

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Московский институт электроники и математики имени А. Н. Тихонова

Лифановский Дмитрий Валентинович, БИВ225
Пыжов Илья Игоревич, БИВ225

Разработка системы управления 3D принтером

Курсовая работа
по направлению 09.03.01 Информатика и вычислительная техника
студента образовательной программы бакалавриата
«Информатика и вычислительная техника»

Студент _____ Лифановский Д.В.
подпись

Студент _____ Пыжов И.И.
подпись

Руководитель
ассистент МИЭМ НИУ ВШЭ
_____ Американов А.А.
подпись

Москва 2023 г.

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Московский институт электроники и математики имени А. Н. Тихонова

ЗАДАНИЕ
на курсовую работу бакалавра

студентов группы БИВ225 Лифановского Дмитрия Валентиновича
и Пыжова Ильи Игоревича

1. Тема работы

«Разработка системы управления 3D принтером»

2. Требования к работе

Необходимо разработать аппаратно-программный комплекс для дистанционного управления 3D принтером Picaso Designer X. Необходимо разработать систему переключения меню настроек (механический поворот ручки управления, нажатие на ручку управления).

Также для программы управления необходимо добавить возможность записывать видеозапись печати и возможность делать скриншоты. Сделать установщик программы. Также необходимо сделать манипулятор для перемещения камеры от области печати к области демонстрации меню.

- Аппаратная реализация на основе платы Arduino
- Программная реализация - язык программирования C, среда разработки Arduino IDE.
- Реализация программы для компьютера - язык программирования C#.

2.1. Требование к составу технических средств: 3d принтер Picaso Designer X; компьютер с минимальными системными требованиями: процессора с тактовой частотой 1ГГц или выше, 1Гб ОЗУ; мышь и клавиатура, монитор, наличие доступа в сеть Интернет; плата с микроконтроллером Arduino Micro/Uno и минимальными требованиями: флеш-память 32 Кб и ОЗУ 2Кб; веб-камера с USB интерфейсом; соленоид-толкатель на 20 Ньютонов; шаговый мотор и драйвер к нему; реле на 3 группы / самодельный транзисторный драйвер; источник внешнего питания постоянного тока на 5-6 вольт; набор алюминиевых профилей и ременная передача.

2.2. Требования к программной части: ОС Windows 7, 8, 10; Arduino IDE 1.8.X и новее.

2.3. Требования к функциональным характеристикам: Программа должна реализовывать дистанционное управление 3d принтером, а точнее: посредством разработанной программы и приложения удаленного рабочего стола можно будет управлять меню настроек принтера, осуществлять перемещение камеры от области печати к области демонстрации меню, создавать видеозаписи и скриншоты. Также отдельным пунктом является создание установочного файла программы для более удобного распространения программы. Приложение-программа позволяет осуществить связь между устройством-манипулятором на базе Arduino и непосредственно компьютером.

3. Содержание работы:

3.1. Введение;

3.2. Разработка структуры проекта;

3.3. Разработка манипулятора управления меню настроек;

3.4. Подключение камеры и разработка манипулятора для нее;

3.5. Написание приложения для компьютеров и разработка связи между созданным устройством и компьютером;

3.6. Тестирование;

3.7. Выводы.

4. Требования к программной документации:

4.1. Отчет;

4.2. Описание алгоритма решения задачи;

4.3. Описание кода программы;

4.4. Описание тестирования.

5. Сроки выполнения этапов работы

| Содержание этапа | Срок сдачи |
|--|---------------|
| Выбор темы курсовой работы | 28.11 2022 |
| Составление задания на курсовую работу | 10.01 2023 |
| Проект программы - первая версия ПО | 28.02 2023 |
| Проект документации - первая версия отчета | 31.03 2023 |

| | |
|--|---------------|
| Окончательная версия ПО | 15.05 2023 |
| Отчет и составление программной документации | 31.05 2023 |

Первый вариант КР предоставляется студентом в срок до «28» февраля 2023г.

Итоговый вариант КР предоставляется студентом в срок до «15» мая 2023г.

Задание выдано «28» ноября 2022 г. _____А.А. Американов

Задание принято к «28» ноября 2022 г. _____Д.В. Лифановский
исполнению

_____И.И. Пыжов

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Московский институт электроники и математики имени А. Н. Тихонова

**График выполнения
курсовой работы бакалавра**

студентов группы БИВ225 Лифановского Дмитрия Валентиновича
и Пыжова Ильи Игоревича

Тема работы

«Разработка системы управления 3d принтером»

Дата согласования первого «__» _____ 2023г. _____ А.А. Американов
варианта КР

Дата согласования «__» _____ 2023 г. _____ А.А. Американов
итогового варианта КР

АННОТАЦИЯ

Объектом данной курсовой работы является разработка аппаратно-программного комплекса для дистанционного управления 3d принтером Picaso Designer X. В ходе работы разработаны система переключения меню настроек (механический поворот ручки управления, нажатие на ручку управления), манипулятор для перемещения камеры от области печати к области демонстрации меню. Также в программу управления добавлена возможность записывать видеозапись печати и возможность делать скриншоты. Был сделан установщик программы.

Курсовая работа выполнена на 52 листах с использованием 23 источников, содержит 21 рисунков.

ОГЛАВЛЕНИЕ

| | |
|--|----|
| ВЕДЕНИЕ | 7 |
| 3D МОДЕЛЬ АППАРАТНО-ПРОГРАММНОГО КОМПЛЕКСА..... | 9 |
| ТОЛКАТЕЛЬ-СОЛЕНОИД | 10 |
| Характеристики..... | 10 |
| Программирование платы для управления соленоидом-толкателем..... | 11 |
| ШАГОВЫЙ ДВИГАТЕЛЬ | 12 |
| Характеристики шагового двигателя 28BYJ-48: | 12 |
| Технические характеристики драйвера ULN2003 для шагового двигателя 28BYJ-48..... | 14 |
| Режим работы двигателя | 14 |
| Программирование платы для управления шаговым двигателем | 15 |
| ПРИЛОЖЕНИЕ..... | 18 |
| ТЕСТИРОВАНИЕ | 22 |
| ВЫВОД | 23 |
| СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ | 24 |
| ПРИЛОЖЕНИЕ (ЛИСТИНГ КОДА) | 26 |
| 1..... | 26 |
| 2.1..... | 30 |
| 2.2..... | 37 |
| 3.1..... | 47 |
| 3.2..... | 48 |
| 4..... | 51 |

ВЕДЕНИЕ

Современный мир постоянно пополняется новыми технологиями. Одними из которых являются аддитивные технологии и сопутствующие им 3D принтеры, набирающие популярность. Широкое распространение 3D принтеры приобрели за счет доступной цены и простоты использования, а также обширного применения напечатанных деталей в бытовых и производственных целях. Но вместе с ростом популярности стали заметны и многие недостатки домашних 3D принтеров.

Самый главный недостаток любого 3D принтера - время печати. Печать больших и сложных деталей занимает очень много времени. Также на качество работы принтера влияют и окружающие факторы. Если на производстве создаются специальные условия для печати, то в домашних условиях их достичь очень сложно. Качество печати может ухудшиться за счет лишних вибраций при ходьбе и наличии сквозняков, которые сложно контролировать в домашних условиях. А в некоторых случаях деталь может сдвинуться и не напечататься до конца или даже сломаться и повредить элементы самого принтера. Поэтому за процессом печати необходимо постоянное наблюдение, чтобы в экстренных случаях приостановить печать или начать ее снова. И если на производстве организовать контроль внешних факторов и самой печати не составит труда, то в бытовых условиях организовать наблюдение достаточно сложно, так как это длительный процесс. Но пользователь принтера не может постоянно находиться рядом с ним и контролировать печать, иначе бытовые и развлекательные цели этой печати смещаются в производственные.

Пример использования 3D принтера можно увидеть в МИЭМ НИУ ВШЭ в лабораториях 3D-визуализации и компьютерной графики и систем автоматического проектирования.

Для более эффективной работы принтера печать на нем можно вести круглосуточно. Но персонал, ответственный за работу принтера не может постоянно находиться возле него, особенно в ночные и нерабочие часы института. Данная курсовая работа направлена на решение этой проблемы – создание аппаратно-программного комплекса для дистанционного управления и контроля работы 3D принтера.

Для реализации поставленной цели необходимо выполнить следующие задачи:

- Изучение ряда необходимых элементов (шаговые двигатели, соленоиды толкатели, модули камеры и советующие им драйвера) и выбор подходящих для реализации работы;

- Замер габаритов 3D принтера и размеров всех выбранных элементов. Последующее создание их 3D моделей в CAD для определения общей картины габаритов деталей, необходимых для создания комплекса;
- Конструирование в CAD 3D модели системы крепления, позволяющую корректно осуществлять дистанционное управление принтером, свободно перемещаться подвижным элементам, не затрудняющую доступ к меню принтера, области печати и защитной крышке;
- Сборка итоговой системы крепления при помощи алюминиевых профилей и фигурных профилей, их соединителей и других элементов;
- Программирование платы Arduino для реализации работы двух шаговых двигателей и толкателя;
- Создание приложения для реализации управления собранной установкой на языке программирования C# и его установщика на персональный компьютер;
- Тестирование разработанного аппаратно-программного комплекса.

3D МОДЕЛЬ АППАРАТНО-ПРОГРАММНОГО КОМПЛЕКСА

В T-Flex CAD и Autodesk Inventor была создана 3D модель конструкции, реализующей дистанционное управления 3D принтером Picaso Designer X. Разработанная конструкция включает в себя профильным каркас, крепление-каретку для вертикального движения камеры от области печати к области демонстрации меню и обратно, также на вертикальный фигурный профиль под углом(перпендикулярно плоскости экрана меню настроек) закреплены две пластины-держателя для толкателя-соленоида для управления нажатием на кнопку и одного из шаговых двигателей для переключения настроек – кручения кнопки. Кручение осуществляется с помощью ременной передачи. Также в верхней части расположен второй шаговый двигатель для реализации наматывания троса, на котором подвешена каретка для камеры, и соответствующего передвижения камеры. Также на каркасе закреплены драйвера к элементам и сама плата Arduino, а также провода к ним.

