

Министерство образования и науки Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение высшего  
профессионального образования  
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ  
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)  
Кафедра безопасности информационных систем (БИС)

ОТЧЕТ  
ПО РЕЗУЛЬТАТАМ  
производственной практики

Студент гр. 743

\_\_\_\_\_ Т.С. Койшинов

«\_\_\_»\_\_\_\_\_ 2016 г.

Руководитель

Аспирант каф. РЭТЭМ

\_\_\_\_\_ С.П. Шкарупо  
оценка

«\_\_\_»\_\_\_\_\_ 2016 г.

Томск 2016

## ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

на производственную практику студенту Койшинову Тимуру Саматулы группы 743, факультета безопасности.

1 Тема работы: Создание кухонного манипулятора

2 Исходные данные к работе:

2.1 Данные по программированию на ARDUINO

2.2 Данные по библиотеке OPENCV

3 Срок сдачи студентом законченной работы

---

4 Содержание производственной практики:

4.1 Обзор манипулятора;

4.2 Описание ARDUINO;

4.3 Описание библиотеки компьютерного зрения OPENCV;

4.4 Алгоритм работы программ:

- поиска предметов,
- управления манипулятором;

5 Содержание пояснительной записки:

- титульный лист;
- задание;
- реферат;
- содержание;
- введение;
- обзор манипулятора;
- обзор ARDUINO;
- обзор OPENCV;
- алгоритм работы программ;
- испытания;
- заключение;
- список использованных источников;

– приложения.

Отчет должен быть оформлена согласно ОС ТУСУР 01-2013.

6 Дата выдачи задания: \_\_\_\_\_

Задание согласовано:

Руководитель

\_\_\_\_\_

«\_\_\_\_» \_\_\_\_\_ 2016г. \_\_\_\_\_

Задание принято к исполнению

«\_\_\_\_» \_\_\_\_\_ 2016г. \_\_\_\_\_

## РЕФЕРАТ

Отчет содержит 13 страниц, 0 рисунка, 5 источников, 1 приложений.

МАНИПУЛЯТОР, OPENCV, ARDUINO, РОБОТ, КОМПЬЮТЕРНОЕ  
ЗРЕНИЕ.

Цель работы — разработка кухонного манипулятора с использованием  
компьютерного зрения.

Отчет по производственной практике написан при помощи системы  
компьютерной вёрстки L<sup>A</sup>T<sub>E</sub>X.

## Содержание

1	Введение . . . . .	6
2	Обзор манипулятора . . . . .	7
3	Обзор ARDUINO . . . . .	8
4	Обзор OPENCV . . . . .	9
5	Алгоритм работы программ . . . . .	10
5.1	Алгори работы поиска предметов . . . . .	10
5.2	Алгоритм управления манипулятором . . . . .	10
6	Испытания . . . . .	10
7	Заключение . . . . .	11
	Список используемых источников . . . . .	12
	Приложение А Компакт-диск . . . . .	13

## Введение

Целью данной работы является разработка кухонного манипулятора с использованием системы компьютерного зрения, собрать кухонный манипулятор по заранее спроектированным схемам, протестировать систему компьютерного зрения и кухонного манипулятора и отладить найденные ошибки.

## Обзор манипулятора

Манипулятор – совокупность пространственного рычажного механизма и системы приводов, осуществляющая под управлением программируемого автоматического устройства или человека-оператора действия (манипуляции), аналогичные действиям руки человека.

Промышленные роботы предназначены для замены человека при выполнении основных и вспомогательных технологических операций в процессе промышленного производства. При этом решается важная социальная задача – освобождения человека от работ, связанных с опасностями для здоровья или с тяжелым физическим трудом, а также от простых монотонных операций, не требующих высокой квалификации. Гибкие автоматизированные производства, создаваемые на базе промышленных роботов, позволяют решать задачи автоматизации на предприятиях с широкой номенклатурой продукции при мелкосерийном и штучном производстве. Промышленные роботы являются важными составными частями современного промышленного производства.

Манипулятор по принципу действия напоминает человеческую руку. В нём присутствуют поворотные соединения, которые обеспечивают наклон в плечевом соединении и сгибание в локте, механический захват, который позволит роботу хватать и перемещать предметы в разных направлениях.

Разрабатываемый в данной работе манипулятор будет иметь следующие параметры классификации:

- степень универсальности – специальный, предназначен для переноски небольших лёгких предметов;
- тип приводов – электрический;
- грузоподъемность – сверхлёгкий (до 1 кг);
- подвижность робота – стационарный;
- способ размещения – напольный;
- способ управления – программный и ручной.

## Обзор ARDUINO

Система управления манипулятором, как правило, имеет несколько уровней, каждый из которых может обслуживаться собственной микропроцессорной системой. Так, на уровне привода обеспечивается управление двигателем, осуществляющим движение одной или нескольких степеней подвижности. На следующем уровне системы управления манипулятором с помощью центрального процессора организуется координированная работа приводов манипулятора. При этом входной информацией является траектория, т. е. последовательность положений схвата манипулятора или связанного с ним объекта (инструмента, нагрузки).

