

МОСКОВСКАЯ ОБЛАСТЬ  
ПУШКИНСКИЙ МУНИЦИПАЛЬНЫЙ РАЙОН

**Инженерная книга  
команды «Позит»**

**IKR-kl-10**

**проект «Фабрика клонов»**

Инженерная книга команды «ПОЗИТ», Пушкинский район Московская область  
2019 год

ВИЗИТКА КОМАНДЫ .....	5
Команда разработчиков робота-транспортировщика и автоматизированного сборщика клонов .	5
ВВЕДЕНИЕ.....	6
ПОСТАНОВКА ЗАДАЧИ ИСЛЕДОВАНИЯ:.....	7
<b>Актуальность проекта.....</b>	<b>7</b>
<b>Цель проекта.....</b>	<b>7</b>
<b>Задачи. ....</b>	<b>7</b>
ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ ПРОЕКТА .....	8
Изучение существующих процессов создания моделей по прототипу. ....	8
Поиск способа модернизации технологического процесса сканирования и создания клона. ....	9
ОБЩЕЕ ОПИСАНИЕ ПРОЕКТА «ФАБРИКА КЛОНОВ».....	10
Передача данных реализована по протоколу Serial.....	11
Платформа сканирования и контроля ” SD-01“ .....	11
Робот-Манипулятор .....	13
Проектирование, сборка и отладка узлов сборочной платформы.....	15
ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ ПРОЕКТА. ЭТАП 2.....	17
ПРОГРАММЫ.....	<b>Ошибка! Закладка не определена.</b>
1.Подсчет черных столбов .....	<b>Ошибка! Закладка не определена.</b>
2.Алгоритм доставки кубика-посылки на склад .....	<b>Ошибка! Закладка не определена.</b>
3.Алгоритм движения в лабиринте. ....	<b>Ошибка! Закладка не определена.</b>
ПРАВИЛА ТЕХНИКИ БЕЗОПАСНОСТИ ПРИ РАБОТЕ ЗА КОМПЬЮТЕРОМ. ....	18
ЖУРНАЛ РАБОТЫ НАД ПРОЕКТОМ.....	19
ПРИЛОЖЕНИЯ.....	22
Приложение 4. Договор о сотрудничестве с ЗАО «Зеленоградское» .....	<b>Ошибка!</b>
<b>Закладка не определена.</b>	

Приложение 5. Договор о сотрудничестве с УФПС МО ф-л ФГУП «Почта России»

..... **Ошибка! Закладка не определена.**

Приложение 6. Договор о сотрудничестве с ООО «ПОЗИТ». **Ошибка! Закладка не определена.**

Приложение 7. Договор о сотрудничестве с «РОСТАГРОКОМПЛЕКС».. **Ошибка! Закладка не определена.**

СПИСОК ЛИТЕРАТУРЫ: ..... 45

Инженерная книга команды «ПОЗИТ», Пушкинский район Московская область  
ВИЗИТКА КОМАНДЫ



Команда разработчиков робота-транспортировщика и автоматизированного сборщика клонов

Проект команды «Позит» представляет собой макет роботизированной линии сборки клонов, выполненный с минимальным использованием механических конструкторов и ориентированный на использование микроконтроллеров широкого применения производителей Microchip и Atmel.

Инженерная книга выполнена в соответствии с требованиями робототехнических соревнований ИКаР-2019 («Инженерные кадры России»), направленных на популяризацию научно-технического творчества и повышение инженерных компетенций обучающихся.

Роботизированные сборочные линии является основой современного автомобилестроения (сварка кузова, монтаж агрегатов, покраска) и электронной промышленности (производство микросхем). В наше время особую актуальность приобретает создание точных копий различных объектов. Достаточно вспомнить быстрое распространение 3D-печати как в промышленном производстве, так и в быту. При этом важно сосредоточить в одном производственном цикле измерение характеристик образца (3D-сканирование), получение его рабочих чертежей и изготовление копии. Важно, чтобы это производство функционировало при минимальном участии человека, чтобы исключить возможные ошибки. Решать такие задачи, несомненно, должны роботизированные системы.

По нашему мнению, очень важным является унификация используемых технических решений, что позволяет упростить создание, настройку и обслуживание таких систем.

Главной целью нашего проекта являлась разработка макета, позволяющего продемонстрировать эффективность робототехнических систем в полном цикле создания клона входного объекта (кубика Лего) – от сканирования прототипа, до транспортировки на склад и проверки качества готового продукта.

### **Актуальность проекта.**

Роботизированные системы клонирования предметов способны осуществить переворот в различных областях нашей жизни. Вместо производства в фиксированных местах и доставки необходимых товаров на далекие расстояния, становится возможным производство нужных вещей непосредственно на месте их потребления. Это исключает дорогостоящие операции транспортирования и существенно сокращает время получения необходимых товаров. При этом фабрики клонов могут строиться по единой схеме и оснащаться местным сырьем. Пересылаться могут только файлы-задания.

### **Цель проекта.**

Создание автоматизированного комплекса сканирования, сборки и доставки клонированного объекта (кубика Лего).

### **Задачи.**

- Изучение существующих технологических процессов применяемых в современном производстве;
- Поиск способов модернизации этих технологических процессов;
- Проектирование механизмов;
- Сборка и отладка разработанных механизмов;
- Приведение сборочного комплекса в соответствие задачам соревнований «ИКаР 2019»;
- Программирование блоков комплекса;
- Формирование навыков работы с различными материалами;
- Применение знаний, умений, навыков, полученных на уроках информатики и занятиях в школе робототехники;
- Использование современных технологий для получения информации;
- Современное оформление представляемых результатов проектирования;
- Умение критически оценивать результаты своего труда.

**Изучение существующих процессов создания моделей по прототипу.**

Исследовательская часть проекта состоит из двух этапов: аналитического, на котором мы собрали и проанализировали теоретический материал, сравнили мировые тенденции и тенденции развития данного процесса в нашей стране, и практического, в ходе которого мы опробовали принципиально различные варианты построения фабрики клонов с целью выбора наиболее удобного в реализации варианта.

Так как в нашем районе нет современных интеллектуальных линий сборки (на заводах используются конвейерные линии с жесткими программами производства), то мы решили, что необходимую информацию об устройстве роботизированных линий сборки мы сможем найти в сети Интернет.

Таким образом, собрав теоретический материал и исследовав на практике, возможные варианты построения интеллектуальной сборочной линии для модернизации процесса сканирования и сборки клона.



Инженерная книга команды «ПОЗИТ», Пушкинский район Московская область  
**Поиск способа модернизации технологического процесса сканирования  
и создания клона.**

Согласно технологии сканирования и создания клона мы выбрали схему расположения механизмов. Учитывая особенности технологического процесса, мы выбрали линейную схему расположения механизмов сборки клона.

Согласно технологии при создании клона оператор должен выполнить ряд операций, которые мы решили разделить на группы. Первая группа – это операции, которые способен выбрать сам заказчик, в анализе операций эти функции отмечены знаком (П) – пользователь интеллектуальной линии создания клонов, вторая группа – это операции, выполняемые роботом (Р) и третья группа – это операции, выполняемые оператором (О). Таким образом, технологический процесс приема посылки стал таким:

- выяснить, из каких материалов должен состоять клон (дерево, металл, пластик); (П)
- задать масштаб клона; (П)
- проверить прочность и правильность сборки, (О) точность доставки в ячейку пользователя;(Р)
- проверить наличие на складе нужных материалов; (Р)
- проиндексировать номер заказа;(Р)
- проверить наличие подписи пользователя в графе «неразрешенных к созданию объектов нет»; (Р)
- оценить вес готового продукта, (Р)
- оценить стоимость его создания, (Р)
- оформить ярлык. 2, (Р)

Таким образом, подавляющее большинство этих операций можно поручить машине. Отправитель сам может выбрать на дисплее, из какого материала будет клон. Оператору останется следить за работоспособностью линии.

