

Методические рекомендации по решению
задач практического этапа
Московского конкурса межпредметных
навыков и знаний



Интеллектуальный мегаполис.

Потенциал



В номинации

Инженерный класс



по направлению

Инженерно-техническое

Оглавление

1. Задание	3
1.1. Схема	5
1.2. Пример работы программы	5
1.3. Критерии оценки работоспособности программы	5
1.4. Ссылки на проект	6
2. Решение	6
2.1. Настройка портов и интерфейсов	6
2.2. Управление магнитным захватом	7
2.3. Управление сервоприводом	8
2.4. Вычисление координат захвата	9
3. Платформа Tinkercad	10
3.1. Регистрация	10
3.2. Копирование проектов	12
3.3. Изменение проектов	13
3.4. Эмуляция и отладка	14

1. Задание

Необходимо решить практическую задачу по программированию перемещения детали, что изображена на рис. 1 условно в виде коричневого цилиндра, по четырем конвейерным участкам роботизированного сортировщика.

Составить программные алгоритмы управления элементами роботизированного сортировщика (рис. 2): электродвигателем конвейера Lx , сервомоторы конвейеров $Ly1$, $Ly2$, $Ly3$, светофорами **0-5**, для сигнализации – на каком участке находится деталь. Данные для перемещения деталей из стартовой позиции, расположенной в начале конвейера Lx , по четырем участкам роботизированного сортировщика приведены в табл. 1.

Необходимо создать контроллер системы распределения деталей на базе платы для прототипирования – Arduino UNO.

