**РАЙОННАЯ УЧЕБНО-ИССЛЕДОВАТЕЛЬСКАЯ КОНФЕРЕНЦИЯ**

**«ЮНОСТЬ ПИНЕЖЬЯ»**

Направление **Программирование и информационные технологии**

**Сортировщик с системой технического зрения**

**Исследовательская работа**

Выполнена учеником 10 класса

муниципального бюджетного

общеобразовательного учреждения

«Междуреченская СШ №6»

Родионовым Германом Леонидовичем

Научный руководитель – учитель

Муниципального бюджетного

Общеобразовательного учреждения

«Междуреченская СШ №6»

Игнатьев Павел Алексеевич

**с. Карпогоры, 2023**

**Оглавление**

[Введение 3](#_Toc118118838)

[Принцип работы машинного зрения 4](#_Toc118118839)

[Сортировщик 5](#_Toc118118840)

[Программа для сортировщика 7](#_Toc118118841)

[Заключение 8](#_Toc118118842)

[Библиографический список 9](#_Toc118118843)

[Приложение 1](#_Toc118118844)

[Программа для сортировщика 10](#_Toc118118845)

[Приложение 2](#_Toc118118846)

[Интерфейс настройки модуля TrackingCam с монитором порта Arduino 11](#_Toc118118847)

[Приложение 3](#_Toc118118848)

[Модель сортировщика 13](#_Toc118118849)

[Приложение 4](#_Toc118118850)

[Контроллер с камерой и микрокомпьютером 14](#_Toc118118851)

# Введение

В настоящее время системы технического зрения активно используются в различных отраслях экономики. Например, системы распознавания лиц контролируют доступ, помогают искать преступников. На промышленных конвейерах данные системы сортируют продукцию, в сельском хозяйстве помогают очищать товарное зерно от испорченного и прочее[3].

Цель – создать сортировщик с системой технического зрения.

Объект исследования – сортировщик.

Предмет исследования – программа для сортировщика.

В соответствии с данной целью были поставлены следующие задачи:

1. Описать принцип работы технического зрения.

2. Создать сортировщик.

3. Создать и описать программу для демонстрации работы технического зрения.

Методы исследования: теоретическое и практическое моделирование, эксперименты.

Гипотеза – с помощью системы технического зрения можно сортировать объекты.

# Принцип работы машинного зрения

Цифровая камера, в отличие от точечного фотореле или датчика цвета, представляет собой двумерный массив из тысяч оптических датчиков, что позволяет получить намного больше сведений об окружающей среде. Одной из ключевых задач систем является распознавание образов, заданных определенными признаками объектов сцены. Системы технического зрения состоят из аппаратной части и программного обеспечения. В настоящее время для обработки изображений в робототехнике получило широкое распространение библиотека OpenCV, состоящая из большого количества алгоритмов, поставляемых с документированным интерфейсом (API) и примерами приложений на самых популярных языках программирования. Процесс преобразования информации в системах технического зрения можно представить в виде этапов ввода информации, предварительной обработки, сегментации (выделении необходимых объектов), описания, распознавания и интерпретации. Как правило, при распознавании однотонных областей, задаются их цветовые и морфологические признаки. Есть специальные алгоритмы для распознавания различных маркеров, окружностей, геометрических фигур. Используя алгоритмы, основанные на нейронных сетях, можно решать задачи слежения за объектами и их распознавания (например, нейросеть архитектуры YOLO, встроенная в модуль технического зрения TrackingCam)[6].

# Сортировщик

Сортируемые объекты будут двигаться по наклонному жёлобу. Камера машинного зрения будет отслеживать их по заданным параметрам (в нашем случае – цвет). Если объект соответствует параметрам, заслонка будет открываться перед движущимся объектом, и он будет помещён в отсек для сортировки, иначе продолжит движение дальше. Для проверки гипотезы достаточно одного сервопривода.(Приложение 3)

В сортировщике будем использовать модуль TrackingCam. TrackingCam – модуль технического зрения, способный выполнять операции по распознаванию как одноцветных объектов, так и составных объектов, состоящих из нескольких цветовых областей. Разрешение матрицы модуля 640 на 480 пикселей, потребляемое напряжение 5 вольт[4].