Проект состоит из пяти конструктивно независимых частей — платформы сканирования и контроля, робота-сборщика клонов, автоматических складов (4шт), робота-транспортировщика «SD-01» (1 поле) и мобильного робота для 2 этапа «SD-02».

#### Платформа сканирования и контроля

Технические характеристики:

- габаритные размеры без навесных устройств 350x150x65мм
- масса 1,5кг
- напряжение питания 7,5-9v

Конструкция на основе шагового двигателя вращает платформу сканирования и контроля синхронно за счёт ременной передачи. Для поднятия датчиков цвета на следующий уровень используется серво двигатель. Мозговым элементом устройства является Arduino Nano.

#### Описание платы управления Arduino Nano

Напряжение питания 5В. Входное питание 7-12В (рекомендованное). Количество цифровых пинов – 14, из них 6 могут использоваться в качестве выходов ШИМ. 8 аналоговых входов. Максимальный ток цифрового выхода 40 мА. Флэш- память 16 Кб или 32 Кб, в зависимости от чипа. ОЗУ 1 Кб или 2 Кб, в зависимости от чипа EEPROM 512 байт или 1 Кб. Частота 16 МГц Размеры 19 x 42 мм Вес 7 г.

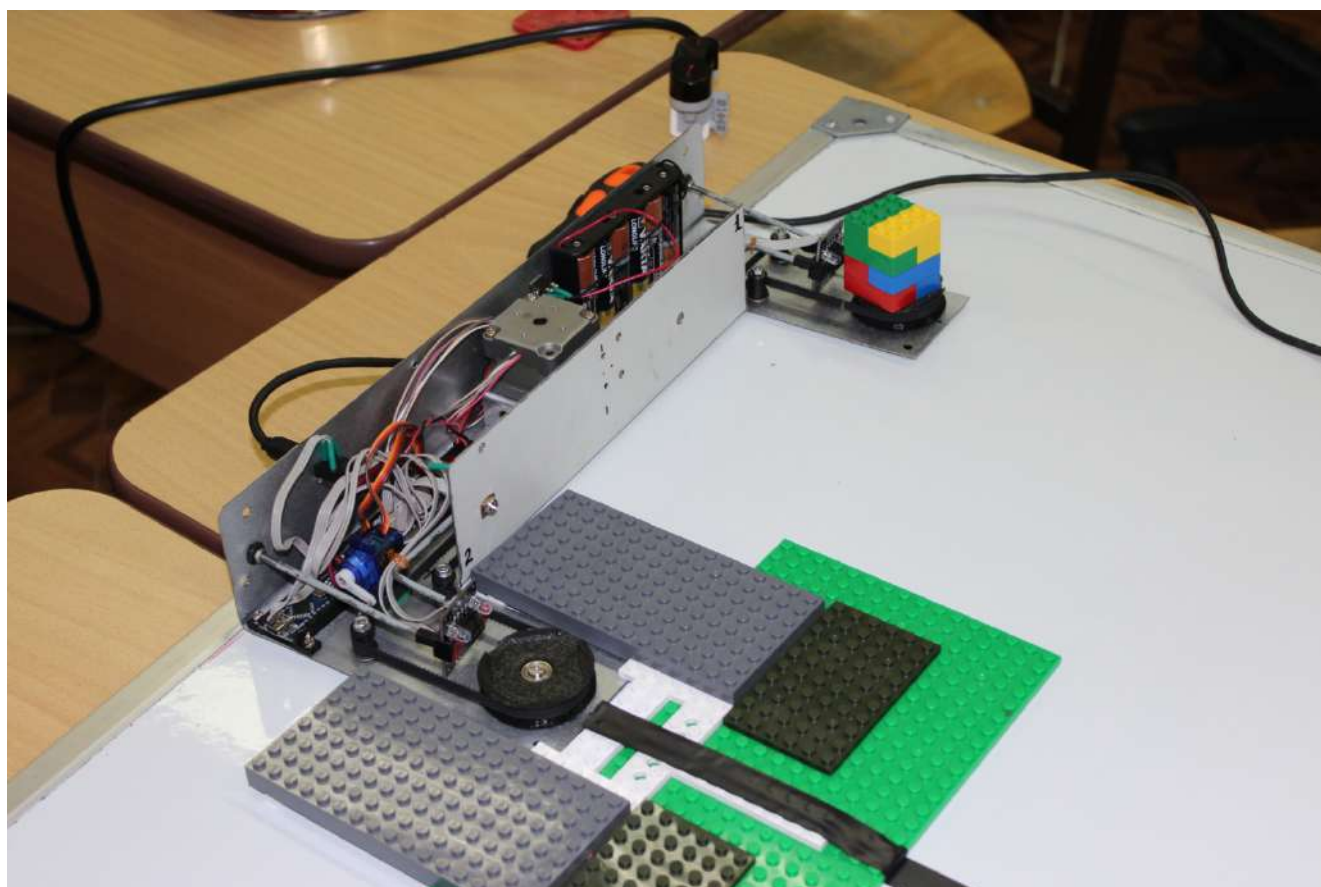
Для программирования платы Arduino Nano мы используем Arduino IDE. Для подключения компьютера к плате используется USB кабель. Все стадии

Инженерная книга команды «ПОЗИТ», Пушкинский район Московская область операции программирования можно пронаблюдать в окне Программного обеспечения на мониторе компьютера.

Для выполнения поставленных задач к Arduino Nano подключены:

Шаговый двигатель(12-9), 2 параллельно подключенных датчика цвета tcs-3200 (8-4), серводвигатель (3), кнопка старта (2) и разъём для подключения кабеля передачи данных (1-0).

Передача данных реализована по протоколу Serial.



Платформа сканирования и контроля ” SD-01“

Тип мотора	Функциональное назначение
Сервопривод GS09A	Перемещение планки с датчиками
TEAC 14PM-M204-P2	Вращение платформ

	сканирования и контроля
<b>Тип датчика</b>	<b>Функциональное назначение</b>
Датчик цвета tcs-3200 (1)	Сканирования цвета
Датчик цвета tcs-3200 (2)	Кирпичиков кубика