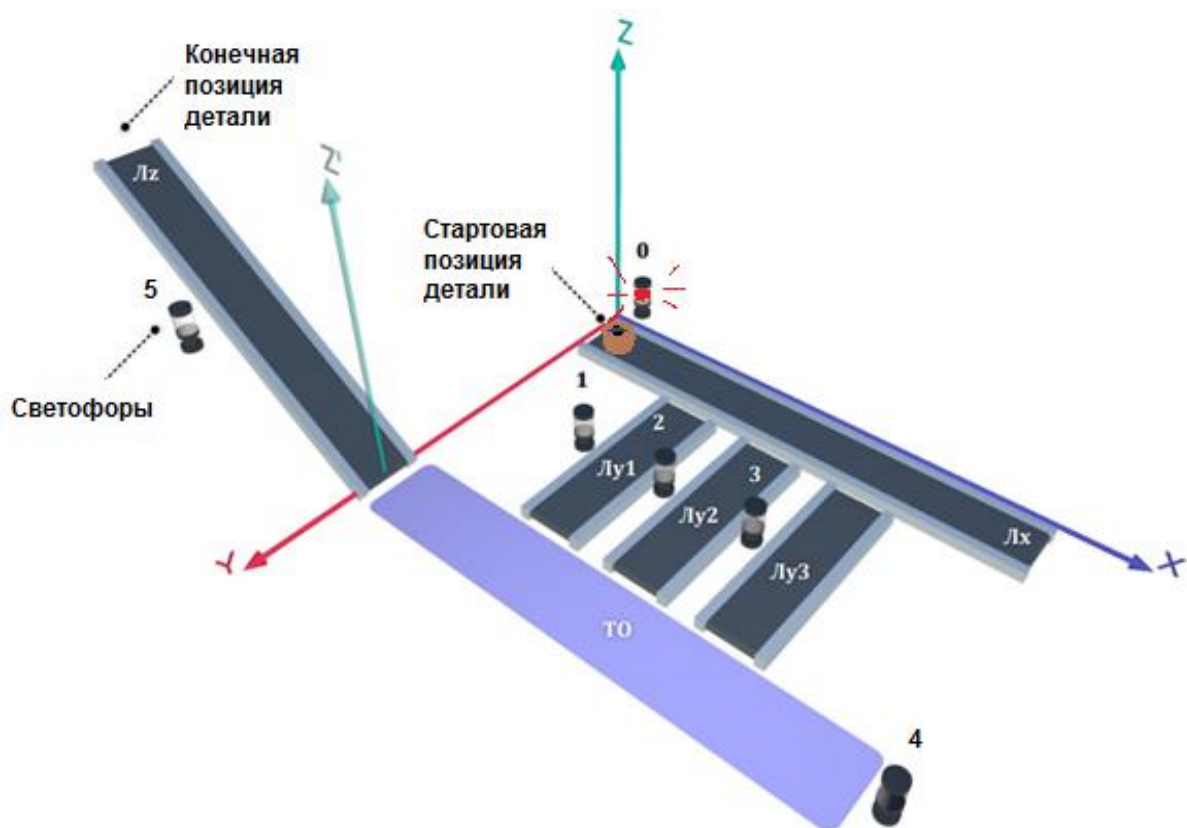


Рис. 1. Автоматизированная система перемещения деталей

Прохождение деталью каждого участка выполняется последовательно. Светофоры должны работать в двух режимах: не горит – деталь не находится на участке, горит – деталь находится на участке. Во время перемещения детали

электродвигатель на участке Лх должен вращаться в положительном направлении. Сервомоторы участков Лу и Лз, при прохождении по ним детали, должны повернуться на угол 90 градусов по часовой стрелке.

Запуск конвейера осуществляется после нажатия на кнопку «Начать моделирование» в ПО Tinkercad.

1.1. Схема

Внесение изменений в приведённую на рис. 2 схему устройства не допускается.