Информация, поступающая с модуля камеры машинного зрения, будет использоваться в программе, загружаемой в компактный контроллер AR-DXL-IoT. Данный универсальный вычислительный модуль, предназначенный для взаимодействия с Dinamixel-совместимыми устройствами, представляет собой электронную плату с размещённым на ней вычислительным микроконтроллером Atmega2560, чипом беспроводной связи по интерфейсам Wi-Fi и Bluetooth, набором цифровых и аналоговых линий передачи связи, а также портами для подключения Dynamixel-совместимых устройств [1]. Для управления заслонкой сортировки, подключен сервопривод на 3 цифровой выход контроллера.(Приложение 4)

Для настройки камеры, в силу своей компактности, мы будем использовать одноплатный компьютер Raspberry Pi с подключённым к нему 3,5 дюймовым сенсорным TFT дисплеем разрешением 320 на 480 точек.

Зная цветовые признаки распознаваемой области, можно легко выявить её на исходном изображении, путём сравнения цветовых параметров каждого пикселя изображения с признаками распознаваемой области. Для того чтобы выбрать объект сцены во вкладке «Blob Detector», который нужно распознать, необходимо нажать «Pick color» и указать курсором на объект, который нужно распознать. После выполнения этого действия программное обеспечение модуля TrackingCam v3 автоматически подберёт набор оттенков для распознавания выбранного объекта. В интерфейсе настройки камеры данной вкладки можно настроить следующие параметры: «Binarization» (яркость, синяя цветоразность, красная цветоразность), «Morphology» (площадь распознаваемой области, округлость, вытянутость, выпуклость, площадь фигуры, размер и параметры окна).(Приложение 2)

# Программа для сортировщика

Поскольку весь процесс обмена данными с камерой по протоколу Dinamixel реализован в библиотеке, то, прежде всего, в новом скетче необходимо эту библиотеку подключить. Эта команда будет выглядеть следующим образом: «#include «TrackingCamDxl.h»»[5].Строкой «int servoPin = 3; Servo Servo1;» инициализируем выход для сервопривода управления заслонкой и задаём переменную для сервопривода[2]. У каждого Dinamixel-устройства свой id, id нашей камеры равен 1, что мы и указываем строчкой «TrackingCamDxl trackingCam(1);».

Строка «Servo1.attach(servoPin);» подключает Servo к указанному выходу, с которого осуществляется управление приводом. Этими строками «DxlMaster.begin(1000000);Serial.begin(1000000);» задаётся скорость обмена данными с монитором порта. Частота вывода данных будет 10 раз в секунду (delay(100);). Для контроля будет загораться индикаторный светодиод, свидетельствующий о нахождении заданного объекта (pinMode(10,OUTPUT);). В цикле мы будем использовать 5 переменных типа uint8, данные, о которых будем выводить в монитор порта(unit8\_t n=trackingCam.readBlobs(5);). Переменная «type» хранит значение типа объекта, «dummy» - это номер шаблона соответствия, «cx» - это центр распознанной области по оси x, «cy» - это центр распознанной области по оси y, «area» - показывает нам площадь объекта[6]. В нашем случае мы будем использовать выводимую переменную о площади заданного объекта с камеры машинного зрения. Если переменная «area» больше 10, то включается индикаторный светодиод, а сервопривод заслонки поворачивается на 90 градусов (Servo1.write(90);), иначе заслонка закрывается. Переход к следующему фрейму (кадру) задаётся строками «while(millis()-previousMillis<33)». (Приложение 1)

# Заключение

В ходе проделанной работы были уяснены принципы работы систем машинного зрения, разобрались с порядком использования модуля машинного зрения TrackingCam, получения данных с него и настройки. Была создана действующая модель сортировщика с использованием системы машинного зрения, написана программа.

В дальнейшем наработки по данной тематике могут быть использованы в других более совершенных устройствах. Работа имеет большой прикладной характер.