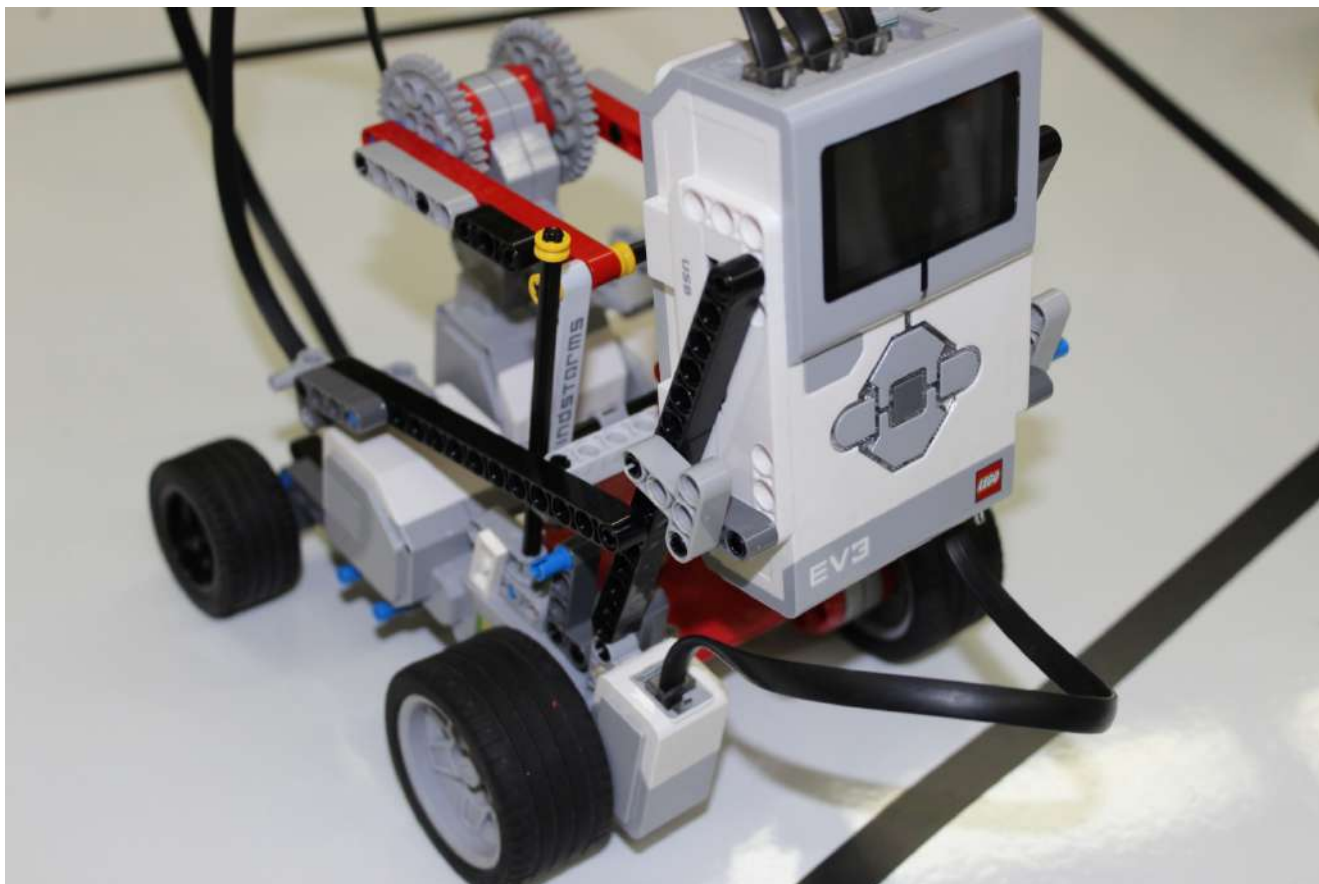
## Алгоритм работы Платформа сканирования и контроля

Программа начинается с того что платформа сканирования считывает поворачивается на 180 градусов считывает и поднимается затем цикл повторяется. Цикл является замкнутым благодаря тому что мы считываем углы кубика, а не его грани как планировалось изначально. Сразу после 1 сканирования значение полученное с датчика передаётся по Serial порту передаётся на сборочную платформу на 1ом поле.

После 1 сканирования робот ожидает нажатия кнопки, которое подтверждает наличия кубика на контрольной платформе. Переключаются датчики. Поворот на 45 градусов. Запускается протокол сканирования, но в отличии от первого раза вместо передачи полученных с датчика данных происходит их сравнение с первыми показаниями заблаговременно записанными в массив.

Листинг программы платформы приведен в Приложении №2.

## Робот-Манипулятор



### Принцип работы

Захват закреплён под крышкой что делает конструкцию более герметичной. Захват представляет из себя наше новое технологическое решение которое нашло широкое применение в этом проекте. После некоторых раздумий мы пришли к выводу что наилучшим способом взятия лего кубика является захват его маленьким кирпичиком сверху. Преимуществом данного захвата является то что он не чувствителен к вибрациям создаваемым при передвижении робота.

Управление манипулятором осуществляется логическими уровнями, обрабатываемыми микросхемами манипулятора.

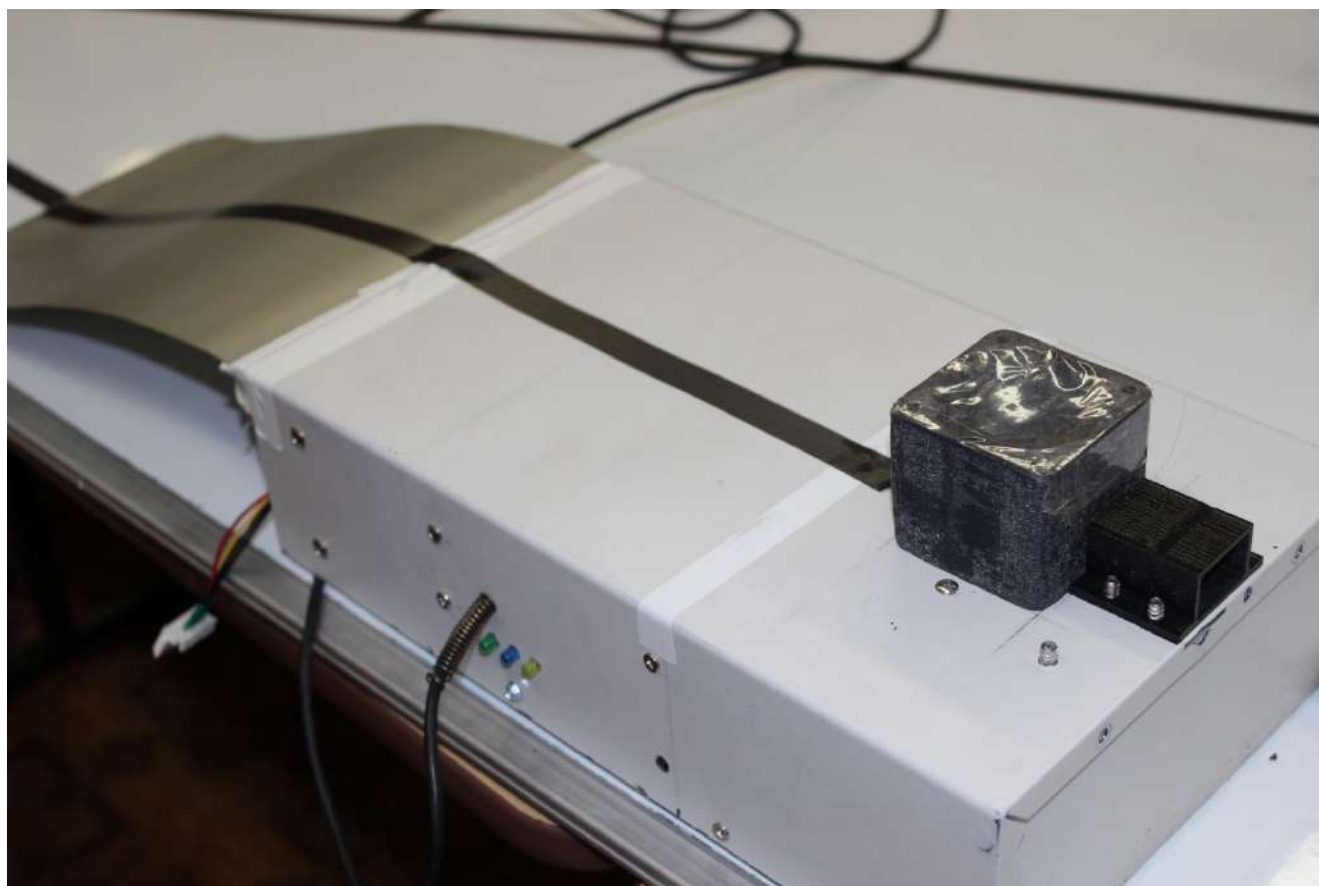
Сам робот играет роль доставщика готового изделия на склад, а так же является частью сборочного процесса, собирая кубик своим манипулятором.