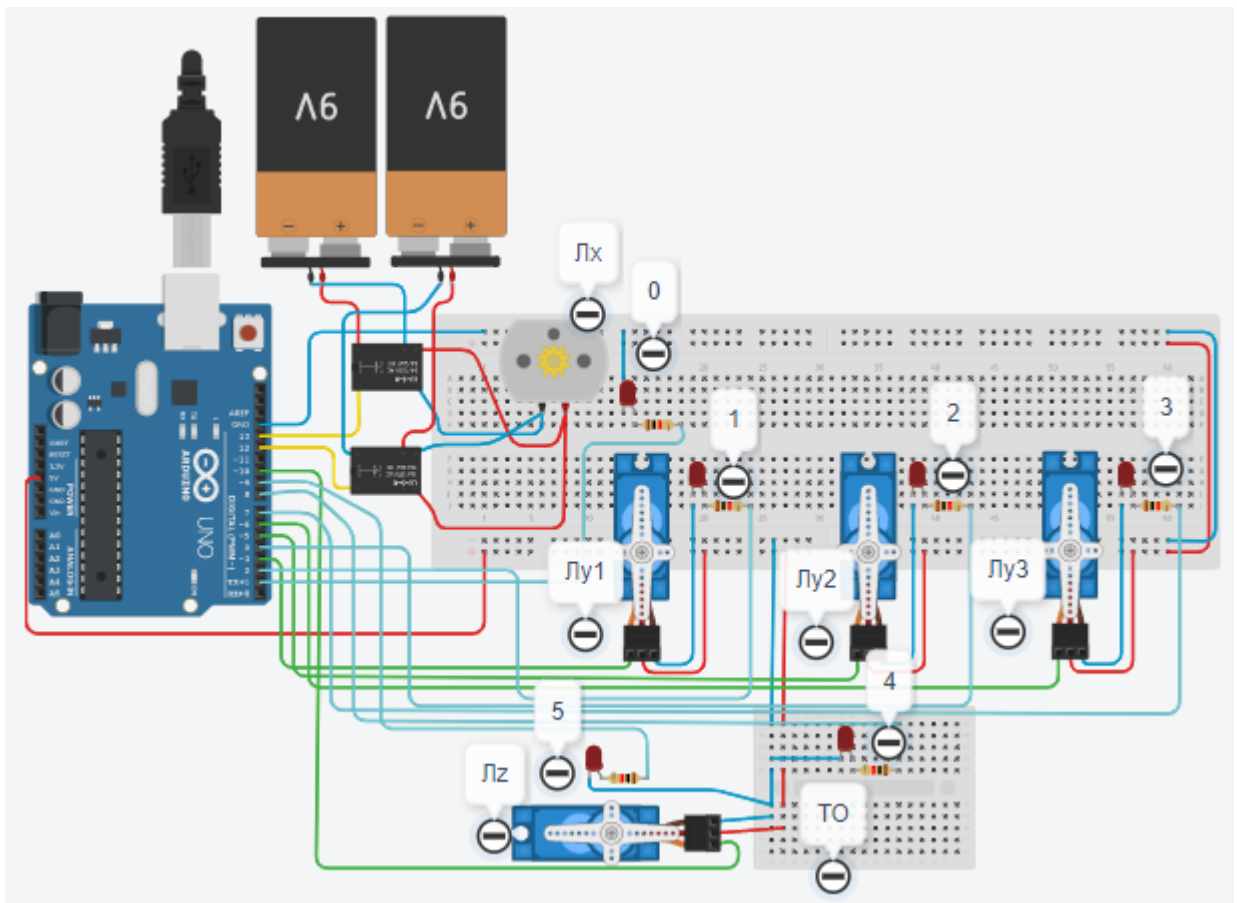


Рис. 2. Схема контроллера

Электродвигатель, сервомоторы и светофоры работают только на той ленте, где в заданное время находится деталь, на остальных участках они должны быть выключены. Начальная позиция детали может отличаться от рис. 1, например, деталь может находиться в начале конвейера Лу1. Когда деталь достигла конечной позиции Лз, выполнение программы останавливается. Программа завершается всегда в Лз.

1.2. Пример работы программы

Пример 1:

Входные данные:

№ п/п	Участок роботизированного сортировщика	Время, за которое деталь перемещается по участку, сек	На какой следующий участок происходит перемещение детали
1	2	3	4
1	Лх	2	Лу1
2	Лу1	3	ТО
3	ТО	2	Лз
4	Лз	1	Перемещение завершено

Таблица 1. Параметры сортировки деталей

Выходные данные:

8.0 сек

Первая колонка содержит порядковый номер участка. Вторая колонка отражает последовательность участков которые проходит деталь. В третьей колонке таблицы 1 указано количество секунд, на которое запускается электродвигатель и светофор, при нахождении детали на соответствующем участке (колонка 2). В колонке 4 указаны участки, на которые должна переходить деталь. Например, находясь в начальной позиции на конвейере Лх, деталь должна пройти по участку Лх до участка Лу1 и далее ТО, Лз.

1.3. Критерии оценки работоспособности программы.

- Реализована эмуляция последовательности включения светодиодов (индикация) согласно заданию — 15 баллов.
- Реализовано эмуляция управления электродвигателями согласно заданию — 15 баллов.

- Время, за которое деталь перемещается по всем участкам вычислено корректно и выведено через Serial интерфейс — 30 баллов.

1.4. Ссылки на проект

[Ссылка на схему в Tinkercad.](#)

2. Решение

2.1. Настройка портов и интерфейсов

Необходимо задать всем используемым портам и значениям имена, это повысит понимание кода и при необходимости упростит его изменение. Для работы с сервомоторами выполним импорт библиотеки Servo.h в самом начале программы с помощью директивы #include.

Импорт библиотеки Servo:

```
#include <Servo.h>
```

Объявление и инициализация объектов Servo и пинов для светофоров (светодиодов):

```
unsigned long start_time = 0;

Servo L1;
Servo L2;
Servo L3;
Servo LZ;

const int lamp_pinZ = 9;
const int lamp_pinTO = 8;
const int lamp_pin0 = 11;
const int lamp_pin1 = 2;
const int lamp_pin2 = 4;
const int lamp_pin3 = 7;
```

Начальная установка:

```
void setup()
```

```
{  
  Serial.begin(9600);  
  start_time = millis();  
  
  lamp_setup();  
  servo_setup();  
}
```

2.2. Основной цикл

Данная программа на Arduino используется для управления лампами и сервоприводами. В функции **setup** вызываются функции `lamp_setup` и `servo_setup` для инициализации ламп и сервоприводов соответственно. Затем в функции **loop** происходит последовательное управление лампами и сервоприводами в заданном порядке.

Функция **control_lamps** управляет состоянием лампы в зависимости от переданного значения **value**. Функция **perform_servo** исходя из переданного номера включает соответствующий сервопривод для проворота.

Функции **rotate_LZ**, **rotate_L1**, **rotate_L2**, и **rotate_L3** отвечают за поворот соответствующих сервоприводов на определенные углы.

Программа в цикле **loop** выполняет следующие последовательные действия:

1. Включает определенную лампу и проворачивает соответствующий сервопривод.
2. После этого программа выполняет аналогичные действия для других комбинаций управления лампами и сервоприводами.