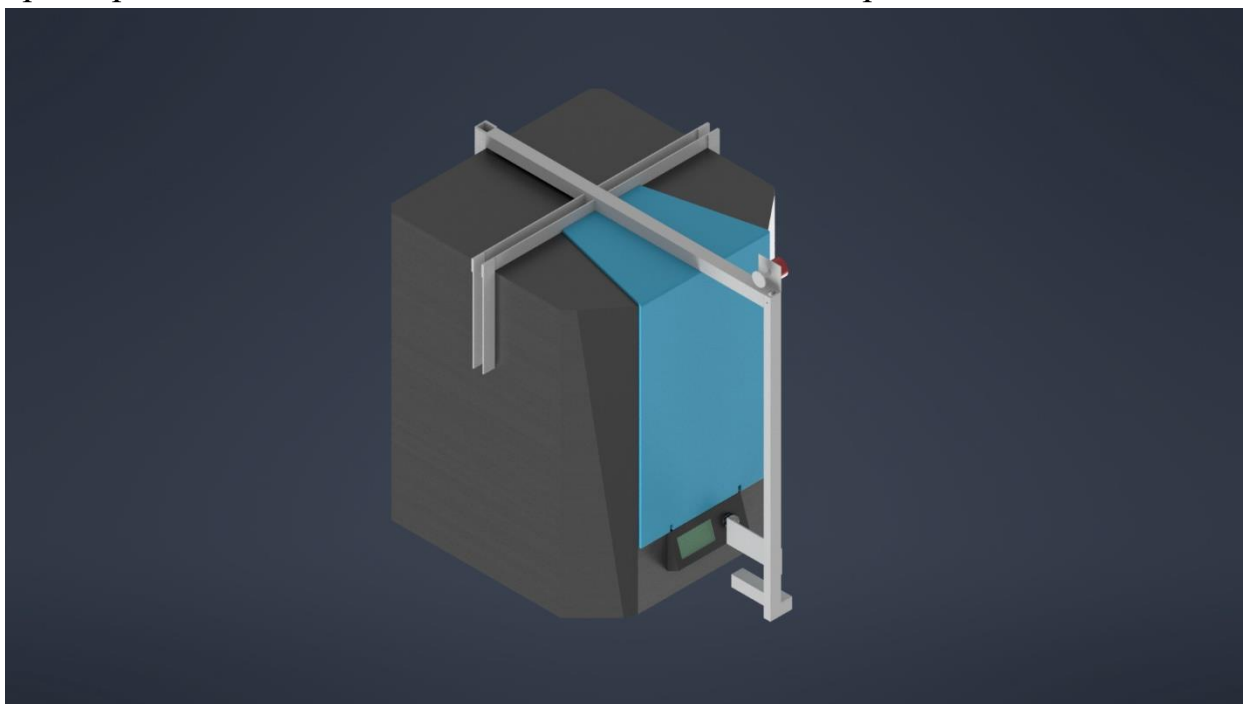


Рис.1 3D модель сборки вместе с 3D принтером

ТОЛКАТЕЛЬ-СОЛЕНОИД

Электрохимический соленоид — это электромагнитное устройство, которое преобразует электрическую энергию в механическое толкающее движение по поступательно-возвратному принципу.

Толкатель осуществляет нажатие на кнопку меню настроек. Установлен перпендикулярно кнопке на расстоянии нажатия. Для подключения используется транзистор irf3205 для самодельного драйвера для толкателя.



Рис.2 Соленоид JF-0826B (20 Н, 12 В)

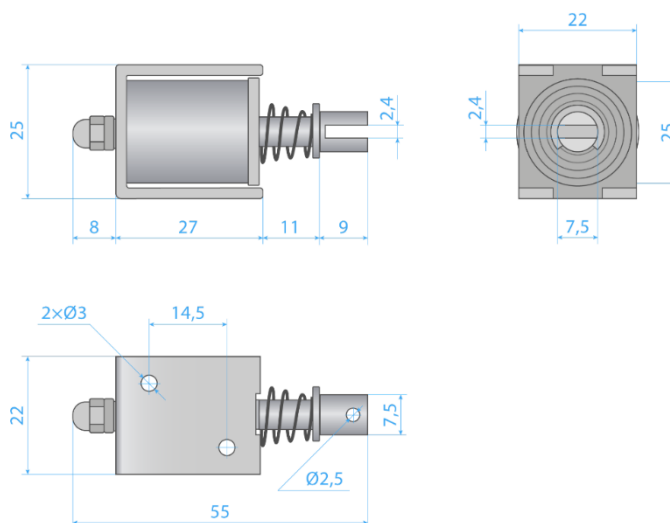


Рис.3 Габаритный чертёж соленоида

Характеристики

- Модель: Соленоид JF-0826B (20 Н, 12 В)
- Усилие: 20 Н
- Напряжение питания: 12 В
- Потребляемый ток: до 2 А
- Сопротивление обмотки: 6 Ом
- Ход штока: 10 мм
- Диаметр штока: 7,5 мм

Программирование платы для управления соленоидом-толкателем

```
#define Push_pin 12
```

Рис.4 фрагмент кода 1

В этой строке определяется пин (Push_pin), к которому подключен соленоид-толкатель.

Далее в представленном фрагменте кода обрабатывается команда "push" (нажатие на ручку управления). Когда команда получена, устанавливается флаг push_check для обозначения активного состояния соленоид-толкателя. Затем происходит проверка временного интервала push_interval, в течение которого соленоид-толкатель находится в активном состоянии. Если интервал еще не истек, на пин Push_pin подается высокий уровень (HIGH), что активирует соленоид-толкатель. После истечения интервала уровень на пине Push_pin устанавливается в низкий (LOW), соленоид-толкатель деактивируется, и флаг push_check сбрасывается.

```
if(str == "push") //нажатие на ручку управления
{
    if (!push_check) {
        push_check = true;
        push_current = millis();
    }
    if (millis() - push_current <= push_interval) {
        digitalWrite(Push_pin, HIGH);
    } else {
        digitalWrite(Push_pin, LOW);
        push_check = false;
        str = "";
    }
}
```

Рис.5 фрагмент кода 2

ШАГОВЫЙ ДВИГАТЕЛЬ

Шаговые двигатели преобразуют электрические импульсы в перемещение вала на определенный угол.

Двигатель подключается к плате Arduino с драйвером на микросхеме ULN2003.

В данной работе шаговые двигатели используются для 1. реализации кручения кнопки переключения меню настроек, 2. реализации кручения шпильки для передвижения вдоль него камеры.

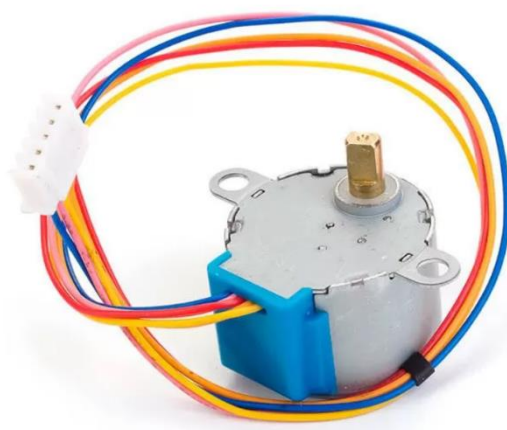


Рис. 6 Шаговый двигатель 28BYJ-48

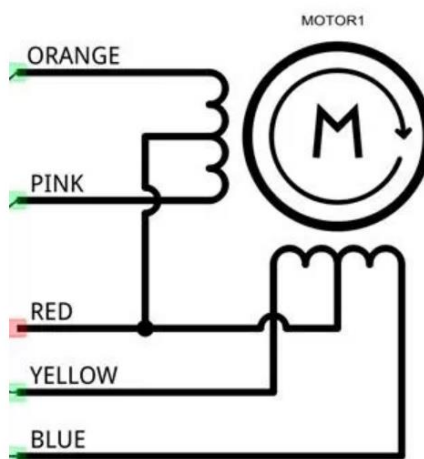


Рис. 7 Принципиальная схема шагового двигателя 28BYJ-48.

Характеристики шагового двигателя 28BYJ-48:

- Напряжение питания – 5В;
- Число фаз – 4;
- Коэффициент редукции – 1/63.68395;
- Количество шагов ротора – 64;

- Номинальная скорость вращения – 15 оборот/мин;
- Крутящий момент – 450 г*см;
- Размеры (диаметр, высота) – 25х18 мм;
- Вес – 40 граммов.

Драйвер двигателя состоит из 7 пар транзисторов Дарлингтона и является усилителем. Выводы IN1 – IN7 предназначены для подключения к микроконтроллеру, GND и VCC – для питания шагового двигателя.