Инженерная книга команды «ПОЗИТ», Пушкинский район Московская область  
Такое решение позволяет экономить ресурсы, как материалы на создание, так и топливо на работу.

### Общее описание и алгоритм функционирования

Сборочная платформа состоит из основной вращающейся круглой платформы (на которую помещаются кубики), пылесоса (часть пневмодоставки) и светодиодов (сигнализирующей о различных этапах сборки кубика)

При завершении сканирования платформа начинает сборку кубика, втягивая кирпичики соответствующего цвета. После доставки кирпичиков на платформу один из светодиодов информирует робота манипулятора о том что пришло время собирать кубик. Затем, продолжается втягивание кирпичиков и сборка продолжается. После сборки платформа сообщает о завершении работы и о цвете первого кубика роботу манипулятору при помощи мигания светодиодом.



Габаритные размеры участка:

Длина	Ширина	Высота
63 См	19 См	13 См

## Программное обеспечение автоматизированного участка «SD-02»

Основой управления сборочной платформой является микроконтроллер ARDUINO.

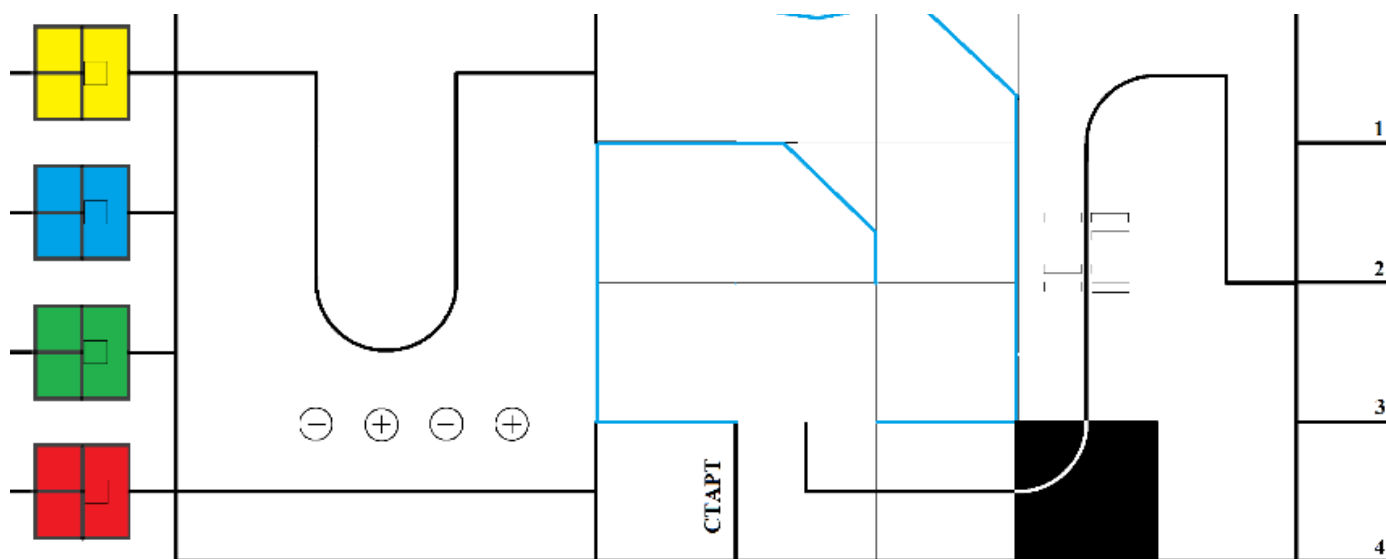
Для оптимизации программы она была разделена на отдельные модули.

Листинг программы приведён в приложении №3.

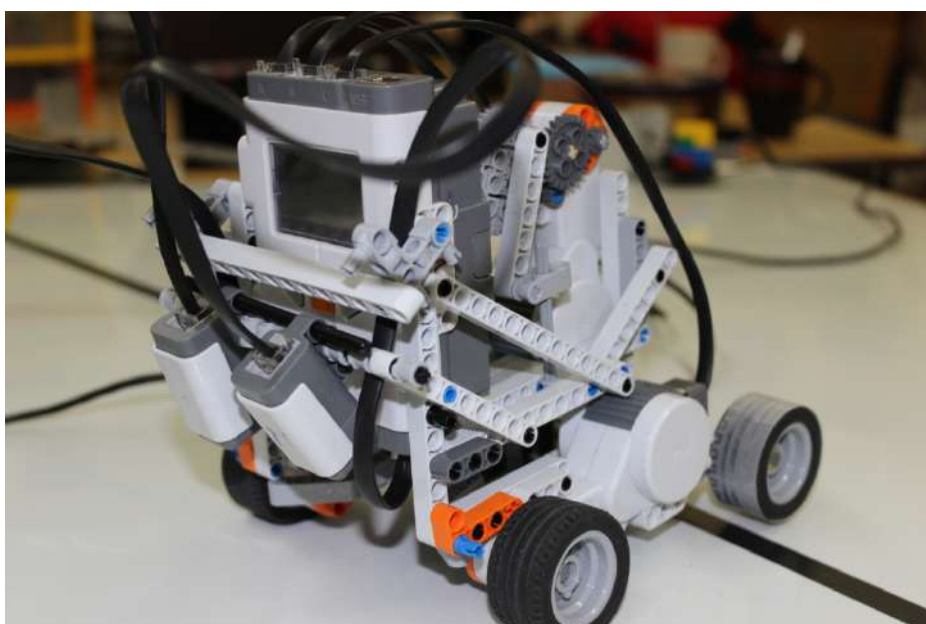


## ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ ПРОЕКТА. ЭТАП 2

Вид поля



Внешний вид робота



Инженерная книга команды «ПОЗИТ», Пушкинский район Московская область  
Датчик расстояния служит для подсчёта банок.

Датчики цвета служат для проезда по траектории. Робот выполнен из набора Lego Mindstorms. Робот должен захватить готового клона захватом который по устройству похож на захват робота манипулятора из 1 этапа, а затем преодолевая препятствия довести его до контрольной платформы платформы.

## ПРАВИЛА ТЕХНИКИ БЕЗОПАСНОСТИ ПРИ РАБОТЕ ЗА КОМПЬЮТЕРОМ.

Компьютер – это сложное устройство, работающее под напряжением, поэтому на него тоже распространяются техника безопасности к электрическим устройствам. Применимо к компьютеру соблюдайте основные правила:

- Не работайте с компьютером при наличии внешних повреждений корпуса или изоляции силовых кабелей. В этом случае требуется замена кабелей или обращение в сервисный центр.