Чтобы указать сервоприводу желаемое положение, по предназначенному для этого проводу необходимо посылать управляющий сигнал. Управляющий сигнал — импульсы постоянной частоты и переменной ширины.

То, какое положение должен занять сервопривод, зависит от длины импульсов. Когда сигнал поступает в управляющую схему, имеющийся в ней генератор импульсов производит свой импульс, длительность которого определяется через потенциометр. Другая часть схемы сравнивает длительность двух импульсов. Если длительность разная, включается электромотор. Направление вращения определяется тем, какой из импульсов короче. Если длины импульсов равны, электромотор останавливается.

Для управления сервоприводами кухонного манипулятора использовалась плата Arduino Uno R3, с встроенной библиотекой «Servo» в которой по умолчанию выставлены следующие значения длин импульса: 544 мкс — для  $0^\circ$  и 2400 мкс — для  $180^\circ$ .

Многие сервоприводы могут быть подключены к Arduino непосредственно. Для этого от них идёт шлейф из трёх проводов:

- красный — питание; подключается к контакту 5V или напрямую к источнику питания
- коричневый или чёрный — земля



- жёлтый или белый — сигнал; подключается к цифровому выходу Arduino.

Все сервоприводы манипулятора были подключены непосредственно к плате Arduino Uno R3, которая в свою очередь при получении команды от ПК, подавала сигнал вращения на нужный угол, нужного сервопривода. Библиотека Servo позволяет осуществлять программное управление сервоприводами. Для этого заводится переменная типа Servo. Управление осуществляется следующими функциями:

- `attach()` — присоединяет переменную к конкретному пину. Возможны два варианта синтаксиса для этой функции: `servo.attach(pin)` и `servo.attach(pin, min, max)`. При этом `pin` — номер пина, к которому присоединяют сервопривод, `min` и `max` — длины импульсов в микросекундах, отвечающих за углы поворота  $0^\circ$  и  $180^\circ$ . По умолчанию выставляются равными 544 мкс и 2400 мкс соответственно;
- `write()` — отдаёт команду сервоприводу принять некоторое значение параметра. Синтаксис следующий: `servo.write(angle)` , где `angle` — угол, на который должен повернуться сервопривод;
- `writeMicroseconds()` — отдаёт команду послать на сервопривод импульс определённой длины, является низкоуровневым аналогом предыдущей команды. Синтаксис следующий: `servo.writeMicroseconds(uS)` , где `uS` — длина импульса в микросекундах;
- `read()` — читает текущее значение угла, в котором находится сервопривод. Синтаксис следующий: `servo.read()`, возвращается целое значение от 0 до 180;
- `attached()` — проверка, была ли присоединена переменная к конкретному пину. Синтаксис следующий: `servo.attached()` , возвращается логическая истина, если переменная была присоединена к какому – либо пину, или ложь в обратном случае;
- `detach()` — производит действие, обратное действию `attach()` , то есть отсоединяет переменную от пина, к которому она была приписана. Синтаксис следующий: `servo.detach()`.

## Обзор OPENCV

Алгоритм работы программ

Алгоритм работы поиска предметов

Алгоритм управления манипулятором

Испытания

## Заключение

В данной работе была спроектирована и реализована база данных для прохождения теста Климова. Для разработки базы данных был выбран веб-интерфейс администрирования phpMyAdmin и язык программирования Python, а для разработки приложения использовался микрофреймворк Flask.

Был разработан дизайн для веб-страницы прохождения теста с помощью фреймворка bootstrap.

Также была написана программа для отображения результатов пройденных тестов с возможностью фильтрации по логину, группам и дате прохождения.

## Список используемых источников

- 1 MySQL. [Электронный ресурс] // ru.wikipedia.org:[сайт]. 2016. URL: <https://ru.wikipedia.org/wiki/MySQL>.
- 2 Python. [Электронный ресурс] // ru.wikipedia.org:[сайт]. 2016. URL: <https://ru.wikipedia.org/wiki/Python>.
- 3 Flask. [Электронный ресурс] // flask.pocoo.org:[сайт]. 2016. URL: <http://flask.pocoo.org/docs/0.11/>.
- 4 Bootstrap. [Электронный ресурс] // getbootstrap.com:[сайт]. 2016. URL: <https://getbootstrap.com/getting-started/>.
- 5 Тест Климова. [Электронный ресурс] // vsetesti.ru:[сайт]. 2016. URL: <http://vsetesti.ru/280/>.

Приложение А  
(Обязательное)  
Компакт-диск

Компакт-диск содержит:

- Электронную версию пояснительной записки в форматах \*.tex и \*.pdf;
- Актуальную версию программы для проведения теста Климова.