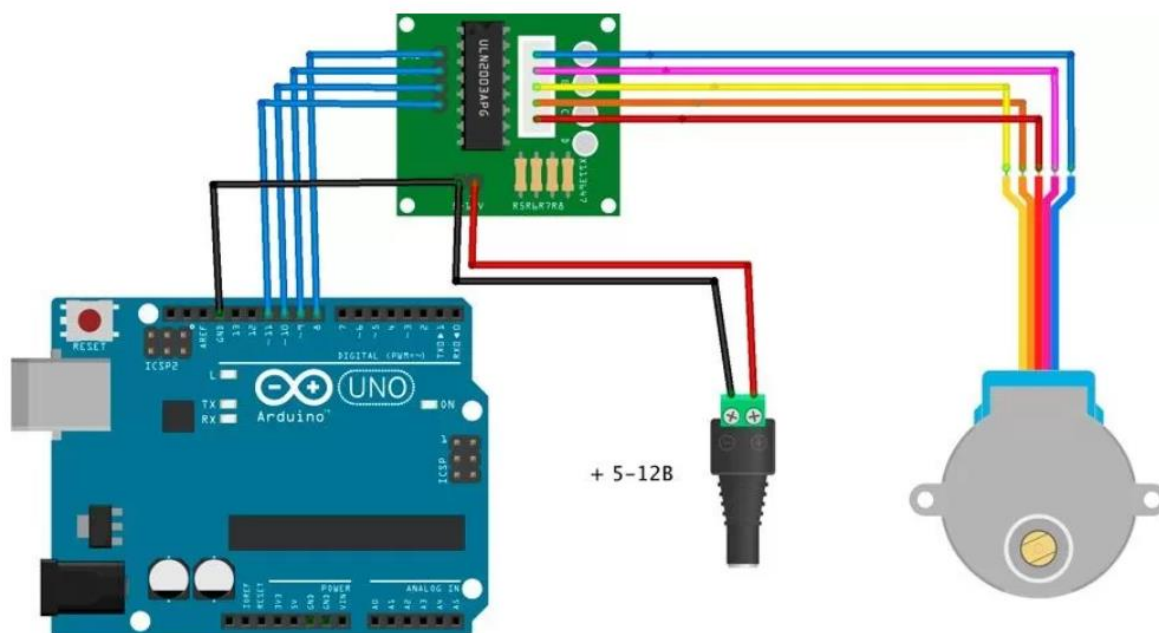


Рис. 8 Схема подключения драйвера к плате Arduino

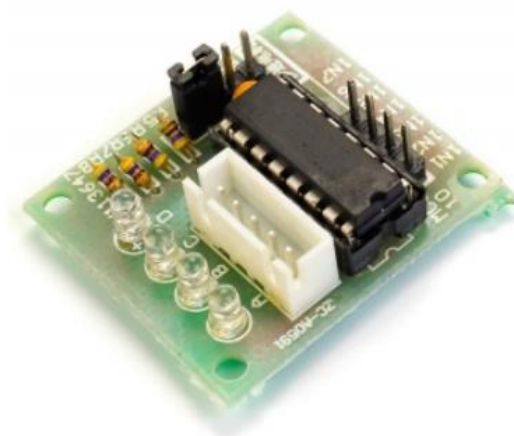


Рис. 9 Драйвер ULN2003 для шагового двигателя 28BYJ-48

Шаговый двигатель не подключается непосредственно к контроллеру по причине возможного увеличения тока на обмотках до 160 мА. Для управления мотором используется специальная микросхема – драйвер. Он представляет собой компактную микросхему с 7 контактами для микроконтроллера, 5

контактами для двигателя и 2 контактами для подключения питания. У драйвера ULN2003 для шагового двигателя 28BYJ-48 есть перемычка, с помощью которой разрывается цепь питания. В работе с этим драйвером двигатель дает небольшой крутящий момент.

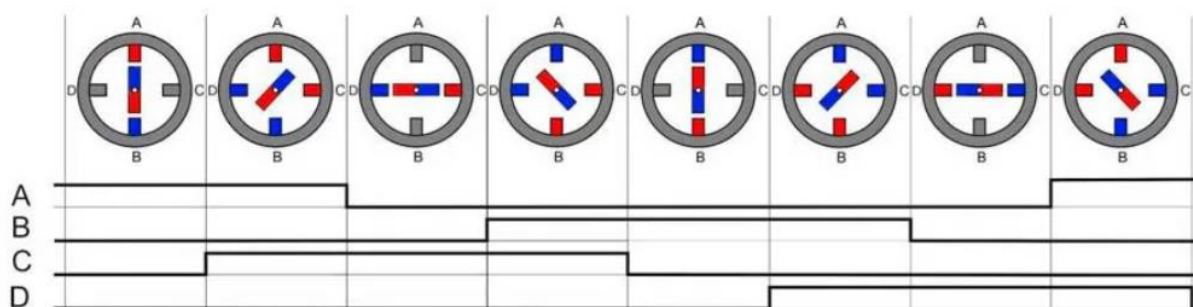
Технические характеристики драйвера ULN2003 для шагового двигателя 28BYJ-48

- размеры платы – 32х35 мм;
- межосевые размеры платы – 30х27 мм;
- напряжение питания – 5-12В;
- управляющее напряжение – 5В;
- вес – 8 г;
- номинальный ток коллектора каждого ключа – 0,5А;
- наличие светодиодной индикации фаз;
- вход адаптирован к различным видам логики.

Режим работы двигателя

Шаговый двигатель имеет 2 основных режима работы: стандартный на 2048 шагов на полный оборот и полушаговый. Второй режим используется для повышения точности работы ШД.

Полушаговый режим – 8 ступеней импульсов на 1 шаг, при нем двигатель делает полный оборот за 4096 шагов.



| Контакт | Такты | | | | | | | |
|----------|-------|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| A-orange | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| B-yellow | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| C-pink | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| D-blue | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

Рис. 10 Схема работы полушагового режима ШД

Программирование платы для управления шаговым двигателем

```
#define Type_of_working 8
#define INA 11
#define INB 9
#define INC 10
#define IND 8
```

Рис. 11 Настройка пинов

Эти определения задают пины микроконтроллера, к которым подключены шаговые двигатели. INA, INB, INC, и IND определяют пины для основного шагового двигателя (для ручки управления), а INA_cam, INB_cam, INC_cam, и IND_cam - для двигателя, управляющего позицией камеры.

```
AccelStepper stepper_joy(Type_of_working, INA, INB, INC, IND);
AccelStepper stepper_cam(Type_of_working, INA_cam, INB_cam, INC_cam, IND_cam)
```

Рис. 12 Создание экземпляров AccelStepper

Здесь создаются два экземпляра класса AccelStepper - stepper_joy и stepper_cam. Конструктор AccelStepper принимает в качестве параметров тип работы двигателя (Type_of_working) и пины управления (INA, INB, INC, IND для stepper_joy и INA_cam, INB_cam, INC_cam, IND_cam для stepper_cam).

```
stepper_joy.setMaxSpeed(1000);
stepper_joy.setAcceleration(500);
stepper_cam.setMaxSpeed(1000);
stepper_cam.setAcceleration(500);
```

Рис. 13 Настройка параметров двигателей

Здесь устанавливаются максимальная скорость (setMaxSpeed()) и ускорение (setAcceleration()) для обоих двигателей. В данном случае максимальная скорость установлена на 1000 шагов в секунду, а ускорение - на 500 шагов в секунду за секунду.


```

if(str == "turn left")
{
    //...
    stepper_joy.move(Step_joy);
    //...
    stepper_joy.run();
    //...
    if (stepper_joy.distanceToGo() == 0) {
        //...
        stepper_joy.stop();
        //...
    }
}

```

Рис. 14 Управление двигателем для поворота влево

Если получена команда "turn left", то двигатель stepper_joy будет перемещаться влево. Сначала вызывается метод move(), чтобы задать расстояние (в данном случае Step_joy шагов), на которое нужно переместить двигатель. Затем вызывается метод run(), который активирует двигатель и выполняет перемещение. Проверка stepper_joy.distanceToGo() == 0 позволяет определить, достиг ли двигатель заданной позиции, и если достиг, то вызывается метод stop(), чтобы остановить двигатель.

```

if(str == "turn right")
{
    //...
    stepper_joy.move(-Step_joy);
    //...
    stepper_joy.run();
    //...
    if (stepper_joy.distanceToGo() == 0) {
        //...
        stepper_joy.stop();
        //...
    }
}

```

Рис. 15 Управление двигателем для поворота вправо

Аналогично предыдущему пункту, если получена команда "turn right", то двигатель stepper_joy будет перемещаться вправо. Здесь используется отрицательное значение -Step_joy, чтобы указать направление вправо.

```

if (str=="cam menu")
{
    //...
    stepper_cam.moveTo(Step_cam);
    //...
    stepper_cam.run();
    //...
    if (stepper_cam.distanceToGo() == 0) {
        //...
        stepper_cam.stop();
        //...
    }
}

```

Рис. 16 Управление двигателем для перемещения камеры к меню

Если получена команда "cam menu", то двигатель stepper_cam будет перемещаться к меню. Метод moveTo() задает позицию, до которой нужно переместить двигатель (здесь используется значение Step_cam). Затем метод run() активирует двигатель, и метод distanceToGo() проверяет, достиг ли двигатель указанной позиции. Если достиг, вызывается метод stop(), чтобы остановить двигатель.

```

if (str=="cam workspace")
{
    //...
    stepper_cam.move(-Step_cam);
    //...
    stepper_cam.run();
    //...
    if (stepper_cam.distanceToGo() == 0) {
        //...
        stepper_cam.stop();
        //...
    }
}

```

Рис. 17 Управление двигателем для перемещения камеры к рабочей области

Аналогично предыдущему пункту, если получена команда "cam workspace", то двигатель stepper_cam будет перемещаться к рабочей области.

Здесь используется отрицательное значение `-Step_cam`, чтобы указать направление к рабочей области.

В каждом из участков кода присутствуют дополнительные действия, такие как обнуление пинов или изменение состояний флагов, которые выполняются после остановки двигателей.

ПРИЛОЖЕНИЕ

Было реализовано приложение на языке C# с функциями управления камерой и кручением, нажатием кнопки управления меню, также была добавлена возможность видеозаписи печати и возможность делать скриншоты. Был сделан установщик программы.

Для начала происходит подключение необходимых библиотек для работы с изображением при помощи диспетчера пакетов NuGet. А именно это библиотеки EMGU.CV, DirectShowLib и остальные производные от них.

В данном коде при загрузке формы происходит инициализация комбобокса `comboBox_Cam`, который содержит список доступных камер. Каждая доступная камера добавляется в комбобокс. Затем выбирается первая камера в списке по умолчанию.

Также инициализируется комбобокс `comboBox1`, содержащий список доступных виртуальных портов. Аналогично, первый порт в списке выбирается по умолчанию.

Для захвата изображения с камеры создается объект `capture` типа `VideoCapture`. При этом устанавливается обработчик события `ImageGrabbed`, который будет вызываться при получении нового кадра с камеры.

```

private void Form1_Load(object sender, EventArgs e)
{
    // Инициализация комбобокса с доступными камерами
    webCams = DsDevice.GetDevicesOfCat(FilterCategory.VideoInputDevice);
    foreach (var camera in webCams)
    {
        comboBox_Cam.Items.Add(camera.Name);
    }
    if (comboBox_Cam.Items.Count > 0)
        comboBox_Cam.SelectedIndex = 0;

    // Инициализация комбобокса с доступными портами
    comboBox1.Items.AddRange(Ports_List);
    if (comboBox1.Items.Count > 0)
        comboBox1.SelectedIndex = 0;

    // Установка параметров для захвата изображения с камеры
    capture = new VideoCapture(selectedCameraId);
    capture.ImageGrabbed += Capture_ImageGrabbed;
}

```

Рис. 18 Инициализация формы и элементов управления при загрузке приложения

```

private void Capture_ImageGrabbed(object sender, EventArgs e)
{
    Mat frame = new Mat();
    capture.Retrieve(frame);

    if (frame != null && stream_on)
    {
        pictureBox1.Image = frame.Bitmap;
    }
}

```

Рис. 19 Захват изображения с камеры и отображение на PictureBox

Метод `Capture_ImageGrabbed` вызывается при событии захвата нового изображения с камеры. Внутри метода создается объект `Mat` для хранения кадра изображения. Затем используется метод `Retrieve` объекта `capture`, чтобы получить текущий кадр и сохранить его в объект `frame`.