- Не кладите на корпус системного блока и не храните на нем разные предметы, особенно тяжелые, т.к. в этом случае может возникнуть вибрация, которая может вызвать нарушения работы компьютера.

- Не рекомендуется включать компьютер в розетки без заземления. Розетки и вилки должны быть цельными, без повреждений.

- Не включайте компьютер в помещении с высокой влажностью.

- Не оставляйте работающий ПК без присмотра длительное время.

- Провода и силовые кабеля компьютера должны быть расположены так, чтобы исключить возможность наступить на них или поставить что-то тяжелое.

1. После окончания сборки, проверки на компьютере, конструкция разбирается, детали укладываются в коробку, компьютер выключается и сдается учителю.

2. По всем вопросам неполадок компьютера обращаться к учителю-консультанту.

ЖУРНАЛ РАБОТЫ НАД ПРОЕКТОМ.

Дата	Направление деятельности	Вид деятельности	Результат
10.09.2019	Постановка задачи	Поиск производств, направление работы которых совпадает с темой «ИКаР» сезона 2018-2019	Заключение договора о сотрудничестве.
01.10 – 06.10.2019	Сбор и анализ информации по теме	Поиск информации в Интернете и литературе, рекомендованной научными консультантами	Собраный материал оформлен в формате pptx и презентован на РМО заместителей директора по УВР
19.10-23.11	Изучение основных видов передач	Сборка моделей на различных передачах, исследование эффективности применения для конкретных узлов будущей производственной линии	Установлены виды передач, которые наиболее эффективно могут быть использованы для создания производственной линии
ноябрь	Изучение работы реального производства	Экскурсия на склад УФПС МО ф-л ФГУП "Почта России" и почтовое отделение "Правдинский 1"	Получено представление о реальной технологическом процессе.
ноябрь	Изучение работы	Экскурсия в ЗАО	Получено представление о реальных

Инженерная книга команды «ПОЗИТ», Пушкинский район Московская область

<b>Дата</b>	<b>Направление деятельности</b>	<b>Вид деятельности</b>	<b>Результат</b>
ь	реального производства	«Зеленоградское», ООО «ПОЗИТ», ЗАО «РОСТАГРОКОМПЛЕКС»	производственных линиях. Уточнена необходимость автоматизации отдельных технологических операций
19 декабря-15 января	Проектирование производственной линии.	Создание схемы линии и оценка эффективности расположения основных узлов. Сборка.	Найдено наиболее рациональное расположение основных узлов линии
13-18 января	Программирование	Установка программируемых блоков и корректировка механизмов.	Запрограммированы блоки. Скорректирована конструкция
20-22 января	Пробный заезд.	Установка комплекса на поле и тренировка	Учтены недочеты в работе программы
23 января	Проверка работы всей производственной	Тестирование. Оформление инженерной книги	Учтены недоработки. Распечатана и оформлена инженерная книга проекта

Инженерная книга команды «ПОЗИТ», Пушкинский район Московская область

Дата	Направление деятельности	Вид деятельности	Результат
	линии.		
27 январ я	Региональные соревнования ИКАР 2019	Соревнования, защита инженерной книги	

## ПРИЛОЖЕНИЯ

### Приложение 1. Оптические датчики

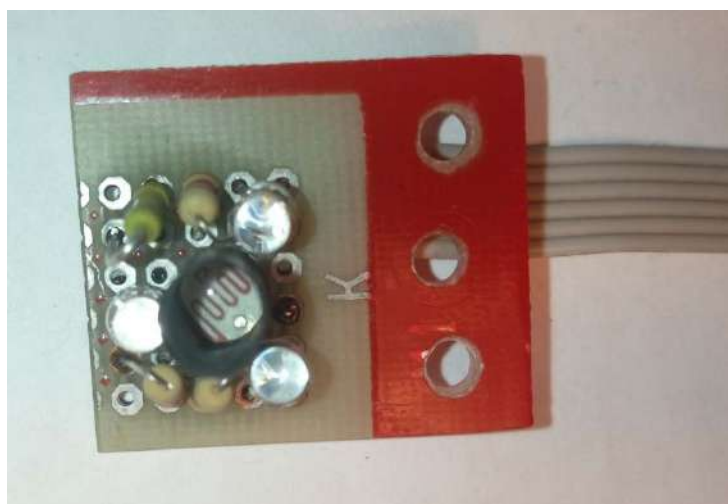
#### Датчик отслеживания линий

Как правило, для обнаружения линии используется инфракрасный датчик. Он создан для обнаружения с помощью инфракрасного излучения объектов, находящихся в непосредственной близости. Он состоит из инфракрасного светодиода, закрытого синим, прозрачным для ИК-лучей, колпаком, и инфракрасного фототранзистора, закрытого черным колпаком. Когда инфракрасное излучение, испускаемое светодиодом, отражается от поверхности и возвращается в окно фототранзистора, попадая на его базу, то возникает ток проводимости. Чем больше излучения попадает на базу с фототранзистора, тем больше его ток проводимости. Но, к сожалению, такое устройство не подходит для корректного выполнения необходимых нам задач, так как получаемый результат сильно зависит от расстояния до отражающей поверхности. При наличии неровностей поля (складки, стыки плит подложки) возможны ошибки, приводящие к неправильной интерпретации расположения линий и сбои в маршруте перемещения. В связи с этим, мы используем оптический датчик линии. Для уменьшения влияния внешних источников света и ограничения зоны зрения датчика корпус фоторезистора закрывается непрозрачным цилиндром.

Рабочий диапазон измерения расстояния до линии или пола составляет **10мм.**

Выходное напряжение датчика изменяется **от 3 до 20 В.** Таким образом, оптический датчик линии будет подходящим для использования его в качестве

датчика отслеживания линий.