Ниже приведён пример листинга для демонстрационного варианта 0-1-ТО-
z:

```
void loop()  
{  
  // 0
```

```
control_lamps(0);
perform_servo(1);

// 1
control_lamps(1);
perform_servo(1);

// TO
control_lamps(-2);
perform_servo(0);

// Z
control_lamps(-1);
perform_servo(0);

unsigned long current_time = millis();
unsigned long time_taken = current_time - start_time;
Serial.print("Cycle time: ");
Serial.print(time_taken);
Serial.println(" msec");
start_time = 0;
}
```

2.3. Функции

Настройка ламп:

```
void lamp_setup()
{
    // Инициализация пинов для ламп
}
```

Управление лампами:

```
void control_lamps(int value)
{
    // Установка состояния ламп в зависимости от value
}
```

Настройка сервоприводов:


```
void servo_setup()
{
  // Инициализация сервоприводов
}
```

Управление сервоприводами:

```
void perform_servo(int number)
{
  // Поворот выбранного сервопривода
}
```

Селектор сервоприводов:

```
void rotate_LZ()
{
  // Поворот LZ
}

void rotate_L1()
{
  // Поворот L1
}

void rotate_L2()
{
  // Поворот L2
}

void rotate_L3()
{
  // Поворот L3
}
```

Функция поворота сервоприводов (на примере LZ):

```
void rotate_LZ()
{
  for(int i=0; i<1; i++)
```

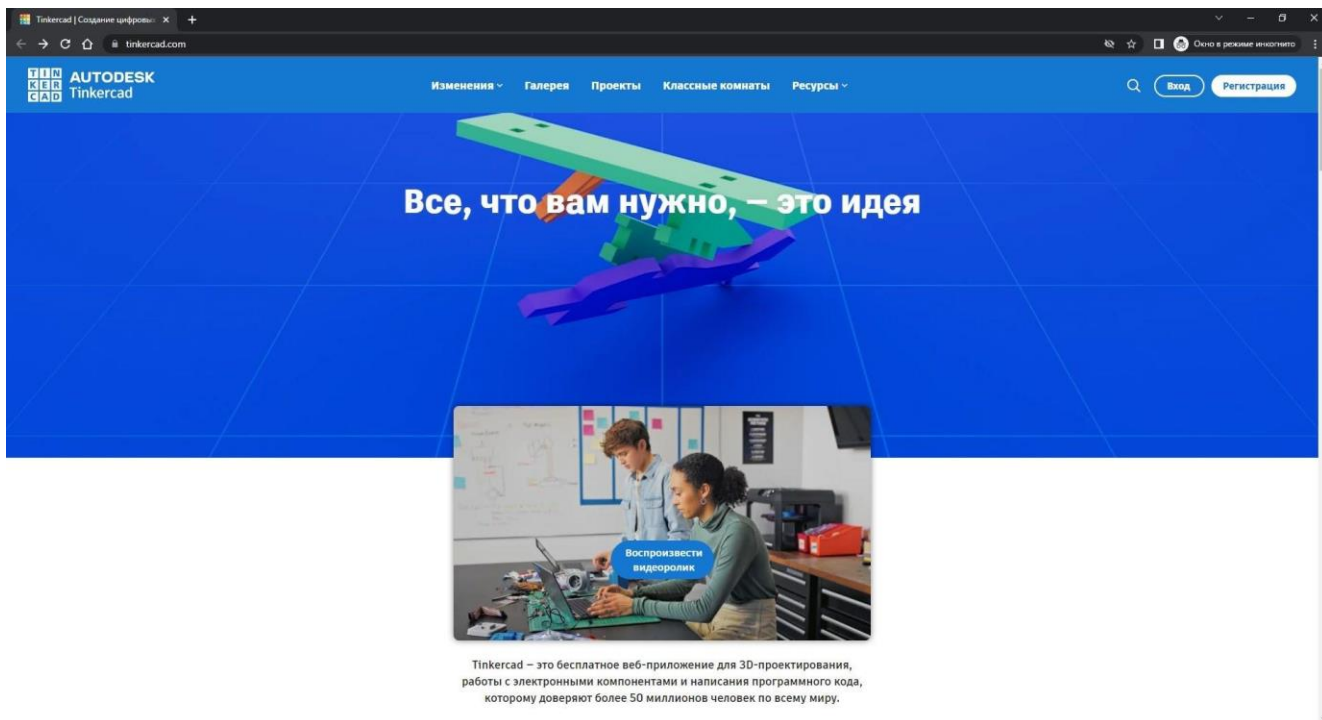
```
{
  for(int angle = 0; angle <= 180; angle++)
  {
    LZ.write(angle);
    delay(1);
  }
  for(int angle = 180; angle >= 0; angle--)
  {
    LZ.write(angle);
    delay(1);
  }
}
}
```

Программа управляет лампами и сервоприводами в цикле, используя функции для управления состоянием ламп и поворотом сервоприводов. В основном цикле **loop** последовательно вызываются различные комбинации управления лампами и сервоприводами в соответствии с заданными значениями. По завершению цикла выводится время его выполнения.