Если получен кадр и флаг stream_on установлен в true, то изображение конвертируется в формат Bitmap и отображается на PictureBox с помощью свойства Image.

```
private void button1_Click(object sender, EventArgs e)
{
    if (Current_Port.IsOpen)
    {
        Current_Port.Write("turn left");
        TEXT.Text = "Джойстик управления: поворот влево";
    }
    else
    {
        TEXT.Text = "Выполнить действие нельзя! Выберите необходимый порт для подключения";
        MessageBox.Show("Необходимо выбрать порт", "Внимание!", MessageBoxButtons.OK);
    }
}
```

Рис. 20 Функция button1_Click - обработчик события нажатия на кнопку "Turn left" (поворот влево)

Этот код предназначен для отправки команды "turn left" на текущий порт и обновления пользовательского интерфейса с помощью текстового поля TEXT. Если порт открыт, команда "turn left" отправляется, и текстовое поле обновляется соответствующим сообщением. В противном случае, когда порт не открыт, текстовое поле устанавливается на сообщение об ошибке, а пользователю выводится предупреждающее окно с информацией о необходимости выбора порта для подключения.

Аналогично реализуются команды "Turn right" и "Push".

Также предусмотрены случаи ошибок при работе приложения:

- «Ошибка вывода изображения»
- «Нет подходящих камер» (программа не распознает камеры, они не отображаются в выпадающем списке)
- «Необходимо выбрать камеру»
- «Трансляция изображения уже ведется» (при повторном нажатии кнопки «stream»)
- «Выполнить действие нельзя! Выберите необходимый порт для подключения» (при невыбранном порте подключения)
- «Необходимо выбрать порт»

Приложение выполняет следующие функции:

- Инициализирует форму и элементы управления при загрузке приложения.
- Получает список доступных видеокамер и заполняет комбобокс с их именами.
- Захватывает изображение с выбранной камеры и отображает его на PictureBox.
- Позволяет пользователю выбрать порт и управлять 3D принтером с помощью кнопок "Turn left", "Turn right", "Push", "To menu" и "To Workspace".
- Реализует функции создания скриншота с изображения с камеры и записи таймлапса.
- Отображает информацию о состоянии системы и обрабатывает возможные исключения при работе с камерой и портами.

Был реализован интерфейс к приложению:



Рис. 21 общий вид интерфейса приложения

В окне приложения находятся два выпадающих меню для выбора порта подключения (1) и камеры (5), кнопки управления шаговым двигателем и толкателем, реализующих управление кнопкой меню настроек: поворот влево (2), поворот вправо (3), нажатие (4). Также кнопка подключения трансляции изображения с камеры (6). Кнопками (7) и (8) можно менять положение камеры – установить ее на уровне меню и на уровне печати соответственно. Также функция скриншота вызывается кнопкой (9), кнопки (10) и (11) отвечают за начало и конец записи трансляции соответственно.

ТЕСТИРОВАНИЕ

В результате тестирования была проверена работоспособность аппаратно-программного комплекса. На практике было доказано, что разработанная конструкция полностью справляется с поставленной задачей и реализует дистанционное управление 3D принтером Picaso Designer X. Собранный механизм осуществляет передвижение камеры вдоль вертикального фигурного профиля, который составляет часть общего каркаса, а также без затруднений и ошибок реализует управление меню настроек принтера. Доступны функции скриншота и записи трансляции печати.

ВЫВОД

В ходе работы над проектом был разработан, собран и запрограммирован аппаратно-программный комплекс для дистанционного управления 3d принтером Picaso Designer X. Были применены навыки работы в CAD, написания кода на языках программирования Arduino C и C#.

Была запрограммирована плата Arduino с подключенными к ней двумя шаговыми двигателями 28BYJ-48 (вместе с драйверами к ним на микросхеме ULN2003) и соленоидом-толкателем JF-0826B.

Также написано приложение с функциями управления подключенных к плате элементов, осуществляющих движение камеры и нажатие, поворот кнопки меню настроек. Реализован интерфейс приложения и его установщик.

Комплекс успешно протестирован и способен безошибочно выполнять поставленную задачу – осуществлять дистанционное управление принтером ночью или в любое другое нерабочее время в лаборатории МИЭМ НИУ ВШЭ.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. "WPF Overview" [Электронный ресурс]. – Режим доступа: <https://learn.microsoft.com/ru-ru/dotnet/desktop/wpf/overview/?view=netdesktop-7.0>.
2. "Arduino библиотека AccelStepper для шагового двигателя" [Электронный ресурс]. – Режим доступа: <https://hobbytech.com.ua/arduino>
3. "Creating a Virtual Port" [Электронный ресурс]. – Режим доступа: <https://learn.microsoft.com/ru-ru/windows-hardware/drivers/network/creating-a-virtual-port>.
4. "Интерфейс передачи данных UART" [Электронный ресурс]. – Режим доступа: <https://3d-diy.ru/wiki/arduino-moduli/interfeys-peredachi-dannykh-uart/>.
5. "Create a Windows Forms app with Visual Studio" [Электронный ресурс]. – Режим доступа: <https://learn.microsoft.com/ru-ru/dotnet/desktop/winforms/get-started/create-app-visual-studio?view=netdesktop-7.0>.
6. "Windows Forms в .NET" [Электронный ресурс]. – Режим доступа: <https://learn.microsoft.com/ru-ru/dotnet/desktop/winforms/overview/?view=netdesktop-7.0>.
7. "Windows Forms в .NET Framework" [Электронный ресурс]. – Режим доступа: <https://learn.microsoft.com/ru-ru/dotnet/desktop/winforms/windows-forms-overview?view=netframeworkdesktop-4.8>.
8. "How to: Add Controls to Windows Forms" [Электронный ресурс]. – Режим доступа: <https://learn.microsoft.com/ru-ru/dotnet/desktop/winforms/controls/how-to-add-controls-to-windows-forms?view=netframeworkdesktop-4.8>.
9. "How to: Position Controls on Windows Forms" [Электронный ресурс]. – Режим доступа: <https://learn.microsoft.com/ru-ru/dotnet/desktop/winforms/controls/how-to-position-controls-on-windows-forms?view=netframeworkdesktop-4.8>.
10. "Overview of the .NET Framework" [Электронный ресурс]. – Режим доступа: <https://learn.microsoft.com/ru-ru/dotnet/framework/get-started/overview>.
11. Драйвер ULN2003 для шагового двигателя 28BYJ-48 [Электронный ресурс]. – Режим доступа: <https://3d-diy.ru/product/drayver-uln2003-dvigatelya-28byj-48>.

12. "Stepper Motor 28BYJ-48" на Wiki 3D-DIY [Электронный ресурс]. – Режим доступа: <https://3d-diy.ru/wiki/arduino-mechanics/stepper-motor-28BYJ-48/>.
13. "Solenoid JF-0826B" на iarduino.ru [Электронный ресурс]. – Режим доступа: <https://iarduino.ru/shop/Mehanika/solenoid-jf-0826b.html>.
14. "Arduino и Wi-Fi" на vprautinu.com [Электронный ресурс]. – Режим доступа: <https://vprautinu.com/wifi/arduino>.
15. "ESP8266 WiFi Module" на wiki.amperka.ru [Электронный ресурс]. – Режим доступа: <http://wiki.amperka.ru/продукты:esp8266-wifi-module>.
16. "Подключение камеры OV7670 к Arduino" на voltiq.ru [Электронный ресурс]. – Режим доступа: <https://voltiq.ru/camera-ov7670-connecting-to-arduino/>.
17. "Введение в Arduino: от Hello World до Интернета вещей" на habr.com [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/articles/406559/>.
18. "АлексГайвер" на alexgyver.ru [Электронный ресурс]. – Режим доступа: <https://alexgyver.ru/>.
19. "Arduino модуль видеокamеры" на arduino-diy.com [Электронный ресурс]. – Режим доступа: <https://arduino-diy.com/arduino-modul-videokamera/>.
20. "Двигатель на 3D-принтере" на vektorus.ru [Электронный ресурс]. – Режим доступа: <https://vektorus.ru/blog/dvigatel-na-3d-printere.html>.
21. YouTube video: "_OY-SfWitck" [Видеозапись]. – Режим доступа: https://www.youtube.com/watch?v=_OY-SfWitck&t=2540s.
22. Подбельский, В. В. Программирование. Базовый курс C# : учебник для среднего профессионального образования. Москва: Издательство Юрайт, 2023. 369 с.
23. "WPF. Windows Presentation Foundation в .NET 4.5 с примерами на C# 5.0 для профессионалов" Переводчик: Артеменко Ю. Н. Издательство: Вильямс, 2013 г.