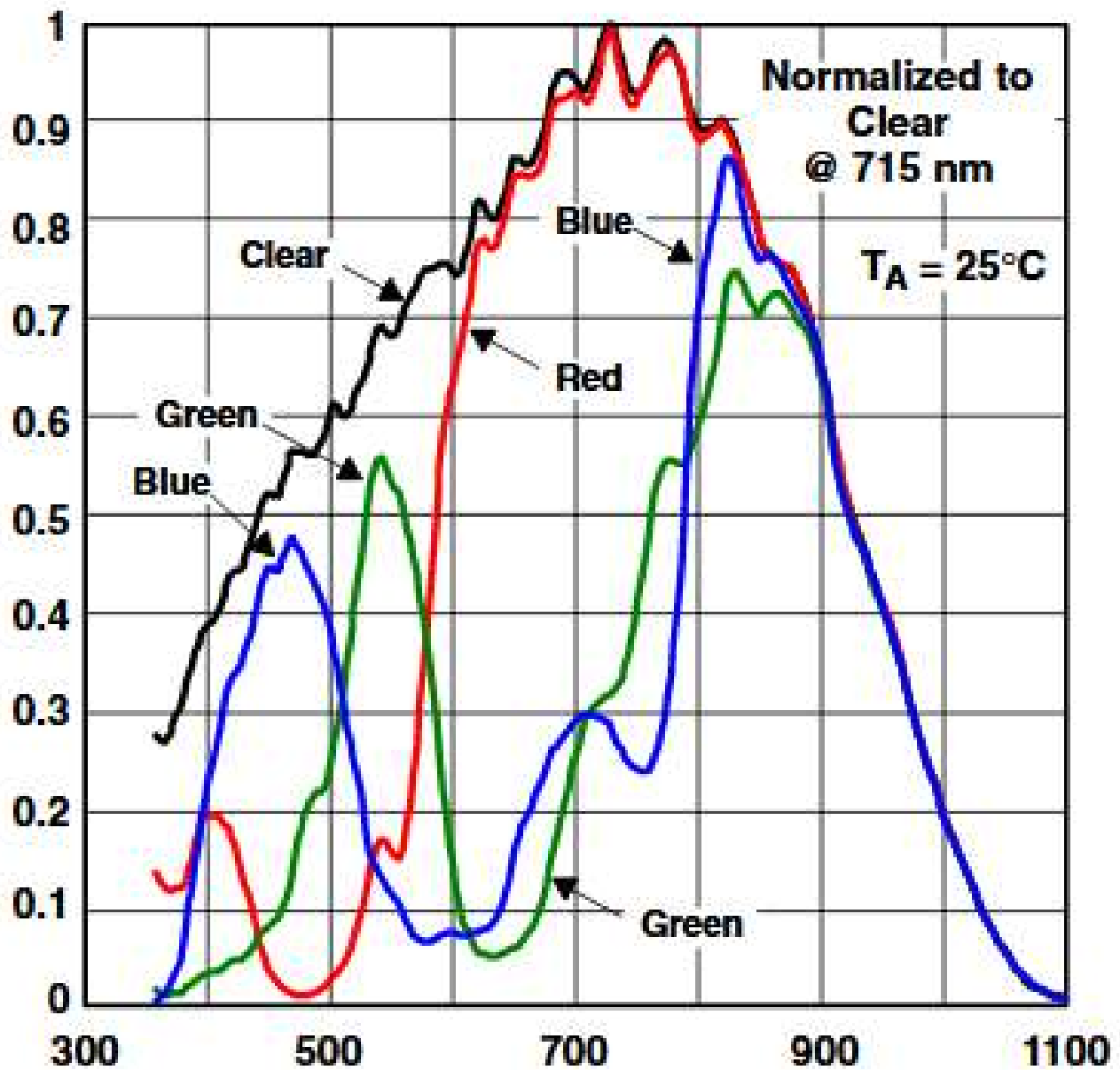
Датчики изготовлены самостоятельно на макетных платах.

Датчик определения цвета

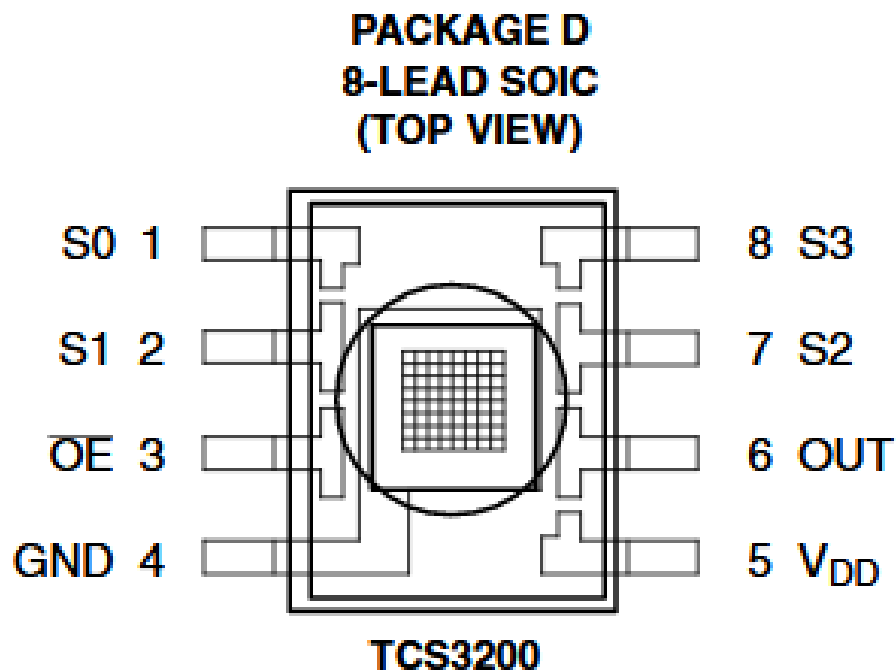
Для определения цвета кирпичика кубики было решено использовать высокоточный заводской датчик tcs-3200

Спектральная чувствительность фотодиодов

## PHOTODIODE SPECTRAL RESPONSIVITY







S0 –S1 Используются для масштабирования сигнала с датчика, мы используем масштаб 20%, так же доступны масштабы 2% и 100%.

S2-S3 Используются для переключения светодиодов с соответствующими фильтрами.

OE После определенных модификаций используется для отключения датчика

OUT Используется для вывода значений с фотодиодов

Инженерная книга команды «ПОЗИТ», Пушкинский район Московская область  
ПРИЛОЖЕНИЕ №2. Листинг программы робота-транспортировщика

```
#include <Servo.h> //используем библиотеку для работы с сервоприводом
```

```
#define ROBOButton 3
```

```
#define OE1 4
```

```
#define OE2 5
```

```
#define S3 6
```

```
#define S2 7
```

```
#define sensOut 8
```

```
int BlockColor[8];
```

```
// для хранения частоты, считанной фотодиодами:
```

```
int RTime = 30;
```

```
int GTime = 30;
```

```
int BTime = 30;
```

```
int RVal = 30;
```

```
int GVal = 30;
```

```
int BVal = 30;
```

```
int YTime = 30;
```

```
byte reserv = 0;
```

```
bool CubClon;
```

```
Servo servo; //объявляем переменную servo типа Servo
```

```
void TestColor(){
```

```
    //delay(100);
```

```
    // включаем фотодиоды без фильтра:
```

```
    digitalWrite(S2,HIGH);
```

```
    digitalWrite(S3,LOW);
```

```
    // считываем длительность
```

```
    YTime = pulseIn(sensOut, LOW);
```

```
    //Serial.println(YTime);
```

```
    // выводим значение яркости
```

```
    //Serial.print("Y = ");
```

```
    //Serial.println(YTime);
```

```
    // включаем R-фотодиоды
```

```
    digitalWrite(S2,LOW);
```

```
    digitalWrite(S3,LOW);
```

Инженерная книга команды «ПОЗИТ», Пушкинский район Московская область  
// считываем выходную частоту:

```
RTime = pulseIn(sensOut, LOW);
```

```
// выводим значение R
```

```
//Serial.print(" R = ");
```

```
int RVal = YTime*5/RTime;
```

```
//Serial.println(RTime);
```

```
// включаем G-фотодиоды
```

```
digitalWrite(S2,HIGH);
```

```
digitalWrite(S3,HIGH);
```

```
// считываем длительность
```

```
GTime = pulseIn(sensOut, LOW);
```

```
// выводим значение G
```

```
//Serial.println(GTime);
```

```
//Serial.print(" G = ");
```

```
GVal = YTime*7/GTime;
```

```
//Serial.print(GVal);
```

```
// включаем B-фотодиоды
```

```
digitalWrite(S2,LOW);
```

```
digitalWrite(S3,HIGH);
```

```
// считываем длительность
```