# Библиографический список

1. Arduino Mega 2560 [Электронный ресурс] [URL: https://arduino.ru/Hardware/ArduinoBoardMega2560](URL:%20https://arduino.ru/Hardware/ArduinoBoardMega2560)
2. Arduino библиотека Servo [Электронный ресурс] [URL: https://all-arduino.ru/biblioteki-arduino/arduino-biblioteka-servo/](URL:%20https://all-arduino.ru/biblioteki-arduino/arduino-biblioteka-servo/)
3. Машинное зрение – что это и где применяется [Электронный ресурс] URL: <https://www.cameraiq.ru/faq/mashinnoe-zrenie-chto-eto-i-gde-primeniaetsia/>
4. Модуль технического зрения TrackingCam [Электронный ресурс] <URL:https://robotgeeks.ru/collection/komplektuyuschie-dlya-avtonomnyh-robotov/product/modul-tehnicheskogo-zreniya-trackingcam/>
5. Программное обеспечение [Электронный ресурс] [URL: https://appliedrobotics.ru/?page\_id=633](URL:%20https://arduino.ru/Hardware/ArduinoBoardMega2560)
6. Техническое зрение роботов с использованием TrackingCam / С.А.Воротников, Е.А.Девятериков, А.О.Панфилов - Электронная книга, ООО «Прикладная робототехника», 2017.

# Приложение 1

## Программа для сортировщика

#include <TrackingCamDxl.h>

#include <DxlMaster.h>

#include <Servo.h>

const long unsigned int baudrate=1000000;

int servoPin = 3;

Servo Servo1;

TrackingCamDxl trackingCam(1);

unsigned long previousMillis=0;

void setup() {

Servo1.attach(servoPin);

DxlMaster.begin(1000000);

Serial.begin(1000000);

delay(100);

pinMode(10,OUTPUT);

}

void loop() {

uint8\_t n=trackingCam.readBlobs(5);

Serial.print('All blobs');

Serial.print(n);

Serial.print(trackingCam.blob[0].type,DEC);

Serial.print(' ');

Serial.print(trackingCam.blob[0].dummy,DEC);

Serial.print(' ');

Serial.print(trackingCam.blob[0].cx,DEC);

Serial.print(' ');

Serial.print(trackingCam.blob[0].cy,DEC);

Serial.print(' ');

Serial.print(trackingCam.blob[0].area,DEC);

Serial.println(' ');

if (trackingCam.blob[0].area>10)

{

Servo1.write(0);

digitalWrite(10,HIGH);

Servo1.write(90);

delay(100);

}

else{

digitalWrite(10,LOW);

Servo1.write(0);

delay(100);

}

while(millis()-previousMillis<33)

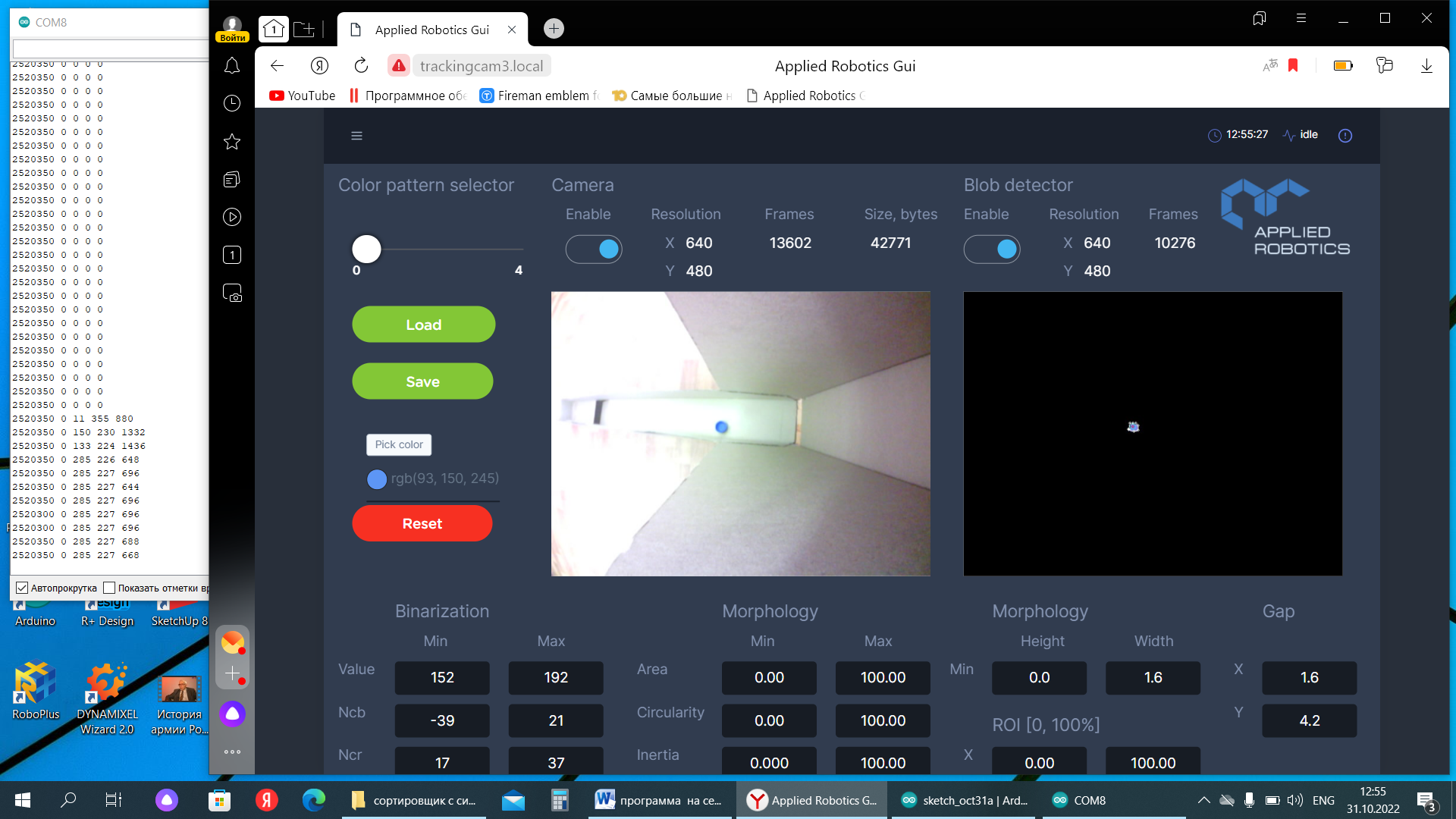
{};