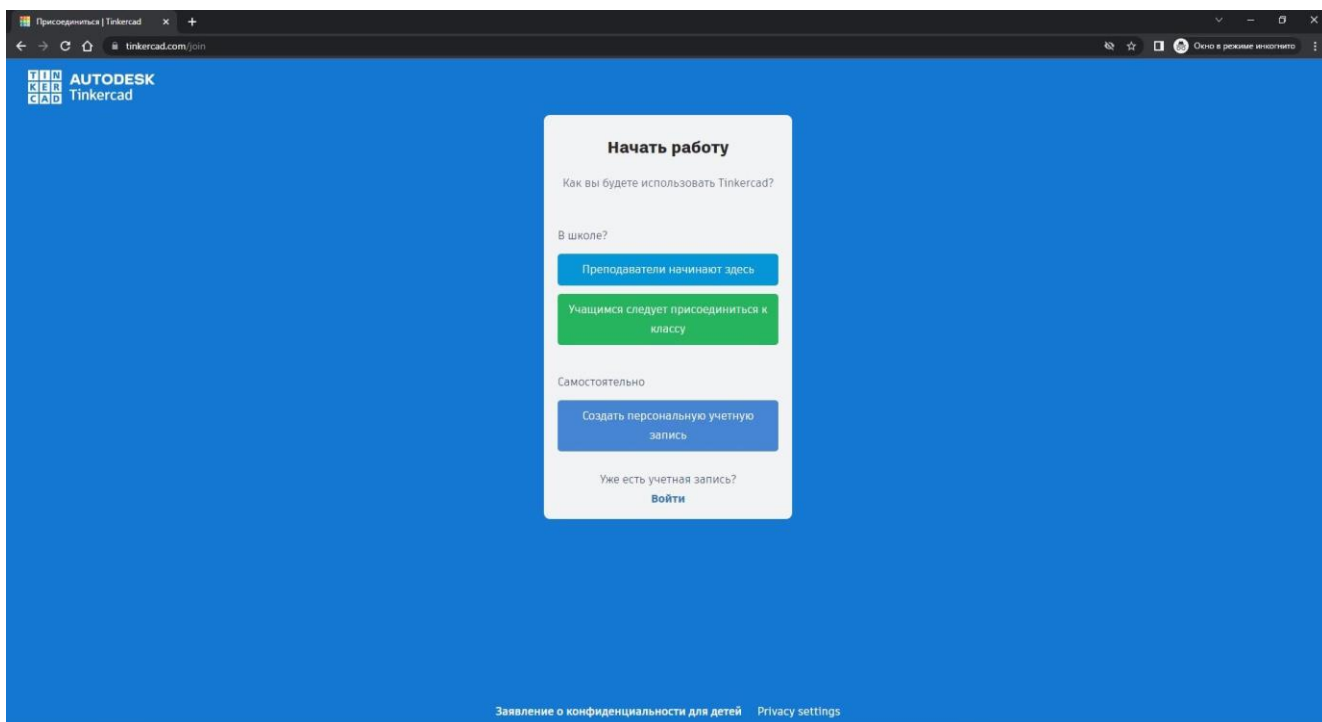
3. Платформа Tinkercad

3.1. Регистрация

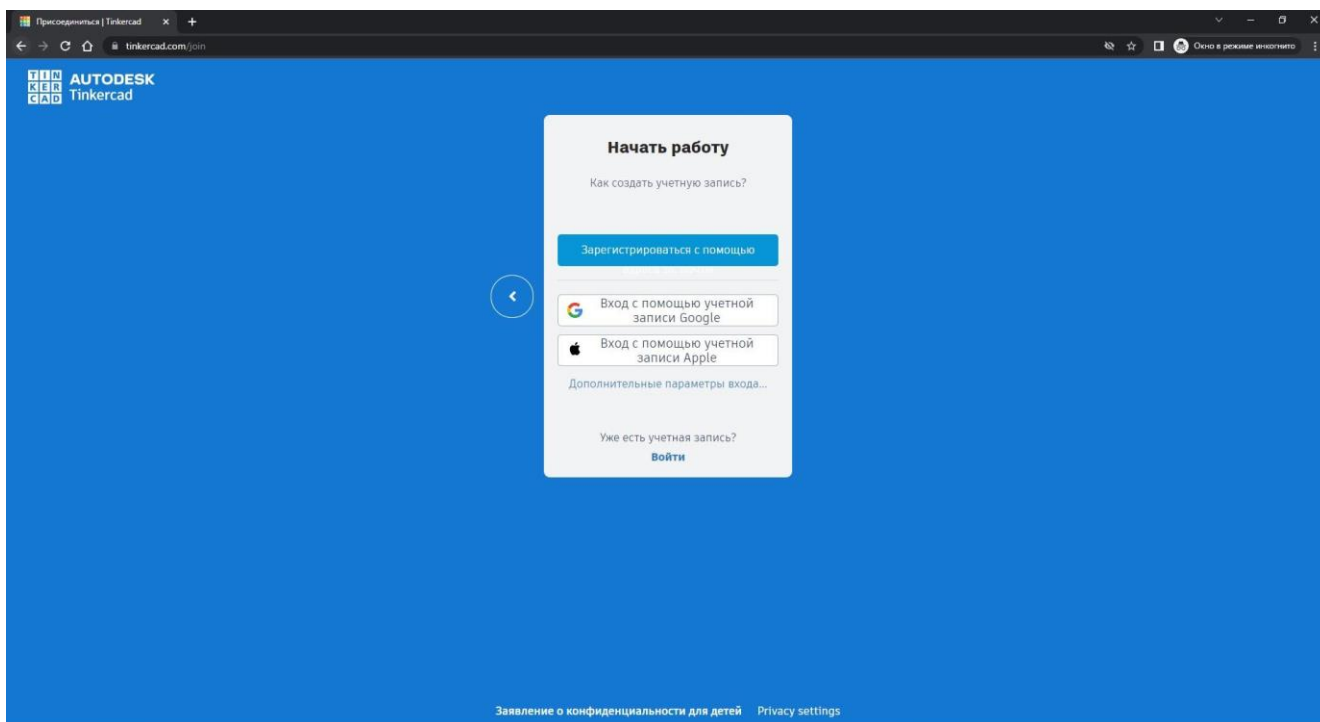
Заходим на страницу <https://www.tinkercad.com/>, нажимаем кнопку “Регистрация”.



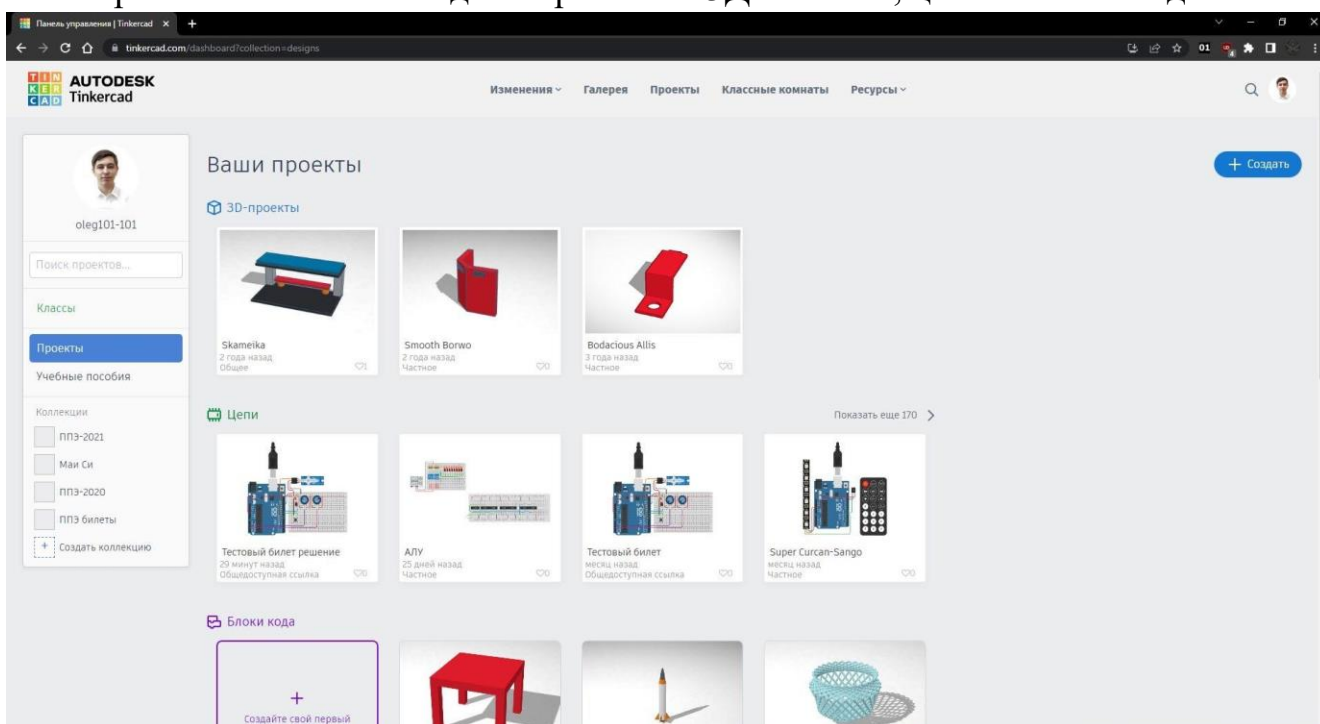
Выбираем вариант “Создать персональную учетную запись”