ПРИЛОЖЕНИЕ (ЛИСТИНГ КОДА)

1

```
#include <EEPROM.h>
#include <AccelStepper.h>

#define Type_of_working 8
#define INA 11
#define INB 9
#define INC 10
#define IND 8

#define INA_cam 4
#define INB_cam 5
#define INC_cam 6
#define IND_cam 7

#define Angle 360/24 //один шаг поворота ручки управления, 24 - количество шагов на 1
//полный оборот, + необходимо учесть различие радиусов двух валов - двигателя и самой
//ручки

#define Push_pin 12

String str;

unsigned long push_current = 0;
const long push_interval = 200;
bool push_check = false;
bool left_check = false;
bool right_check = false;
bool menu_check = false;
bool work_check = false;
bool cam_menu; //true - если камера у меню, иначе - false

int Step_joy = 4096;
int Step_cam = 4096*10;

int pos = 0;

AccelStepper stepper_joy(Type_of_working, INA, INB, INC, IND);
AccelStepper stepper_cam(Type_of_working, INA_cam, INB_cam, INC_cam, IND_cam);

void setup()
{
  Serial.begin(115200);
```

```

Serial.setTimeout(50);

pinMode(Push_pin, OUTPUT);

stepper_joy.setMaxSpeed(1000); //максимальная скорость в шагах в секунду
stepper_joy.setAcceleration(500); //ускорение в шагах в секунду за секунду
stepper_cam.setMaxSpeed(1000); //максимальная скорость в шагах в секунду
stepper_cam.setAcceleration(500); //ускорение в шагах в секунду за секунду
}

void loop()
{

if(Serial.available())
{
  str = Serial.readString();
}

if(str == "turn left") //поворот ручки управления влево
{
  if (!left_check){
    left_check = true;
    stepper_joy.move(Step_joy);
  }
  stepper_joy.run();
  if (stepper_joy.distanceToGo() == 0) {
    stepper_joy.stop();
    stepper_joy.move(Step_joy);
    left_check = false;
    str = "";
    digitalWrite(8, 0);
    digitalWrite(9, 0);
    digitalWrite(10, 0);
    digitalWrite(11, 0);
  }
}

if(str == "turn right") //поворот ручки управления вправо
{
  if (!right_check){
    right_check = true;
    stepper_joy.move(-Step_joy);
  }
  stepper_joy.run();
}

```

```

if (stepper_joy.distanceToGo() == 0) {
    stepper_joy.stop();
    stepper_joy.move(-Step_joy);
    right_check = false;
    str = "";
    digitalWrite(8, 0);
    digitalWrite(9, 0);
    digitalWrite(10, 0);
    digitalWrite(11, 0);
}
}

if(str == "push") //нажатие на ручку управления
{
    if (!push_check) {
        push_check = true;
        push_current = millis();
    }
    if (millis() - push_current <= push_interval) {
        digitalWrite(Push_pin, HIGH);
    } else {
        digitalWrite(Push_pin, LOW);
        push_check = false;
        str = "";
    }
}

if (str=="cam menu") //перемещаем камеру к меню
{
    if (!menu_check){
        EEPROM.get(0, cam_menu); //впервые она определится при установки системы и ее
настройке
        if (cam_menu) { //если камера и так уже у меню
            str = "";
            digitalWrite(8, 0);
            digitalWrite(9, 0);
            digitalWrite(10, 0);
            digitalWrite(11, 0);
        } else {
            cam_menu = true;
            EEPROM.put(0, cam_menu);
            menu_check = true;
            stepper_cam.moveTo(Step_cam);
        }
    }
}

```

```

stepper_cam.run();
if (stepper_cam.distanceToGo() == 0) {
  stepper_cam.stop();
  stepper_cam.move(Step_cam);
  menu_check = false;
  str = "";
  digitalWrite(8, 0);
  digitalWrite(9, 0);
  digitalWrite(10, 0);
  digitalWrite(11, 0);
}
}

if (str=="cam workspace")
{
  if (!work_check){
    EEPROM.get(0, cam_menu);
    if (!cam_menu) { //если камера находится у области печати
      str = "";
      digitalWrite(8, 0);
      digitalWrite(9, 0);
      digitalWrite(10, 0);
      digitalWrite(11, 0);
    } else {
      cam_menu = false;
      EEPROM.put(0, cam_menu);
      work_check = true;
      stepper_cam.move(-Step_cam);
    }
  }
  stepper_cam.run();
  if (stepper_cam.distanceToGo() == 0) {
    stepper_cam.stop();
    stepper_cam.move(-Step_cam);
    work_check = false;
    str = "";
    digitalWrite(8, 0);
    digitalWrite(9, 0);
    digitalWrite(10, 0);
    digitalWrite(11, 0);
  }
}
}

```

2.1

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Input;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using Emgu;
using Emgu.CV;
using Emgu.CV.Util;
using Emgu.CV.Structure;
using Emgu.Util;
using DirectShowLib;
using System.Windows.Forms;
using System.IO.Ports;

namespace _3D_Printer___Main_version
{
    public partial class Form1 : Form
    {
        SerialPort Current_Port = new SerialPort();
        string[] Ports_List = SerialPort.GetPortNames();
        bool stream_on = true; //для включения трансляции только 1 раз

        public Form1()
        {
            InitializeComponent();

            private VideoCapture capture = null; //в это поле хранятся захватываемые данные с
камеры
            private DsDevice[] webCams = null; //для хранения списка всех жоступных видео
камер
            private int selectedCameraId = 0; //для управления этими камерами, по умолчанию id=0

            private void Form1_Load(object sender, EventArgs e)
            {
                TEXT.Text = "Состояние системы";
                foreach (string Current_Port in Ports_List)
```

```

    {
        COM.Items.Add(Current_Port);
    }
    webCams =
DsDevice.GetDevicesOfCat(DirectShowLib.FilterCategory.VideoInputDevice); //получили
список всех доступных видео камер с определенными параметрами (videp input devises)

    for (int i = 0; i < webCams.Length; i++) //заполняем комбобокс с камерами
    {
        comboBox_Cam.Items.Add(webCams[i].Name);
    }

}

//КАМЕРА
private void comboBox_Cam_SelectedIndexChanged(object sender, EventArgs e)
//обработка события в комбобоксе с именам камер - определение и сохранения выбора
{
    selectedCameraId = comboBox_Cam.SelectedIndex;
}

private void Capture_ImageGrabbed(object sender, EventArgs e) //Cam - обработка
избражения - end
{
    try
    {
        Mat m = new Mat();
        capture.Retrieve(m);
        pictureBox1.Image = m.ToImage<Bgr, byte>().Bitmap;
    }
    catch
    {
        //MessageBox.Show("Ошибка вывода изображения", "Ошибка!",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

private void button1_Click_1(object sender, EventArgs e) //CAM Stream - end
{
    if (stream_on)
    {
        try
        {
            if (webCams.Length == 0) //программа не распознает камеры: они вообще не
отображаются в списке

```



```

        {
            throw new Exception("Нет подходящих камер");
        }
        else if (comboBox_Cam.SelectedItem == null) //камера не выбрана: она
отображается в списке, но с ней никак не взаимодействуют
        {
            throw new Exception("Необходимо выбрать камеру");
        }
        else //во всех остальных случаях корректно запускаем вывод изображения
        {
            capture = new VideoCapture(selectedCameraId);

            capture.ImageGrabbed += Capture_ImageGrabbed;

            capture.Start();

            stream_on = false; //изменяем значение, для того, чтобы кнопка Stream
больше не реагировала, т.к. изображение и так уже выводится
        }
    }
    catch (Exception ex) //обработка исключений
    {
        MessageBox.Show(ex.Message, "Внимание!", MessageBoxButtons.OK,
        MessageBoxIcon.Warning);
        TEXT.Text = ex.Message;
    }
    } else
    {
        MessageBox.Show("Трансляция изображения уже ведется", "Информация",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        TEXT.Text = "Трансляция изображения уже ведется";
    }
}

private void button6_Click(object sender, EventArgs e) //ScreemShot - end
{
    if (capture != null)
    {
        TEXT.Text = "Создание скриншота";
        try
        {
            Mat m = new Mat();
            capture.Retrieve(m);
            MakeScreenShotForm screenShotForm = new
MakeScreenShotForm(m.ToImage<Bgr, byte>());

```

```

        screenshotForm.Show();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Ошибка", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
        TEXT.Text = ex.Message;
    }
}
else
{
    MessageBox.Show("Необходимо выбрать камеру", "Внимание!",
    MessageBoxButtons.OK, MessageBoxIcon.Warning);
    TEXT.Text = "Необходимо выбрать камеру";
}
}

private void button7_Click(object sender, EventArgs e) //TL: start - end
{
    if (capture != null)
    {
        TEXT.Text = "TimeLapse: начало записи с *значения* параметрами";

    }
    else
    {
        MessageBox.Show("Необходимо выбрать камеру", "Внимание!",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
        TEXT.Text = "Необходимо выбрать камеру";
    }
}

private void button8_Click(object sender, EventArgs e) //TL: stop - end
{
    if (capture != null)
    {
        TEXT.Text = "TimeLapse: запись остановлена и сохранена в памяти устройства";
    }
    else
    {
        MessageBox.Show("Необходимо выбрать камеру", "Внимание!",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
        TEXT.Text = "Необходимо выбрать камеру";
    }
}
}

```

```

//УПРАВЛЕНИЕ
private void comboBox1_SelectedIndexChanged(object sender, EventArgs e) // Выбор
порта - end
{
    try
    {
        if (Current_Port.IsOpen)
        {
            Current_Port.Close();
        }
        Current_Port.PortName = COM.SelectedItem as string;
        Current_Port.BaudRate = 115200;
        Current_Port.Open();
        TEXT.Text = "Порт открыт";
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Внимание!", MessageBoxButtons.OK,
        MessageBoxIcon.Warning);
        TEXT.Text = ex.Message;
    }
}

private void button1_Click(object sender, EventArgs e) //Turn left - end
{
    if (Current_Port.IsOpen)
    {
        Current_Port.Write("turn left");
        TEXT.Text = "Джойстик управления: поворот влево";
    }
    else
    {
        TEXT.Text = "Выполнить действие нельзя! Выберите необходимый порт для
подключения";
        MessageBox.Show("Необходимо выбрать порт", "Внимание!",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
}

private void button2_Click(object sender, EventArgs e) //Turn right - end
{
    if (Current_Port.IsOpen) {
        Current_Port.Write("turn right");

```

