемой генератора псевдослучайных чисел, и нами была разработана система для упрощения и внедрения справедливости в повседневную жизнь.

Цель

Создание независимого от внешних факторов и основанного на работе механизма генератора случайных чисел. Который будет получать комбинацию чисел, выпавших на 1-3 игральных кубиках. После выпадания способен распознать числа при помощи системы технического зрения и вывести полученный результат на сервер.

Изучить новые и отточить имеющиеся навыки в области программирования, конструирования, моделирования, анализа, структуризации.

В нашем устройстве должно быть предусмотрено:

- 1) Автоматизированный лифт для подъёма игральных кубиков в барабан.
- 2) Двухступенчатое перемешивание игральных кубиков(первое при помощи барабана, а второе в башне свободного падения).
- 3) При помощи системы компьютерного зрения происходит считывание и распознавание результатов, полученных на игральных кубиках.
- 4) Сохранение результатов в облако и выводение их на дисплей.
- 5) Система возврата кубиков для дальнейшего использования из системы распознавания в систему перемешивания без непосредственного вмешательства участников.

Задачи

1. Анализ кейсового задания, формирование требований ограничений к разрабатываемому устройству.
2. Анализ предметной области и инструментов для решения задачи.
3. Проектирование устройства:
 - а) эскиз устройства
 - б) проектирование кинематической системы
 - с) UML-диаграммы
4. Проектирование 3D-модели устройства:
 - а) его составных частей
 - б) корпуса
5. Проектирование электротехнической системы устройства.
6. Проектирование алгоритмов работы программного обеспечения.
7. Разработка систем устройства:
 - а) кинематической системы
 - б) электротехнической системы
8. Разработка программного обеспечения.
9. Изготовление и сборка устройства.
10. Прототипирование устройства.
11. Тестирование и отладка устройства.
12. Подготовка документации.

Описание проекта

Общее описание

Наш проект представляет из себя устройство для генерации псевдослучайных чисел. Управление происходит через сенсорный дисплей, встроенный в механизм. У пользователя есть две возможные опции для работы механизма: многократный и однократный повтор цикла.

Работа устройства:

- 1) Запуск системы перемешивания барабана с игральными кубиками. Сначала барабан крутится в одну сторону, а потом крутится в другую. После перемешивания игральные кубики выбрасываются в башню свободного падения.
- 2) Затем игральные кубики попадают на платформу лифта, и при помощи системы компьютерного зрения распознаются полученные результаты.
- 3) Далее полученные результаты сохраняются в базу данных и передаются на дисплей пользователя.
- 4) Запуск системы возврата игровых кубиков в исходную точку.
- 5) Лифт с игральными кубиками поднимается до исходного уровня, наклоняется, сбрасывая игральные кубики в барабан перемешивания для повторного цикла.

Количество повторений системы выбирается пользователем. Предусмотрена возможность экстренного завершения работы системы в случае необходимости.

Элементы системы:

- 1) Подсистема перемешивания игровых кубиков.
- 2) Подсистема компьютерного зрения.
- 3) Подсистема возврата игровых кубиков в барабан перемешивания.
- 4) Башня свободного падения и барабан перемешивания.
- 5) База данных.
- 6) Сенсорный дисплей для управления устройством.

Общая структурная схема

Функционал

Физические кнопки:

1. Аварийная остановка

Дисплей:

1. Система отладки
 - 1.1. Запуск подсистемы возврата игральных костей из области распознавания в подсистему перемешивания
 - 1.2. Запуск подсистемы оптического распознавания численного результата, выпавшего на игральном косте
 - 1.3. Запуск подсистемы перемешивания игральных костей
 - 1.4. Запуск подсистемы возврата игральных костей из области падения в подсистему распознавания
2. Конечное выполнение
 - 2.1. Выбор количества повторений
 - 2.2. Запуск программы
 - 2.3. Завершение программы по завершении цикла
3. Бесконечное выполнение
 - 3.1. Запуск программы
 - 3.2. Завершение программы по завершении цикла
4. История бросков
 - 4.1. Результаты
 - 4.2. Мода
 - 4.3. Среднее арифметическое
 - 4.4. Счетчик выпавших чисел

Система:

1. Связь с удаленным сервером
2. Передача информации

Диаграмма Use Case

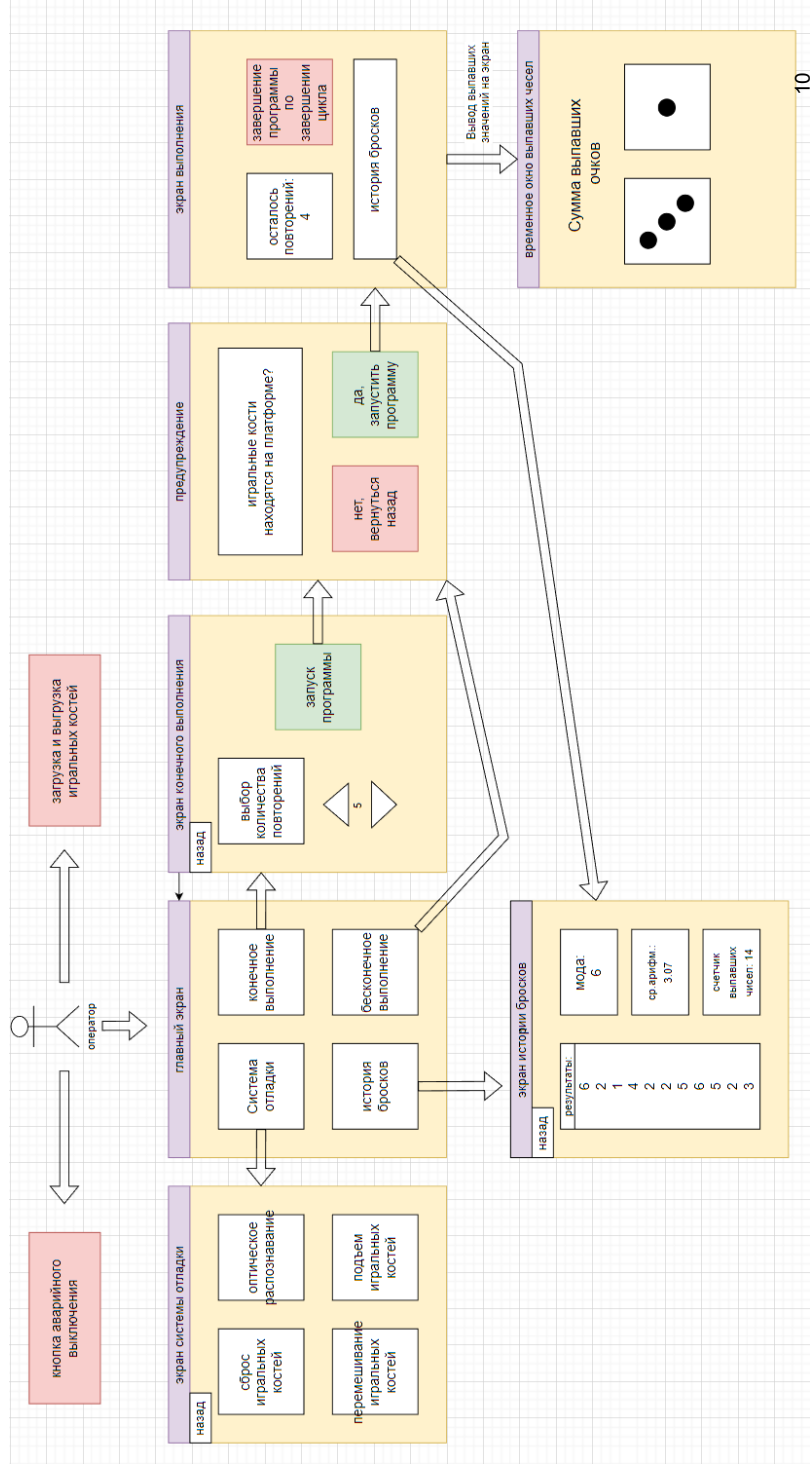


Диаграмма StateMachine