Можно выбрать любой из вариантов регистрации, например с помощью почты.



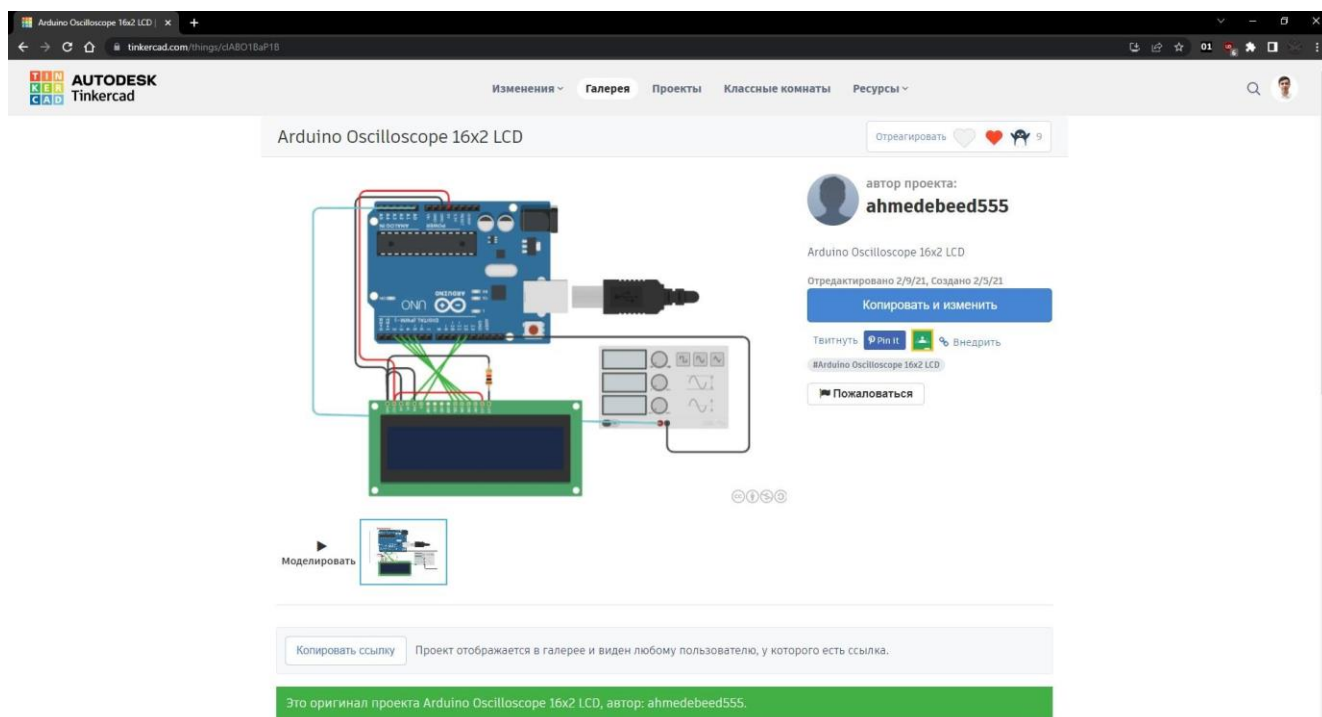
После регистрации или входа пользователь попадает на главную страницу. На ней располагаются последние проекты – 3Д объекты, цепи и блоки кода.