Инженерная книга команды «ПОЗИТ», Пушкинский район Московская область  
BTime = pulseIn(sensOut, LOW);

// выводим значение B

//Serial.print(" B = ");

//Serial.println(BTime);

BVal = YTime\*5/BTime;

//Serial.print(BVal);

if(RVal>GVal&&RVal>BVal){

  reserv = 1;

  // Serial.println("R");

}

  if(GVal>RVal&&GVal>BVal){

    reserv = 2;

    //Serial.println("G");

}

  if(BVal>RVal&&BVal>GVal){

    reserv = 3;

    //Serial.println("B");

}

if(RVal>BVal&&GVal>BVal){

  reserv = 4;

```
// Serial.println("Y");
```

```
}
```

```
//if(GVal==RVal==BVal){
```

```
// Serial.println("XXX");
```

```
}
```

```
void MotorStep(int t){
```

```
    for ( int i=1; i<=74; i++)
```

```
{
```

```
digitalWrite(9,0);
```

```
digitalWrite(10,1);
```

```
digitalWrite(11,0);
```

```
digitalWrite(12,0);
```

```
delay(t);
```

```
digitalWrite(9,0);
```

```
digitalWrite(10,0);
```

```
digitalWrite(11,0);
```

```
digitalWrite(12,1);
```

```
delay(t);
```

```
digitalWrite(9,1);
```

```
digitalWrite(10,0);
```

```
digitalWrite(11,0);
```

```
digitalWrite(12,0);
```

```
delay(t);
```

```
digitalWrite(9,0);
```

```
digitalWrite(10,0);
```

```
digitalWrite(11,1);
```

```
digitalWrite(12,0);
```

```
delay(t);
```

```
}
```

```
delay(100);
```

```
TestColor();
```

```
digitalWrite(9,0);
```

```
digitalWrite(10,0);
```

```
digitalWrite(11,0);
```

```
digitalWrite(12,0);
```

```
}
```

```
void setup() {  
  
  // выставляем контакты S0, S1, S2 и S3 в режим OUTPUT:  
  
  pinMode(OE1, OUTPUT);  
  
  pinMode(OE2, OUTPUT);  
  
  pinMode(S2, OUTPUT);  
  
  pinMode(S3, OUTPUT);  
  
  pinMode(ROBOButton, INPUT);  
  
  
  servo.attach(3); //привязываем привод к порту 3  
  
  
  servo.write(90); //ставим вал в начальное положение  
  
  // выставляем контакт sensOut в режим INPUT:  
  
  pinMode(sensOut, INPUT);  
  
  
  
  // задаем масштабирование частоты 100%:  
  
  digitalWrite(OE1,LOW);  
  
  digitalWrite(OE2,HIGH);  
  
  
  
  // запускаем последовательную коммуникацию:
```



```
//Запуск протокола сканирования
```

```
delay(500);
```

```
for(int i = 0;i<8){
```

```
//Serial.println("SENSOR : 1");
```

```
//Serial.println("Test1");
```

```
TestColor();
```

```
BlockColor[i] = reserv;
```

```
i++;
```

```
Serial.write(reserv);
```

```
delay(100);
```

```
MotorStep(6);
```

```
BlockColor[i] = reserv;
```

```
i++;
```

```
Serial.write(reserv);
```

```
if(i==2){
```

Инженерная книга команды «ПОЗИТ», Пушкинский район Московская область  
servo.write(115); //поднимаем на 2 этаж

delay(1200);

}

if(i==4){

servo.write(135); //поднимаем на 3 этаж

delay(1200);

}

if(i==6){

servo.write(155); //поднимаем на 4 этаж

delay(1200);

}

if(i==8){

// Serial.println("Цикл сканирования завершён! ");

CubClon = false;

digitalWrite(OE1,HIGH);

digitalWrite(OE2,LOW);

servo.write(90); //опускаем на 1 этаж

}

}

}

```
void loop() {  
  
    if(ROBOButton == 1){  
  
        CubClon = true;  
  
        delay(100);  
  
    }  
  
    if(ROBOButton == 0 && CubClon==true){  
  
        for(int i = 0;i<8;){  
  
            //Serial.println("SENSOR : 2");  
  
            TestColor();  
  
            if(reserv == BlockColor){  
  
                //Serial.println("ok");  
  
            }  
  
            i++;  
  
            Serial.write(reserv);  
  
        }  
  
    }  
  
}
```

Инженерная книга команды «ПОЗИТ», Пушкинский район Московская область  
delay(100);

MotorStep(6);

if(reserv == BlockColor){

    //Serial.println("ok");

}

i++;

Serial.write(reserv);

if(i==2){

    servo.write(115); //поднимаем на 2 этаж

    delay(1200);

}

if(i==4){

    servo.write(135); //поднимаем на 3 этаж

    delay(1200);

}

if(i==6){

    servo.write(155); //поднимаем на 4 этаж

    delay(1200);

}

}

}

}

Инженерная книга команды «ПОЗИТ», Пушкинский район Московская область  
: в программу могут быть внесены изменения, не ухудшающие её функционал.

Инженерная книга команды «ПОЗИТ», Пушкинский район Московская область  
Приложение №3. Листинг программы обработки заготовок

```
int кнопка;
int S1;
int S2;
int S3;
int R;
int G;
int B;
int Y;
int i;
int p;

void setup()
{
pinMode(A5,INPUT); //фотодатчик
pinMode(A2,INPUT); //кнопка
pinMode( 2,OUTPUT ); //мотор1
pinMode( 3,OUTPUT ); //мотор2
pinMode( 4,OUTPUT ); //мотор3
pinMode( 5,OUTPUT ); //мотор4
pinMode( 6,OUTPUT ); //фаза1
pinMode( 7,OUTPUT ); //фаза2
pinMode( 8,OUTPUT ); //фаза3
pinMode( 9,OUTPUT ); //фаза4
pinMode( 10,OUTPUT ); //мотор5
pinMode( 11,OUTPUT ); //красный светодиод
pinMode( 12,OUTPUT ); //зеленый светодиод
pinMode( 13,OUTPUT ); //синий/ светодиод
digitalWrite(2,1); //отключение мотора1
digitalWrite(3,1); //отключение мотора2
digitalWrite(4,1); //отключение мотора3
digitalWrite(5,1); //отключение мотора4
digitalWrite(6,0); //отключение фазы1
digitalWrite(7,0); //отключение фазы2
digitalWrite(8,0); //отключение фазы3
digitalWrite(9,0); //отключение фазы4
digitalWrite(11,1); //отключение красного светодиода
```

```
digitalWrite(12,1); //отключение зеленого светодиода
```

```
digitalWrite(13,1); //отключение синего светодиода
```

```
motorinit();
```

```
}
```

```
void motor1( int n, int t) //вращение по часовой стрелке; угол - n, ширина импульса - t
```

```
{
```

```
for ( int i=1; i<=n; i++)
```

```
{
```

```
digitalWrite(6,1);
```

```
delay(t);
```

```
digitalWrite(6,0);
```

```
digitalWrite(7,1);
```

```
delay(t) ;
```

```
digitalWrite(7,0);
```

```
digitalWrite(8,1);
```

```
delay(t);
```

```
digitalWrite(8,0);
```

```
digitalWrite(9,1);
```

```
delay(t);
```

```
digitalWrite(9,0);
```

```
}
```

```
}
```

```
void motor2(int n, int t) //вращение против часовой стрелки; угол - n, ширина импульса - t
```

```
{
```

```
for ( int i=1; i<=n; i++)
```

```
{
```

```
digitalWrite(9,1);
```

```
delay(t);
```

```
digitalWrite(9,0);
```

```
digitalWrite(8,1);
```

```
delay(t) ;
```

```
digitalWrite(8,0);
```

```
digitalWrite(7,1);
```

```
delay(t);
```

```
digitalWrite(7,0);
```

```
digitalWrite(6,1);
```

```
delay(t);
digitalWrite(6,0);
}
}

void motorinit() //инициализация моторов
{
digitalWrite(2,0);
motor1(1,100);
motor2(1,100);
digitalWrite(2,1);
digitalWrite(3,0);
motor2(1,100);
motor1(1,100);
digitalWrite(3,1);
digitalWrite(4,0);
motor1(1,100);
motor2(1,100);
digitalWrite(4,1);
digitalWrite(5,0);
motor1(1,100);
motor2(1,100);
digitalWrite(5,1);
digitalWrite(10,1);
delay(100);
digitalWrite(10,0);
}

void definecolor() //измерения цвета
{
digitalWrite(11,0); //вкл красный светодиод
delay(200);
S1=analogRead(A5); //измерение красного
digitalWrite(11,1); //выкл красный
digitalWrite(12,0); //вкл зеленый
delay(200);
S2=analogRead(A5); //измерение зеленого
digitalWrite(12,1); //выкл зеленый
```



```
delay(200);  
digitalWrite(13,0); //вкл синий  
delay(200);  
S3=analogRead(A5); //измерение синего  
digitalWrite(13,1); //выкл синего  
delay(200);  
}
```