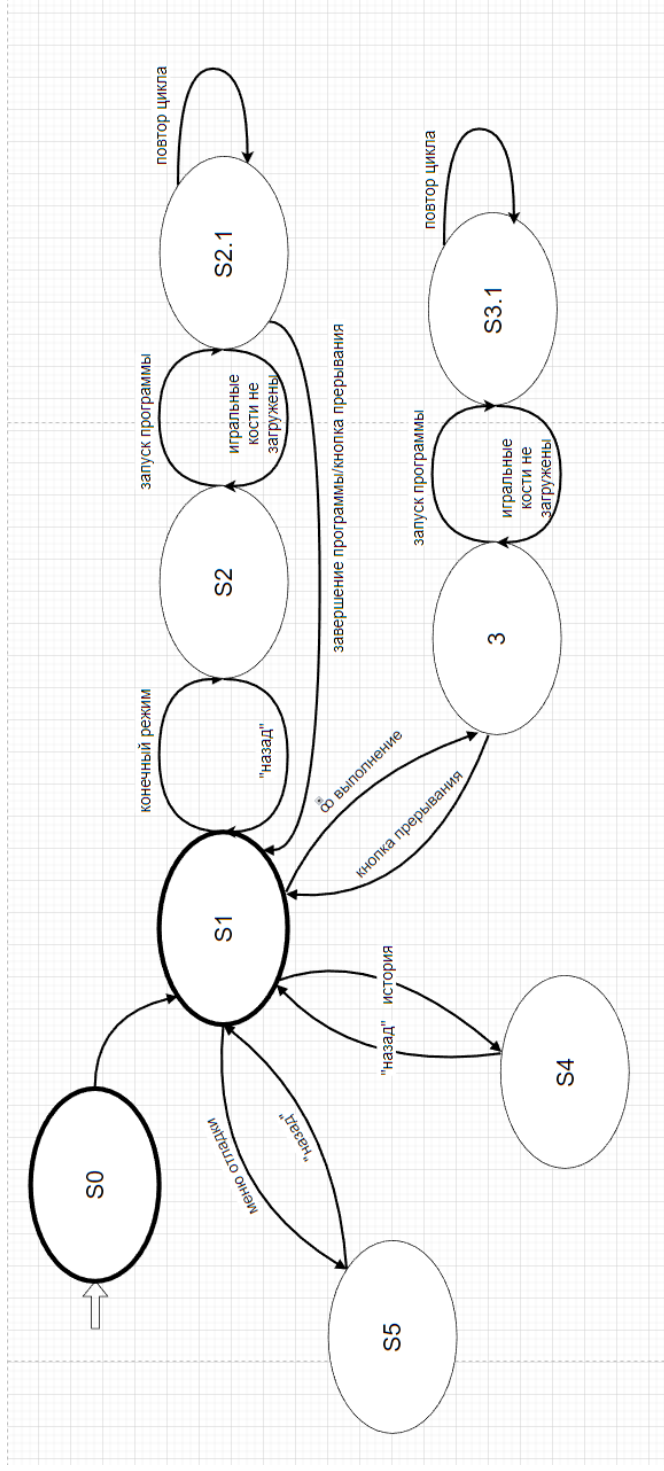
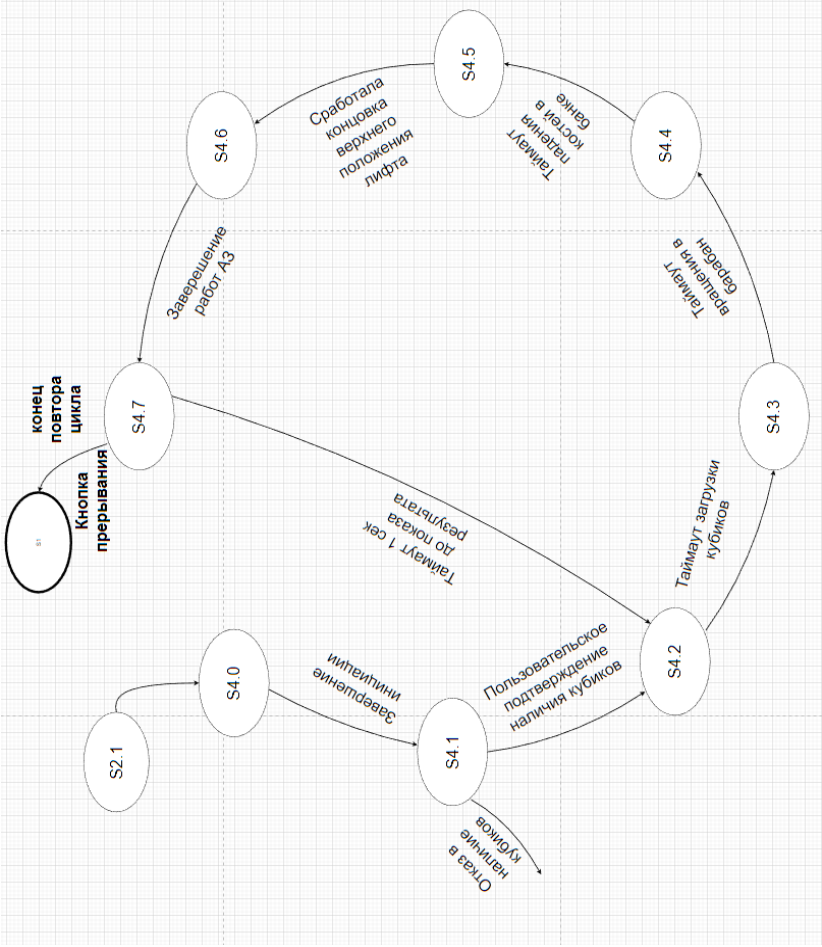
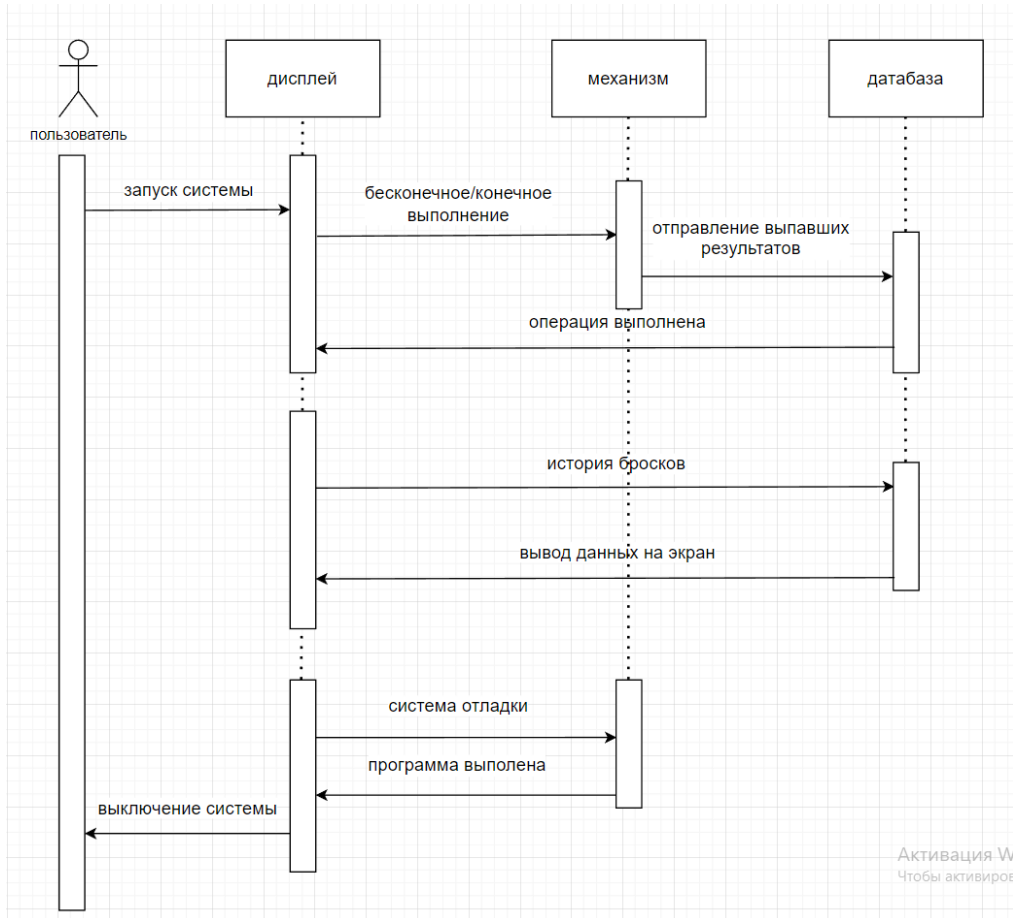