previousMillis=millis();

}

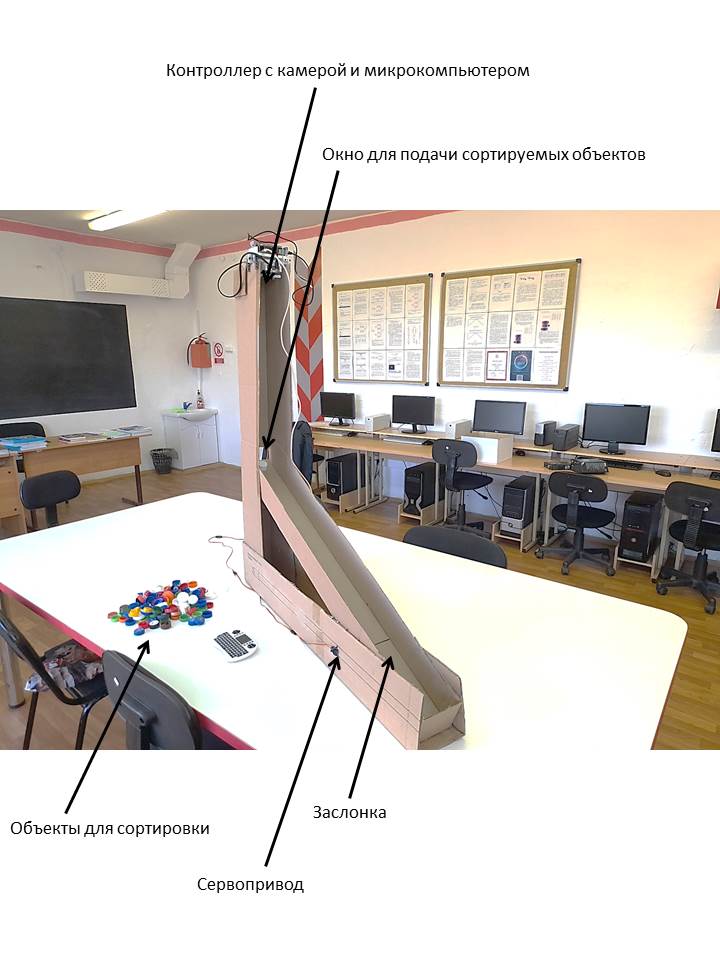
# Приложение 2

## Интерфейс настройки модуля TrackingCam с монитором порта Arduino



# Приложение 3

## Модель сортировщика



# Приложение 4

## Контроллер с камерой и микрокомпьютером