//упрвление мотором перемещения кубика (длительность импульсов меняется от tmax до tmin при разгоне)

```
void motor3(int n, int tmax, int tmin,int dt)  
{  
  int t=tmax;  
  for(int i=0;i<n;i++)  
  {  
    digitalWrite(6,1);  
    if (t>tmin)t=t-dt;  
    delay(t);  
    digitalWrite(6,0);  
    digitalWrite(7,1);  
    if (t>tmin)t=t-dt;  
    delay(t);  
    digitalWrite(7,0);  
    digitalWrite(8,1);  
    if (t>tmin)t=t-dt;  
    delay(t);  
    digitalWrite(8,0);  
    digitalWrite(9,1);  
    if (t>tmin)t=t-dt;  
    delay(t);  
    digitalWrite(9,0);  
  }  
}
```

```
void loop()  
{  
  while(analogRead(A2)<500) //ожидание кнопки  
  {delay(100);}  
}
```

```
delay(3000); //ожидание пока робот положит кубик

definecolor();//измерения цвета

//определение цвета
if (((S1-S2-S3)>-300 ) and ((S1-S2-S3)<0))
  {R=1;B=0;G=0;Y=0;}
if (((S1-S2-S3)>-500) and ((S1-S2-S3)<-300))
  {R=0;Y=1;G=0;B=0;}
if (((S1-S2-S3)>-700) and ((S1-S2-S3)<-500) and (S3<600))
  {R=0;B=0;G=1;Y=0;}
if (((S1-S2-S3)>-1000) and ((S1-S2-S3)<-700) and (S3>600))
  {R=0;B=1;G=0;Y=0;}

//выбор цвета инструмента
//если красный
if (R==1)
{
  digitalWrite(2,0);
  motor2(1,30);
};
//если зелёный
if (G==1)
{
  digitalWrite(2,0);
  motor2(2,30);
};
//если синий
if (B==1)
{
  digitalWrite(2,0);
  motor2(3,30);
};
//если жёлтый
if (Y==1)
{
  digitalWrite(2,0);
```

```
motor2(4,30);
```

```
};
```

```
digitalWrite(2,1);
```

```
//подъезд к I инструменту
```

```
digitalWrite(3,0);
```

```
motor3(10,30,10,1);
```

```
digitalWrite(3,1);
```

```
//работа I инструмента
```

```
digitalWrite(4,0);
```

```
for (p=1;p<=5;p++)
```

```
{
```

```
motor2(15,5);
```

```
motor1(15,5);
```

```
}
```

```
digitalWrite(4,1);
```

```
delay(1000);
```

```
//возврат инструмента
```

```
//если красный
```

```
if (R==1)
```

```
{
```

```
digitalWrite(2,0);
```

```
motor2(4,30);
```

```
};
```

```
//если зелёный
```

```
if (G==1)
```

```
{
```

```
digitalWrite(2,0);
```

```
motor2(3,30);
```

```
};
```

```
//если синий
```

```
if (B==1)
```

```
{
```

```
digitalWrite(2,0);
```

```
motor2(2,30);
```

```
};
```

```
//если жёлтый
if (Y==1)
{
digitalWrite(2,0);
motor2(1,30);
};
digitalWrite(2,1);

//подъезд ко II инструменту
digitalWrite(3,0);
motor3(25,30,10,1);
digitalWrite(3,1);

//работа II инструмента
digitalWrite(5,0);
motor2(20,5);
digitalWrite(5,1);

digitalWrite(10,1);
delay(5000);
digitalWrite(10,0);

digitalWrite(5,0);
motor1(20,5);
digitalWrite(5,1);

//работа III инструмента
digitalWrite(3,0);
motor3(115,30,10,1);
digitalWrite(3,1);

//Завершение
delay(5000);
motorinit();
}
```

Примечание: в программу могут быть внесены изменения, не ухудшающие её функционал.

1. Том Иго Arduino, датчики и сети для связи устройств – MakerMedia/ БХВ-Петербург, 2017г-531 с.
2. Юрий Ревич. Занимательная микроэлектроника – Санкт-Петербург, 2012г-557 с
3. Микросхемы управления двигателями – ОДЭКА, 2003г – 286с
4. Копосов Д.Г. Технология. Робототехника 6 класс: учебное пособие- М.:БИНОМ. Лаборатория знаний,2017.-128с.: ил.
5. Копосов Д.Г. Технология. Робототехника 7 класс: учебное пособие- М.:БИНОМ. Лаборатория знаний,2017.-128с.: ил.
6. Копосов Д.Г. Технология. Робототехника 8 класс: учебное пособие- М.:БИНОМ. Лаборатория знаний,2017.-128с.: ил.
7. Копосов Д.Г. Технология. Робототехника 9 класс: учебное пособие- М.:БИНОМ. Лаборатория знаний,2017.-128с.: ил.
8. Копосов Д.Г. Технология. Робототехника 10 класс: учебное пособие- М.:БИНОМ. Лаборатория знаний,2017.-128с.: ил.
9. Копосов Д.Г.Первый шаг в робототехнику: практикум для 5-6 классов-М.:БИНОМ. Лаборатория знаний,2012.-286с.:ил.
10. Головань К.Б.и др. Технология: сборник проектов.- М.:Издательство"Перо",2016.-184с.
11. Мякушко А.А. Основы образовательной робототехники: уч.-метод.пособие.для.слушателей.курса-М.:Издательство"Перо",2014.-80с.:илл.
12. Яковлева З.В. Образовательная робототехника на уроках информатики. Издательство "Перо",2014.-48с.: илл.
13. Овсяницкая Л.Ю. Курс программирования робота EV3 в среде Lego Mindstorms EV3.-М.:Издательство "Перо", 2016.-300с.