```

        TEXT.Text = "Джойстик управления: поворот вправо";
    }
    else
    {
        TEXT.Text = "Выполнить действие нельзя! Выберите необходимый порт для
подключения";
        MessageBox.Show("Необходимо выбрать порт", "Внимание!",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
}

private void button3_Click(object sender, EventArgs e) //Push - end
{
    if (Current_Port.IsOpen)
    {
        Current_Port.Write("push");
        TEXT.Text = "Джойстик управления: нажатие";
    }
    else
    {
        TEXT.Text = "Выполнить действие нельзя! Выберите необходимый порт для
подключения";
        MessageBox.Show("Необходимо выбрать порт", "Внимание!",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
}

private void button_cam_menu_Click(object sender, EventArgs e) //To menu - end
{
    if (Current_Port.IsOpen || capture != null)
    {
        Current_Port.Write("cam menu");
        TEXT.Text = "Камера: перемещение камеры в область просмотра меню";
    }
    else
    {
        if (capture == null && Current_Port.IsOpen)
        {
            MessageBox.Show("Необходимо выбрать камеру", "Внимание!",
            MessageBoxButtons.OK, MessageBoxIcon.Warning);
            TEXT.Text = "Необходимо выбрать камеру";
        }
        else if (capture != null && !Current_Port.IsOpen)
        {

```

```

        MessageBox.Show("Необходимо выбрать порт", "Внимание!",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
        TEXT.Text = "Выполнить действие нельзя! Выберите необходимый порт для
подключения";
    }
    else
    {
        MessageBox.Show("Необходимо выбрать порт и камеру", "Внимание!",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
        TEXT.Text = "Выполнить действие нельзя! Выберите необходимый порт для
подключения и камеру для трансляции";
    }
}

private void button_cam_workspace_Click(object sender, EventArgs e) //To Workspace -
end
{
    if (Current_Port.IsOpen || capture != null)
    {
        Current_Port.Write("cam workspace");
        TEXT.Text = "Камера: перемещение камеры в область печати";
    }
    else
    {
        if (capture == null && Current_Port.IsOpen)
        {
            MessageBox.Show("Необходимо выбрать камеру", "Внимание!",
            MessageBoxButtons.OK, MessageBoxIcon.Warning);
            TEXT.Text = "Необходимо выбрать камеру";
        }
        else if (capture != null && !Current_Port.IsOpen)
        {
            MessageBox.Show("Необходимо выбрать порт", "Внимание!",
            MessageBoxButtons.OK, MessageBoxIcon.Warning);
            TEXT.Text = "Выполнить действие нельзя! Выберите необходимый порт для
подключения";
        }
        else
        {
            MessageBox.Show("Необходимо выбрать порт и камеру", "Внимание!",
            MessageBoxButtons.OK, MessageBoxIcon.Warning);
            TEXT.Text = "Выполнить действие нельзя! Выберите необходимый порт для
подключения и камеру для трансляции";
        }
    }
}

```

```

    }
  }
}
}
}

```

2.2

```

namespace _3D_Printer___Main_version
{
    partial class Form1
    {
        /// <summary>
        /// Обязательная переменная конструктора.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Освободить все используемые ресурсы.
        /// </summary>
        /// <param name="disposing">истинно, если управляемый ресурс должен быть удален;
        /// иначе ложно.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Код, автоматически созданный конструктором форм Windows

        /// <summary>
        /// Требуемый метод для поддержки конструктора — не изменяйте
        /// содержимое этого метода с помощью редактора кода.
        /// </summary>
        private void InitializeComponent()
        {
            this.components = new System.ComponentModel.Container();
            System.ComponentModel.ComponentResourceManager resources = new
System.ComponentModel.ComponentResourceManager(typeof(Form1));
            this.COM = new System.Windows.Forms.ComboBox();
            this.button_left = new System.Windows.Forms.Button();
            this.button_right = new System.Windows.Forms.Button();
            this.button_push = new System.Windows.Forms.Button();
        }
    }
}

```

```

this.label1 = new System.Windows.Forms.Label();
this.label2 = new System.Windows.Forms.Label();
this.label3 = new System.Windows.Forms.Label();
this.button_cam_menu = new System.Windows.Forms.Button();
this.button_cam_workspace = new System.Windows.Forms.Button();
this.label4 = new System.Windows.Forms.Label();
this.button_screenshot = new System.Windows.Forms.Button();
this.button_tl_start = new System.Windows.Forms.Button();
this.button_tl_stop = new System.Windows.Forms.Button();
this.pictureBox1 = new System.Windows.Forms.PictureBox();
this.comboBox_Cam = new System.Windows.Forms.ComboBox();
this.groupBox_Control = new System.Windows.Forms.GroupBox();
this.label5 = new System.Windows.Forms.Label();
this.button_Stream = new System.Windows.Forms.Button();
this.groupBox1 = new System.Windows.Forms.GroupBox();
this.groupBox2 = new System.Windows.Forms.GroupBox();
this.toolTip_Left = new System.Windows.Forms.ToolTip(this.components);
this.vScrollBar1 = new System.Windows.Forms.VScrollBar();
this.TEXT = new System.Windows.Forms.TextBox();
((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).BeginInit();
this.groupBox_Control.SuspendLayout();
this.groupBox1.SuspendLayout();
this.groupBox2.SuspendLayout();
this.SuspendLayout();
//
// COM
//
this.COM.BackColor = System.Drawing.SystemColors.Window;
this.COM.Cursor = System.Windows.Forms.Cursors.Hand;
this.COM.DropDownStyle = System.Windows.Forms.ComboBoxStyle.DropDownList;
this.COM.FlatStyle = System.Windows.Forms.FlatStyle.System;
this.COM.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
this.COM.ForeColor = System.Drawing.SystemColors.ActiveCaptionText;
this.COM.FormattingEnabled = true;
this.COM.Location = new System.Drawing.Point(10, 50);
this.COM.Margin = new System.Windows.Forms.Padding(0);
this.COM.MinimumSize = new System.Drawing.Size(165, 0);
this.COM.Name = "COM";
this.COM.Size = new System.Drawing.Size(165, 23);
this.COM.TabIndex = 0;
this.COM.SelectedIndexChanged += new
System.EventHandler(this.comboBox1_SelectedIndexChanged);
this.COM.SelectedValueChanged += new
System.EventHandler(this.comboBox1_SelectedIndexChanged);

```

```

//
// button_left
//
this.button_left.BackColor = System.Drawing.SystemColors.ControlLightLight;
this.button_left.ForeColor = System.Drawing.Color.Transparent;
this.button_left.Image =
((System.Drawing.Image)(resources.GetObject("button_left.Image")));
this.button_left.Location = new System.Drawing.Point(10, 113);
this.button_left.MinimumSize = new System.Drawing.Size(80, 40);
this.button_left.Name = "button_left";
this.button_left.Size = new System.Drawing.Size(80, 40);
this.button_left.TabIndex = 2;
this.toolTip_Left.SetToolTip(this.button_left, "Джойстик управления повернется на 1
шаг влево");
this.button_left.UseVisualStyleBackColor = false;
this.button_left.Click += new System.EventHandler(this.button1_Click);
//
// button_right
//
this.button_right.ForeColor = System.Drawing.Color.Transparent;
this.button_right.Image =
((System.Drawing.Image)(resources.GetObject("button_right.Image")));
this.button_right.Location = new System.Drawing.Point(95, 113);
this.button_right.MinimumSize = new System.Drawing.Size(80, 40);
this.button_right.Name = "button_right";
this.button_right.Size = new System.Drawing.Size(80, 40);
this.button_right.TabIndex = 3;
this.button_right.UseVisualStyleBackColor = true;
this.button_right.Click += new System.EventHandler(this.button2_Click);
//
// button_push
//
this.button_push.Image =
((System.Drawing.Image)(resources.GetObject("button_push.Image")));
this.button_push.Location = new System.Drawing.Point(10, 158);
this.button_push.MinimumSize = new System.Drawing.Size(165, 40);
this.button_push.Name = "button_push";
this.button_push.Size = new System.Drawing.Size(165, 40);
this.button_push.TabIndex = 4;
this.button_push.UseVisualStyleBackColor = true;
this.button_push.Click += new System.EventHandler(this.button3_Click);
//
// label1
//
this.label1.BackColor = System.Drawing.Color.Transparent;

```



```

this.label1.FlatStyle = System.Windows.Forms.FlatStyle.Flat;
this.label1.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Underline, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
this.label1.ForeColor = System.Drawing.SystemColors.Window;
this.label1.Location = new System.Drawing.Point(10, 30);
this.label1.MinimumSize = new System.Drawing.Size(165, 15);
this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(165, 15);
this.label1.TabIndex = 5;
this.label1.Text = "Выбор порта подключения";
this.label1.TextAlign = System.Drawing.ContentAlignment.TopCenter;
//
// label2
//
this.label2.BackColor = System.Drawing.Color.Transparent;
this.label2.FlatStyle = System.Windows.Forms.FlatStyle.Flat;
this.label2.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Underline, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
this.label2.ForeColor = System.Drawing.SystemColors.Window;
this.label2.Location = new System.Drawing.Point(10, 93);
this.label2.MinimumSize = new System.Drawing.Size(165, 15);
this.label2.Name = "label2";
this.label2.Size = new System.Drawing.Size(165, 15);
this.label2.TabIndex = 6;
this.label2.Text = "Управление печатью";
this.label2.TextAlign = System.Drawing.ContentAlignment.TopCenter;
//
// label3
//
this.label3.BackColor = System.Drawing.Color.Transparent;
this.label3.FlatStyle = System.Windows.Forms.FlatStyle.Flat;
this.label3.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Underline, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
this.label3.ForeColor = System.Drawing.SystemColors.Window;
this.label3.Location = new System.Drawing.Point(10, 218);
this.label3.MinimumSize = new System.Drawing.Size(165, 15);
this.label3.Name = "label3";
this.label3.Size = new System.Drawing.Size(165, 15);
this.label3.TabIndex = 7;
this.label3.Text = "Выбор камеры";
this.label3.TextAlign = System.Drawing.ContentAlignment.TopCenter;
//
// button_cam_menu
//

```