рис. 1



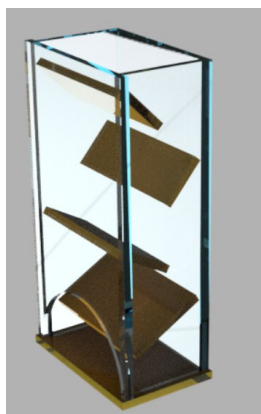
S0	перемещение всех элементов в начальные положения
S1	ожидание команды
S2	предупреждение: игральные кости загружены
S2.1	работа программы(см. рис 1)
S3	предупреждение: игральные кости загружены
S3.1	работа программы(см. рис 1)
S4	таблица результатов выпадения
S5	работа отдельных элементов

Диаграмма Sequence

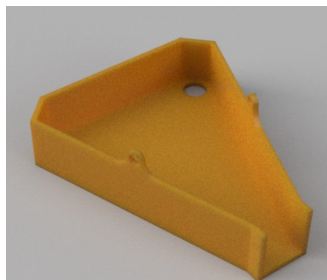


Устройство и кинематика

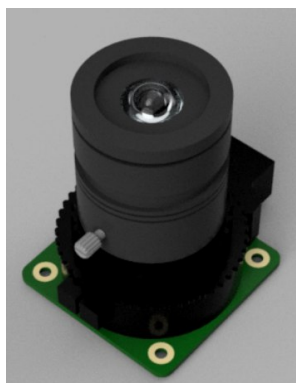
Фото узлов



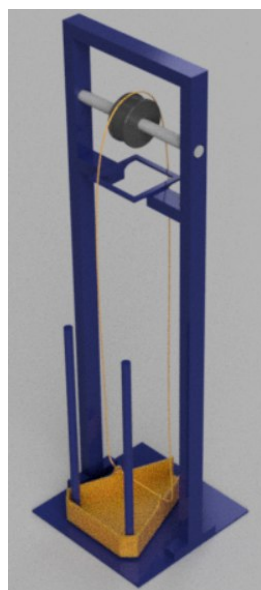
башня падения



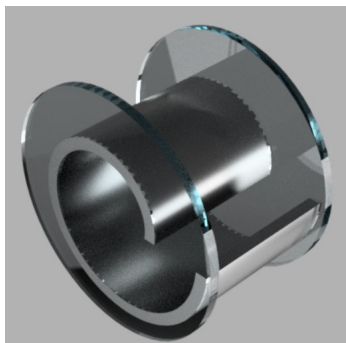
Платформа



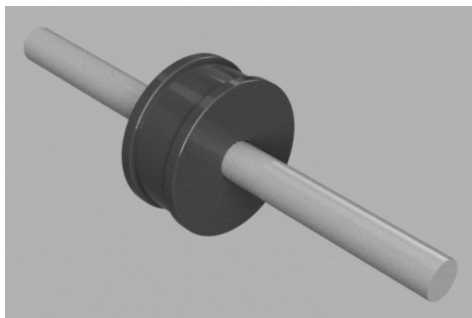
камера



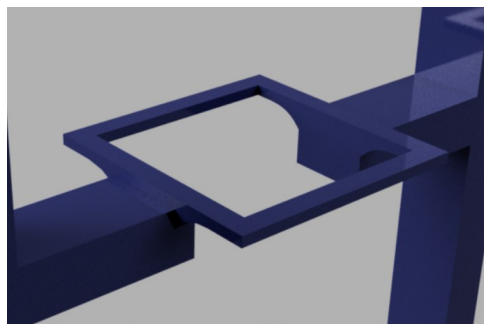
Платформа



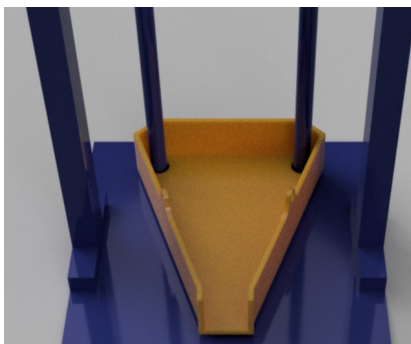
барабан вращения



блок



крепление для камеры



направляющие

Кинематика

3Д модель

Электроника

Текстовое описание

Нами были выбраны двигатели:

1. Электродвигатель DC Motor 555: Момент силы достаточно для вращения системы перемешивания
2. Сервопривод SG90: участвует в цикле возврата игральных костей в систему распознавания.

Энергопотребление

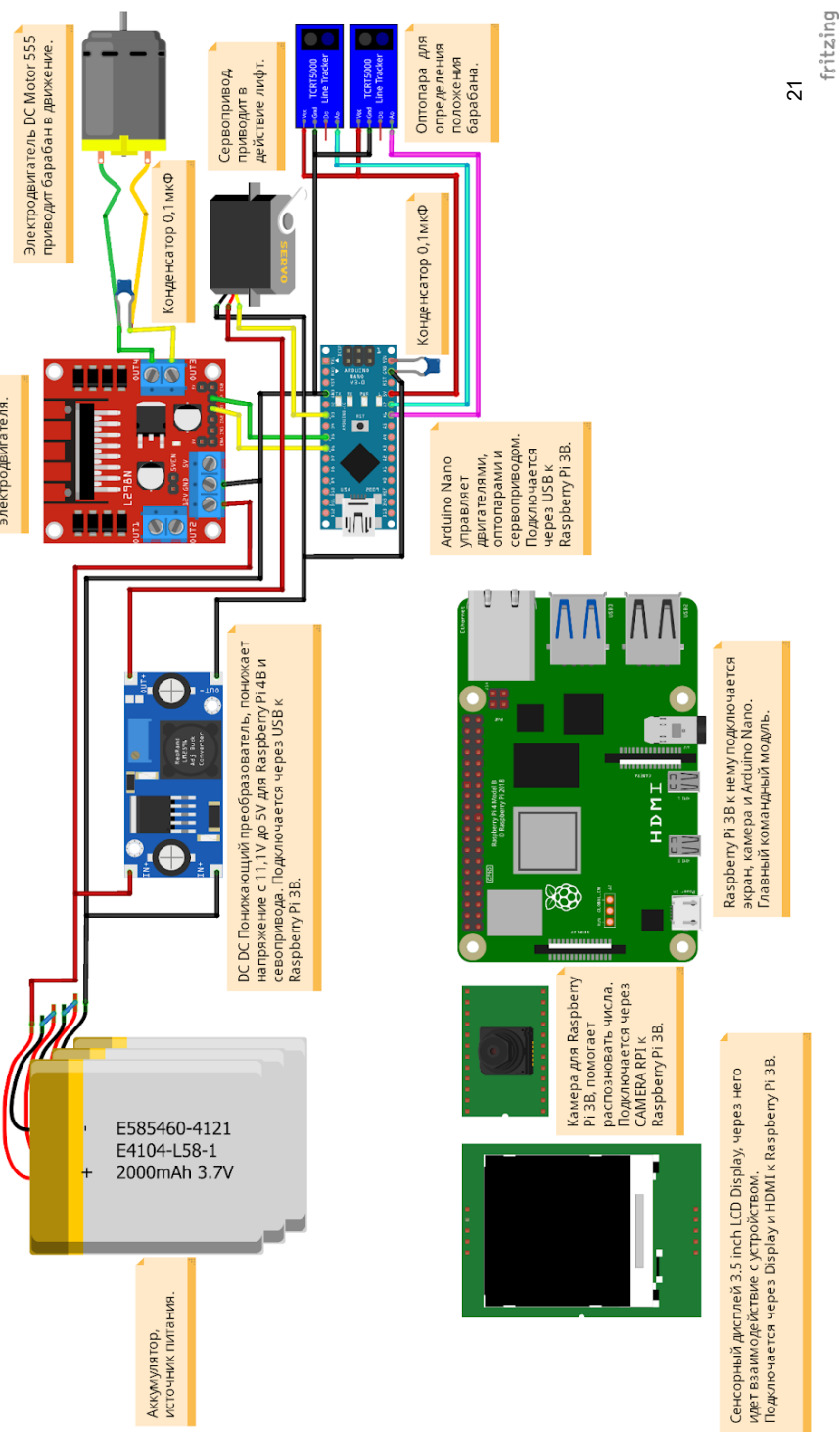
1. Затраты на питание системы - 97 Вт.
2. Для питания системы был выбран оптимальный источник с достаточным запасом мощности Li-Po3S 184 Вт.
3. В системе предусмотрены конденсаторы для предотвращения от проблем с питанием.
4. Присутствует запас мощности в 87 Вт.