```

        this.button_cam_menu.Image =
((System.Drawing.Image)(resources.GetObject("button_cam_menu.Image")));
        this.button_cam_menu.Location = new System.Drawing.Point(10, 346);
        this.button_cam_menu.MinimumSize = new System.Drawing.Size(80, 40);
        this.button_cam_menu.Name = "button_cam_menu";
        this.button_cam_menu.Size = new System.Drawing.Size(80, 40);
        this.button_cam_menu.TabIndex = 8;
        this.button_cam_menu.UseVisualStyleBackColor = true;
        this.button_cam_menu.Click += new
System.EventHandler(this.button_cam_menu_Click);
        //
        // button_cam_workspace
        //
        this.button_cam_workspace.Image =
((System.Drawing.Image)(resources.GetObject("button_cam_workspace.Image")));
        this.button_cam_workspace.Location = new System.Drawing.Point(95, 346);
        this.button_cam_workspace.MinimumSize = new System.Drawing.Size(80, 40);
        this.button_cam_workspace.Name = "button_cam_workspace";
        this.button_cam_workspace.Size = new System.Drawing.Size(80, 40);
        this.button_cam_workspace.TabIndex = 9;
        this.button_cam_workspace.UseVisualStyleBackColor = true;
        this.button_cam_workspace.Click += new
System.EventHandler(this.button_cam_workspace_Click);
        //
        // label4
        //
        this.label4.BackColor = System.Drawing.Color.Transparent;
        this.label4.FlatStyle = System.Windows.Forms.FlatStyle.Flat;
        this.label4.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Underline, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
        this.label4.ForeColor = System.Drawing.SystemColors.Window;
        this.label4.Location = new System.Drawing.Point(10, 326);
        this.label4.MinimumSize = new System.Drawing.Size(165, 15);
        this.label4.Name = "label4";
        this.label4.Size = new System.Drawing.Size(165, 15);
        this.label4.TabIndex = 10;
        this.label4.Text = "Положение камеры";
        this.label4.TextAlign = System.Drawing.ContentAlignment.TopCenter;
        //
        // button_screenshot
        //
        this.button_screenshot.Image =
((System.Drawing.Image)(resources.GetObject("button_screenshot.Image")));
        this.button_screenshot.Location = new System.Drawing.Point(10, 426);
        this.button_screenshot.MinimumSize = new System.Drawing.Size(165, 40);

```

```

this.button_screenshot.Name = "button_screenshot";
this.button_screenshot.Size = new System.Drawing.Size(165, 40);
this.button_screenshot.TabIndex = 11;
this.button_screenshot.UseVisualStyleBackColor = true;
this.button_screenshot.Click += new System.EventHandler(this.button6_Click);
//
// button_tl_start
//
this.button_tl_start.Image =
((System.Drawing.Image)(resources.GetObject("button_tl_start.Image")));
this.button_tl_start.Location = new System.Drawing.Point(10, 471);
this.button_tl_start.MinimumSize = new System.Drawing.Size(165, 40);
this.button_tl_start.Name = "button_tl_start";
this.button_tl_start.Size = new System.Drawing.Size(165, 40);
this.button_tl_start.TabIndex = 12;
this.button_tl_start.UseVisualStyleBackColor = true;
this.button_tl_start.Click += new System.EventHandler(this.button7_Click);
//
// button_tl_stop
//
this.button_tl_stop.Image =
((System.Drawing.Image)(resources.GetObject("button_tl_stop.Image")));
this.button_tl_stop.Location = new System.Drawing.Point(10, 516);
this.button_tl_stop.MinimumSize = new System.Drawing.Size(165, 40);
this.button_tl_stop.Name = "button_tl_stop";
this.button_tl_stop.Size = new System.Drawing.Size(165, 40);
this.button_tl_stop.TabIndex = 13;
this.button_tl_stop.UseVisualStyleBackColor = true;
this.button_tl_stop.Click += new System.EventHandler(this.button8_Click);
//
// pictureBox1
//
this.pictureBox1.Location = new System.Drawing.Point(10, 30);
this.pictureBox1.MinimumSize = new System.Drawing.Size(702, 526);
this.pictureBox1.Name = "pictureBox1";
this.pictureBox1.Size = new System.Drawing.Size(702, 526);
this.pictureBox1.SizeMode = System.Windows.Forms.PictureBoxSizeMode.Zoom;
this.pictureBox1.TabIndex = 14;
this.pictureBox1.TabStop = false;
//
// comboBox_Cam
//
this.comboBox_Cam.BackColor = System.Drawing.SystemColors.Window;
this.comboBox_Cam.Cursor = System.Windows.Forms.Cursors.Hand;

```

```

        this.comboBox_Cam.DropDownStyle =
System.Windows.Forms.ComboBoxStyle.DropDownList;
        this.comboBox_Cam.FlatStyle = System.Windows.Forms.FlatStyle.System;
        this.comboBox_Cam.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
        this.comboBox_Cam.ForeColor = System.Drawing.SystemColors.ActiveCaptionText;
        this.comboBox_Cam.FormattingEnabled = true;
        this.comboBox_Cam.Location = new System.Drawing.Point(10, 238);
        this.comboBox_Cam.Margin = new System.Windows.Forms.Padding(0);
        this.comboBox_Cam.MinimumSize = new System.Drawing.Size(165, 0);
        this.comboBox_Cam.Name = "comboBox_Cam";
        this.comboBox_Cam.Size = new System.Drawing.Size(165, 23);
        this.comboBox_Cam.TabIndex = 15;
        this.comboBox_Cam.SelectedIndexChanged += new
System.EventHandler(this.comboBox_Cam_SelectedIndexChanged);
        //
        // groupBox_Control
        //
        this.groupBox_Control.BackColor = System.Drawing.Color.Transparent;
        this.groupBox_Control.BackgroundImageLayout =
System.Windows.Forms.ImageLayout.None;
        this.groupBox_Control.Controls.Add(this.label5);
        this.groupBox_Control.Controls.Add(this.button_Stream);
        this.groupBox_Control.Controls.Add(this.comboBox_Cam);
        this.groupBox_Control.Controls.Add(this.button_tl_stop);
        this.groupBox_Control.Controls.Add(this.COM);
        this.groupBox_Control.Controls.Add(this.button_tl_start);
        this.groupBox_Control.Controls.Add(this.button_left);
        this.groupBox_Control.Controls.Add(this.button_screenshot);
        this.groupBox_Control.Controls.Add(this.button_cam_workspace);
        this.groupBox_Control.Controls.Add(this.button_cam_menu);
        this.groupBox_Control.Controls.Add(this.button_right);
        this.groupBox_Control.Controls.Add(this.button_push);
        this.groupBox_Control.Controls.Add(this.label1);
        this.groupBox_Control.Controls.Add(this.label4);
        this.groupBox_Control.Controls.Add(this.label2);
        this.groupBox_Control.Controls.Add(this.label3);
        this.groupBox_Control.ForeColor = System.Drawing.Color.Black;
        this.groupBox_Control.Location = new System.Drawing.Point(6, 21);
        this.groupBox_Control.Margin = new System.Windows.Forms.Padding(5);
        this.groupBox_Control.MinimumSize = new System.Drawing.Size(185, 565);
        this.groupBox_Control.Name = "groupBox_Control";
        this.groupBox_Control.Size = new System.Drawing.Size(185, 565);
        this.groupBox_Control.TabIndex = 16;
        this.groupBox_Control.TabStop = false;

```

```

this.groupBox_Control.Text = "Управление";
//
// label5
//
this.label5.BackColor = System.Drawing.Color.Transparent;
this.label5.FlatStyle = System.Windows.Forms.FlatStyle.Flat;
this.label5.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Underline, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
this.label5.ForeColor = System.Drawing.SystemColors.Window;
this.label5.Location = new System.Drawing.Point(10, 406);
this.label5.MinimumSize = new System.Drawing.Size(165, 15);
this.label5.Name = "label5";
this.label5.Size = new System.Drawing.Size(165, 15);
this.label5.TabIndex = 18;
this.label5.Text = "Работа с медиафайлами";
this.label5.TextAlign = System.Drawing.ContentAlignment.TopCenter;
//
// button_Stream
//
this.button_Stream.ForeColor = System.Drawing.Color.Transparent;
this.button_Stream.Image =
((System.Drawing.Image)(resources.GetObject("button_Stream.Image")));
this.button_Stream.Location = new System.Drawing.Point(10, 266);
this.button_Stream.MinimumSize = new System.Drawing.Size(165, 40);
this.button_Stream.Name = "button_Stream";
this.button_Stream.Size = new System.Drawing.Size(165, 40);
this.button_Stream.TabIndex = 17;
this.button_Stream.UseVisualStyleBackColor = true;
this.button_Stream.Click += new System.EventHandler(this.button1_Click_1);
//
// groupBox1
//
this.groupBox1.BackColor = System.Drawing.Color.Transparent;
this.groupBox1.Controls.Add(this.pictureBox1);
this.groupBox1.ForeColor = System.Drawing.Color.Black;
this.groupBox1.Location = new System.Drawing.Point(197, 21);
this.groupBox1.MinimumSize = new System.Drawing.Size(722, 565);
this.groupBox1.Name = "groupBox1";
this.groupBox1.Size = new System.Drawing.Size(722, 565);
this.groupBox1.TabIndex = 17;
this.groupBox1.TabStop = false;
this.groupBox1.Text = "Контент";
//
// groupBox2
//

```

```

this.groupBox2.BackColor = System.Drawing.Color.Transparent;
this.groupBox2.Controls.Add(this.vScrollBar1);
this.groupBox2.ForeColor = System.Drawing.Color.Black;
this.groupBox2.Location = new System.Drawing.Point(925, 21);
this.groupBox2.Name = "groupBox2";
this.groupBox2.Size = new System.Drawing.Size(275, 565);
this.groupBox2.TabIndex = 18;
this.groupBox2.TabStop = false;
this.groupBox2.Text = "Справка";
//
// vScrollBar1
//
this.vScrollBar1.Location = new System.Drawing.Point(258, 16);
this.vScrollBar1.Name = "vScrollBar1";
this.vScrollBar1.Size = new System.Drawing.Size(17, 540);
this.vScrollBar1.TabIndex = 0;
this.toolTip_Left.SetToolTip(this.vScrollBar1, " ");
//
// TEXT
//
this.TEXT.BackColor = System.Drawing.Color.FromArgb(((int)(((byte)(140)))),
((int)(((byte)(82)))), ((int)(((byte)(255)))));
this.TEXT.BorderStyle = System.Windows.Forms.BorderStyle.None;
this.TEXT.Cursor = System.Windows.Forms.Cursors.Arrow;
this.TEXT.Dock = System.Windows.Forms.DockStyle.Top;
this.TEXT.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
this.TEXT.ForeColor = System.Drawing.SystemColors.Window;
this.TEXT.Location = new System.Drawing.Point(0, 0);
this.TEXT.Margin = new System.Windows.Forms.Padding(0);
this.TEXT.Name = "TEXT";
this.TEXT.ReadOnly = true;
this.TEXT.RightToLeft = System.Windows.Forms.RightToLeft.No;
this.TEXT.Size = new System.Drawing.Size(1210, 19);
this.TEXT.TabIndex = 1;
this.TEXT.TextAlign = System.Windows.Forms.HorizontalAlignment.Center;
//
// Form1
//
this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
this.BackColor = System.Drawing.SystemColors.Window;
this.BackgroundImage =
((System.Drawing.Image)(resources.GetObject("$this.BackgroundImage")));
this.BackgroundImageLayout = System.Windows.Forms.ImageLayout.Stretch;

```

```

this.ClientSize = new System.Drawing.Size(1210, 685);
this.Controls.Add(this.groupBox2);
this.Controls.Add(this.groupBox1);
this.Controls.Add(this.groupBox_Control);
this.Controls.Add(this.TEXT);
this.Cursor = System.Windows.Forms.Cursors.Default;
this.DoubleBuffered = true;
this.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
this.Icon = ((System.Drawing.Icon)(resources.GetObject("$this.Icon")));
this.MinimumSize = new System.Drawing.Size(1226, 634);
this.Name = "Form1";
this.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen;
this.Text = "Picaso Designer X: дистанционное управление";
this.WindowState = System.Windows.Forms.FormWindowState.Maximized;
this.Load += new System.EventHandler(this.Form1_Load);
((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).EndInit();
this.groupBox_Control.ResumeLayout(false);
this.groupBox1.ResumeLayout(false);
this.groupBox2.ResumeLayout(false);
this.ResumeLayout(false);
this.PerformLayout();
}

```

#endregion

```

private System.Windows.Forms.ComboBox COM;
private System.Windows.Forms.Button button_left;
private System.Windows.Forms.Button button_right;
private System.Windows.Forms.Button button_push;
private System.Windows.Forms.Label label1;
private System.Windows.Forms.Label label2;
private System.Windows.Forms.Label label3;
private System.Windows.Forms.Button button_cam_menu;
private System.Windows.Forms.Button button_cam_workspace;
private System.Windows.Forms.Label label4;
private System.Windows.Forms.Button button_screenshot;
private System.Windows.Forms.Button button_tl_start;
private System.Windows.Forms.Button button_tl_stop;
private System.Windows.Forms.PictureBox pictureBox1;
private System.Windows.Forms.ComboBox comboBox_Cam;
private System.Windows.Forms.GroupBox groupBox_Control;
private System.Windows.Forms.Button button_Stream;
private System.Windows.Forms.Label label5;

```

```

private System.Windows.Forms.GroupBox groupBox1;
private System.Windows.Forms.GroupBox groupBox2;
private System.Windows.Forms.ToolTip toolTip_Left;
private System.Windows.Forms.VScrollBar vScrollBar1;
private System.Windows.Forms.TextBox TEXT;
    }
}

```

3.1

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Drawing.Imaging;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using Emgu;
using Emgu.CV;
using Emgu.CV.Util;
using Emgu.CV.Structure;
using Emgu.Util;
using System.IO;

namespace _3D_Printer___Main_version
{
    public partial class MakeScreenShotForm : Form
    {
        private Image<Bgr, byte> image = null;
        private string filename = string.Empty;

        public MakeScreenShotForm(Image<Bgr, byte> image)
        {
            this.image = image;
            InitializeComponent();
        }

        private void MakeScreenShotForm_Load(object sender, EventArgs e)
        {
            filename =
$"Picaso712_{DateTime.Now.Day}{DateTime.Now.Month}{DateTime.Now.Year}{DateTime.
Now.Hour}{DateTime.Now.Minute}{DateTime.Now.Second}.png";
            pictureBox1.Image = image.Bitmap;
        }
    }
}

```



```

    }

    private void toolStripButton1_Click(object sender, EventArgs e)
    {
        try
        {
            pictureBox1.Image.Save(filename, ImageFormat.Png);

            if (File.Exists(filename))
            {
                Close();
            }
            else
            {
                throw new Exception("Не удалось сохранить скриншот");
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message, "Ошибка!", MessageBoxButtons.OK,
            MessageBoxIcon.Error);
        }
    }

    private void toolStripButton2_Click(object sender, EventArgs e)
    {
        Close ();
    }
}

```

3.2

```

namespace _3D_Printer___Main_version
{
    partial class MakeScreenShotForm
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
    }
}

```

```

    /// <param name="disposing">true if managed resources should be disposed; otherwise,
    false.</param>
    protected override void Dispose(bool disposing)
    {
        if (disposing && (components != null))
        {
            components.Dispose();
        }
        base.Dispose(disposing);
    }

    #region Windows Form Designer generated code

    /// <summary>
    /// Required method for Designer support - do not modify
    /// the contents of this method with the code editor.
    /// </summary>
    private void InitializeComponent()
    {
        System.ComponentModel.ComponentResourceManager resources = new
System.ComponentModel.ComponentResourceManager(typeof(MakeScreenShotForm));
        this.toolStrip1 = new System.Windows.Forms.ToolStrip();
        this.toolStripLabel1 = new System.Windows.Forms.ToolStripLabel();
        this.toolStripButton1 = new System.Windows.Forms.ToolStripButton();
        this.toolStripButton2 = new System.Windows.Forms.ToolStripButton();
        this.toolStripSeparator1 = new System.Windows.Forms.ToolStripSeparator();
        this.pictureBox1 = new System.Windows.Forms.PictureBox();
        this.toolStrip1.SuspendLayout();
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).BeginInit();
        this.SuspendLayout();
        //
        // toolStrip1
        //
        this.toolStrip1.BackColor = System.Drawing.Color.FromArgb(((int)(((byte)(140)))),
((int)(((byte)(82)))), ((int)(((byte)(255)))));
        this.toolStrip1.Items.AddRange(new System.Windows.Forms.ToolStripItem[] {
            this.toolStripLabel1,
            this.toolStripButton1,
            this.toolStripSeparator1,
            this.toolStripButton2});
        this.toolStrip1.Location = new System.Drawing.Point(0, 0);
        this.toolStrip1.Name = "toolStrip1";
        this.toolStrip1.Size = new System.Drawing.Size(800, 25);
        this.toolStrip1.TabIndex = 0;
        this.toolStrip1.Text = "toolStrip1";
    }

```

```

//
// toolStripLabel1
//
this.toolStripLabel1.ForeColor = System.Drawing.Color.White;
this.toolStripLabel1.Name = "toolStripLabel1";
this.toolStripLabel1.Size = new System.Drawing.Size(118, 22);
this.toolStripLabel1.Text = "Выберете действие: ";
//
// toolStripButton1
//
this.toolStripButton1.DisplayStyle =
System.Windows.Forms.ToolStripItemDisplayStyle.Text;
this.toolStripButton1.ForeColor = System.Drawing.Color.White;
this.toolStripButton1.Image =
((System.Drawing.Image)(resources.GetObject("toolStripButton1.Image")));
this.toolStripButton1.ImageTransparentColor = System.Drawing.Color.Magenta;
this.toolStripButton1.Name = "toolStripButton1";
this.toolStripButton1.Size = new System.Drawing.Size(70, 22);
this.toolStripButton1.Text = "Сохранить";
this.toolStripButton1.Click += new System.EventHandler(this.toolStripButton1_Click);
//
// toolStripButton2
//
this.toolStripButton2.DisplayStyle =
System.Windows.Forms.ToolStripItemDisplayStyle.Text;
this.toolStripButton2.ForeColor = System.Drawing.Color.White;
this.toolStripButton2.Image =
((System.Drawing.Image)(resources.GetObject("toolStripButton2.Image")));
this.toolStripButton2.ImageTransparentColor = System.Drawing.Color.Magenta;
this.toolStripButton2.Name = "toolStripButton2";
this.toolStripButton2.Size = new System.Drawing.Size(146, 22);
this.toolStripButton2.Text = "Закрыть без сохранения";
this.toolStripButton2.Click += new System.EventHandler(this.toolStripButton2_Click);
//
// toolStripSeparator1
//
this.toolStripSeparator1.Name = "toolStripSeparator1";
this.toolStripSeparator1.Size = new System.Drawing.Size(6, 25);
//
// pictureBox1
//
this.pictureBox1.Dock = System.Windows.Forms.DockStyle.Fill;
this.pictureBox1.Location = new System.Drawing.Point(0, 25);
this.pictureBox1.Name = "pictureBox1";
this.pictureBox1.Size = new System.Drawing.Size(800, 425);

```

```

        this.pictureBox1.SizeMode = System.Windows.Forms.PictureBoxSizeMode.Zoom;
        this.pictureBox1.TabIndex = 1;
        this.pictureBox1.TabStop = false;
        //
        // MakeScreenShotForm
        //
        this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
        this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
        this.ClientSize = new System.Drawing.Size(800, 450);
        this.Controls.Add(this.pictureBox1);
        this.Controls.Add(this.toolStrip1);
        this.Name = "MakeScreenShotForm";
        this.Text = "Создание скриншота";
        this.Load += new System.EventHandler(this.MakeScreenShotForm_Load);
        this.toolStrip1.ResumeLayout(false);
        this.toolStrip1.PerformLayout();
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).EndInit();
        this.ResumeLayout(false);
        this.PerformLayout();

    }

#endregion

private System.Windows.Forms.ToolStrip toolStrip1;
private System.Windows.Forms.ToolStripLabel toolStripLabel1;
private System.Windows.Forms.ToolStripButton toolStripButton1;
private System.Windows.Forms.ToolStripSeparator toolStripSeparator1;
private System.Windows.Forms.ToolStripButton toolStripButton2;
private System.Windows.Forms.PictureBox pictureBox1;
    }
}

```

4

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace _3D_Printer___Main_version
{
    internal static class Program
    {
        /// <summary>

```

```
/// Главная точка входа для приложения.  
/// </summary>  
[STAThread]  
static void Main()  
{  
    Application.EnableVisualStyles();  
    Application.SetCompatibleTextRenderingDefault(false);  
    Application.Run(new Form1());  
}  
}
```