Таблица электрокомпонентов

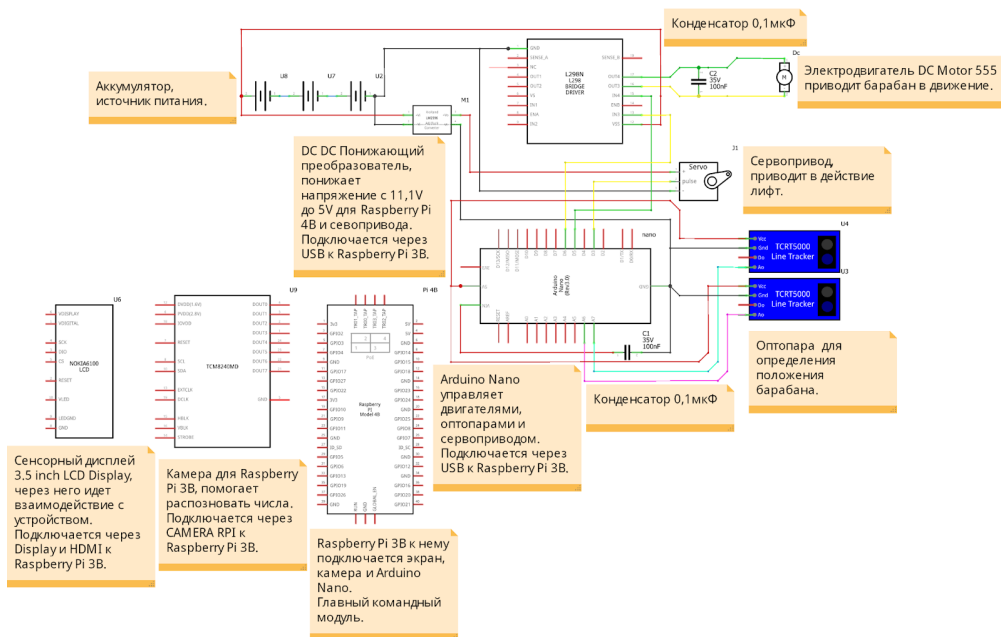
Имя	Устройства	Модель	Параметры	Комментарий
папо	Отладочная плата на базе ATmega328	Arduino Nano	7-12V 16 Mhz 40 mAh ATmega328	Управляет двигателями, оптопарами и сервоприводом.
Pi 4B	Одноплатный компьютер	Raspberry Pi 3B	5V, 1.5A max - 2.5A	Управляет системой распознавания, экраном. Реализует алгоритмы машинного зрения
U3	Оптопара	TCRT5000	3-12mm 3,3-5V 10mA	Датчик для определения положения барабана.
U4				
U95	Камера	Raspberry Pi camera	5 MP OmniVision OV5647 Color CMOS QXGA	Камера для распознавания чисел.
U6	Сенсорный дисплей	3.5 inch LCD Display	480x320 IPS HDMI	Сенсорный дисплей для управления функциями аппарата.
Dc	Электродвигатель	DC Motor 555	12V 140 mA 2200-3000rpm	Двигатель для барабана.
L298N	Драйвер	L298N	5-35V 0-2A 25W	Драйвер для электродвигателя DC Motor 555.
J1	Сервопривод	SG90	1,5 кг*см 180° 4,8-6V 650mA	Сервопривод для лифта.

M1	DC DC Понижающий преобразователь	LM2596S	In:2,4-40V Out:1,2-37V 3A(4,5A max) USB port	Понижает напряжение с 11,1V до 5V для Raspberry Pi 4B
C1	Керамические конденсаторы		35V 0,1мкФ	Защита от перепадов тока
C2				
U2	Аккумулятор	Li-Po3S(11,1V)	3S 11.1V 30s 5200mAh	Источник питания.

Монтажная схема



Принципиальная схема



fritzing

Таблица подключений

Raspberry Pi 3B

Пин	Назначение пина	Устройство	Пин устройства	Комментарий
Шина CSI	Camera Serial Interface	Raspberry Pi Cam Module	Шина CSI	Камера, разработана для Raspberry Pi. Подключается шлейфом CSI и обеспечивает большой FPS.
DISPLAY	Display Interface	3.5 inch LCD Display	DISPLAY	Подключает сенсорный экран

Arduino Nano

Пин	Назначение пина	Устройство	Пин устройства	Комментарий
GND	Заземление	Керамические конденсатор Оптопара L298N LM2596S SG90	GND OUT- -	Общее заземление компонентов.
5V	Питание 5V	Оптопара	VCC	Питание для оптопары.
A6	Прием сигнала	Оптопара	A0	Считывание сигналов от оптопары.
A7				
D3	Передача сигнала	SG90	pulse	Программируемый порт для

				сервопривода.
D5	Передача сигнала	L298N	IN4	Подключается к драйверу.
D6	Передача сигнала	L298N	IN3	Подключается к драйверу.
VIN	Питание	Керамические конденсатор	+	Для безопасности.
USB	Обмен информацией с Raspberry Pi 3B	Raspberry Pi 3B	USB	Связь между Arduino Nano и Raspberry Pi 3B.

Алгоритм

Текстовое описание и оформление

Принцип работы алгоритма:

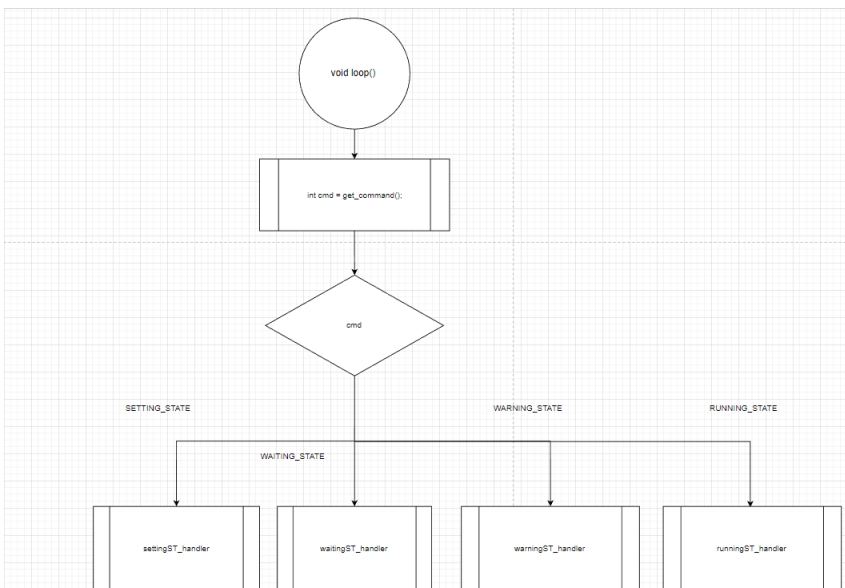
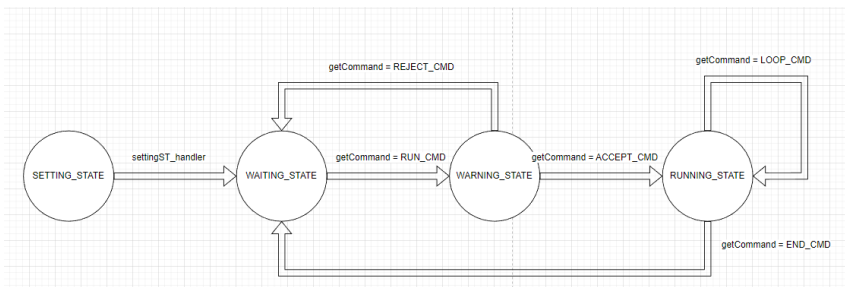
Пользователь загружает игральные кости на платформу

При помощи дисплея выбирается программа для выполнения, это может быть как запуск отдельных узлов, так и прохождения полного цикла.

Физическая кнопка экстренного отключения полностью останавливает работу системы. Кнопка завершения цикла на дисплее завершает работу программы после прохождения текущего цикла.

При старте программы происходит сброс игральных костей в систему перемешивания. После сброса игральных костей из системы перемешивания, они попадают на подъемную платформу. Затем платформа поднимает игральные кости вверх, где камера распознает выпавшее значение на игральных костях, результат отправляется на удаленный сервер, а также выводится на дисплей. Данный цикл выполняется до нажатия кнопки остановки, либо выполнения заданного количества итераций.

Блок-схемы



Код

Ввод переменных

```
1 //Константы-состояния:
2 #define SETTING_STATE 0
3 #define WAITING_STATE 1
4 #define WARNING_STATE 2
5 #define RUNNING_STATE 3
6 #define SENSOR_WAITING 4
7 #define SENSOR_READY 5
8
9 //Константы команды:
10 #define RUN_CMD 0x0
11 #define ACCEPT_CMD 0x1
12 #define REJECT_CMD 0x2
13 #define LOOP_CMD 0x3
14 #define END_CMD 0x4
15 #define ERROR_CMD 0xFE
16 #define NO_CMD 0xFF
17
18 int state = _STATE;
19
20 void setup(){
21 }
22
```

Работа программы

```
81 void loop(){
82   int cmd = getCommand();
83   switch(state){
84     case SETTING_STATE:{
85       settingSt_handler(sensor_cmd);
86       break;
87     }
88     case WAITING_STATE:{
89       waitingSt_handler(cmd);
90       break;
91     }
92     case WARNING_STATE:{
93       errorSt_handler(cmd);
94       break;
95     }
96     case RUNNING_STATE:{
97       runningSt_handler(cmd);
98       break;
99     }
100   }
101 }
```

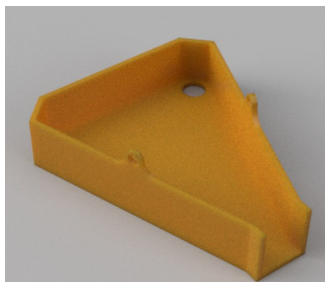
Handler функции

```
24 int getCommand(){
25     int cmd = NO_CMD;
26     if (Serial.available()){
27         cmd = Serial.read();
28         if (cmd > ERROR_CMD)
29             cmd = NO_CMD;
30     }
31     return cmd;
32 }
33
34 void settingST_handler(int sensor_cmd){
35     if (sensor_cmd == SENSOR_WAITING){
36         state = SETTING_STATE;
37     }
38     else{
39         if (checkObject()){
40             state = WAITING_STATE;
41         }
42     }
43 }
44
45 void waitingSt_handler(int cmd){
46     if (cmd == RUN_CMD){
47         state = WARNING_STATE;
48     }
49     else{
50         if (checkObject()){
51             state = WAITING_STATE;
52         }
53     }
54 }
55
56 //elif можно юзать??
57 void warningSt_handler(int cmd){
58     if (cmd == ACCEPT_CMD){
59         state = RUNNING_STATE;
60     }
61     elif (cmd == REJECT_CMD){
62         state = WAITING_STATE;
63     }
64     else{
65         if (checkObject()){
66             state = WAITING_STATE;
67         }
68     }
69
69 void runningSt_handler(int cmd){
70     if (cmd == LOOP_CMD){
71         state = RUNNING_STATE;
72     }
73     else{
74         if (checkObject()){
75             state = WAITING_STATE;
76         }
77     }
78 }
79
80 }
```

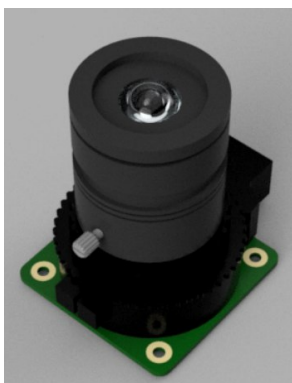
МЕДІАМАТЕРІАЛ



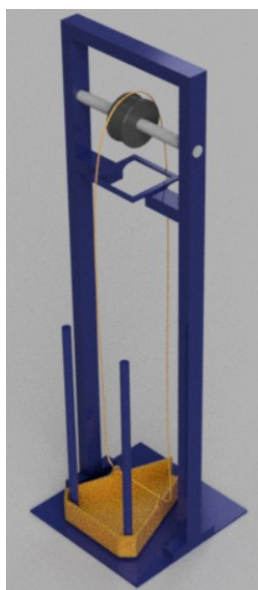
башня падзення



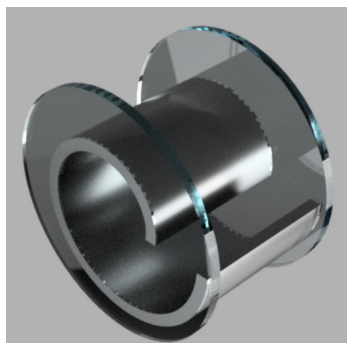
Платформа



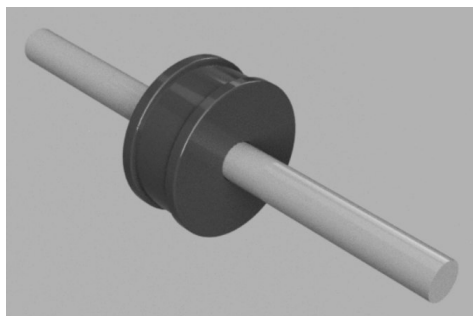
камера



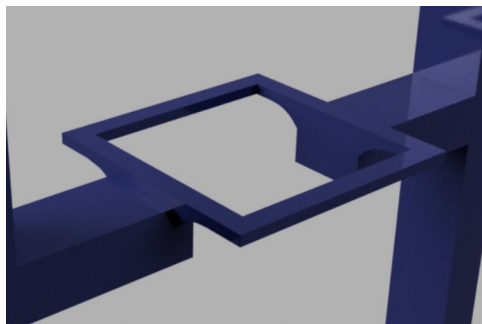
Лифт



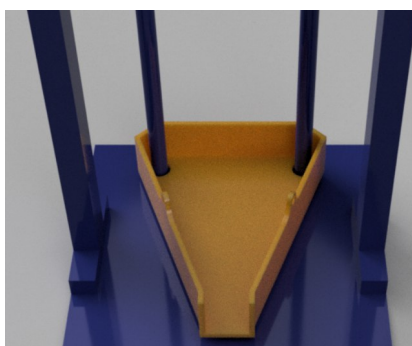
барабан вращения



блок



крепление для камеры



направляющие

ЗАКЛЮЧЕНИЕ

Мы научились пользоваться системой компьютерного зрения, работать в Fusion 360, работать с Arduino, программировать на C++.

Нами были выполнены следующие задачи:

- 1) Разработана конструкция всего устройства и отдельных его узлов.
- 2) Создана программа, позволяющая работать устройству.
- 3) Написана документация.

Задачи, которые мы собираемся выполнить:

- 1) Создание прототипа.
- 2) Отладка устройства и проведение испытаний.

Перспективы развития проекта:

- 1) Улучшение внешнего вида, дизайна.
- 2) Разработка приложения.
- 3) Массовое производство.

Список использованных источников

1. Петин В. А., Практическая энциклопедия Arduino. / Биняковский А.А. – Москва: ДМК издательство, 2017 – 151 стр.
2. Петин В.А., Проекты с использованием контроллера Arduino. – 2-е изд., перераб. и доп. – Спб.: БХВ –Петербург, 2017. – 466 стр.
3. Джереми Блум, Изучаем Arduino. Перевод Петина В.А. - БХВ –Петербург, 2015.
4. Монк Саймон, Программируем Arduino. Профессиональная работа со скетчами. – ИД «Питер», 2017 .
5. База знаний Амперки [Электронный ресурс] URL: <http://wiki.amperka.ru/>
6. Форум Вики iArduino [Электронный ресурс] URL: <https://wiki.iarduino.ru/>
7. Официальный сайт канала AlexGyver [Электронный ресурс] URL: <https://alexgyver.ru/>

Вывод

Данный проект расширил наши представления о системе компьютерного зрения, о случайных числах. Благодаря этому проекту мы развили свои навыки в программировании, в 3D моделировании, в работе с электроникой и в сборке компонентов.

Благодаря нашему упорному труду и командной работе мы справились с выполнением поставленных задач.

И в конечном итоге был создан генератор псевдослучайных чисел, который соответствует изначальной задумке и техническому заданию.