3.2. Копирование проектов

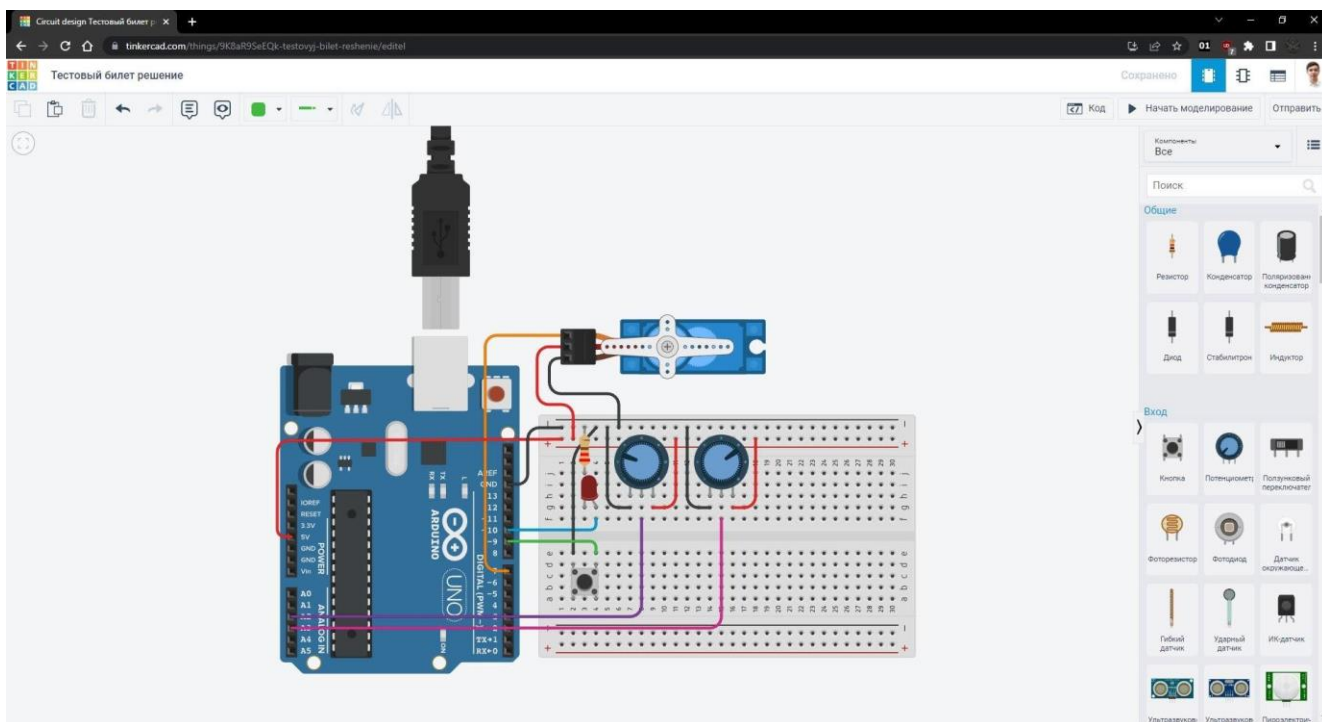
После получения ссылки на проект, или выбора проекта из галереи пользователь может запустить его, чтобы посмотреть на его работу, или

скопировать его себе для хранения или внесения изменений с помощью кнопки “Копировать и изменить”.



3.3. Изменение проектов

При работе с проектами пользователь может добавлять необходимые элементы из панели, расположенной справа, элементы достаются перетягиванием “Drag-and-drop” или нажатием левой кнопкой на элемент для выбора, переносом в рабочее пространство и повторным нажатием для расположения элемента. По умолчанию выбраны только основные компоненты, можно переключить меню для отображения всех доступных элементов.



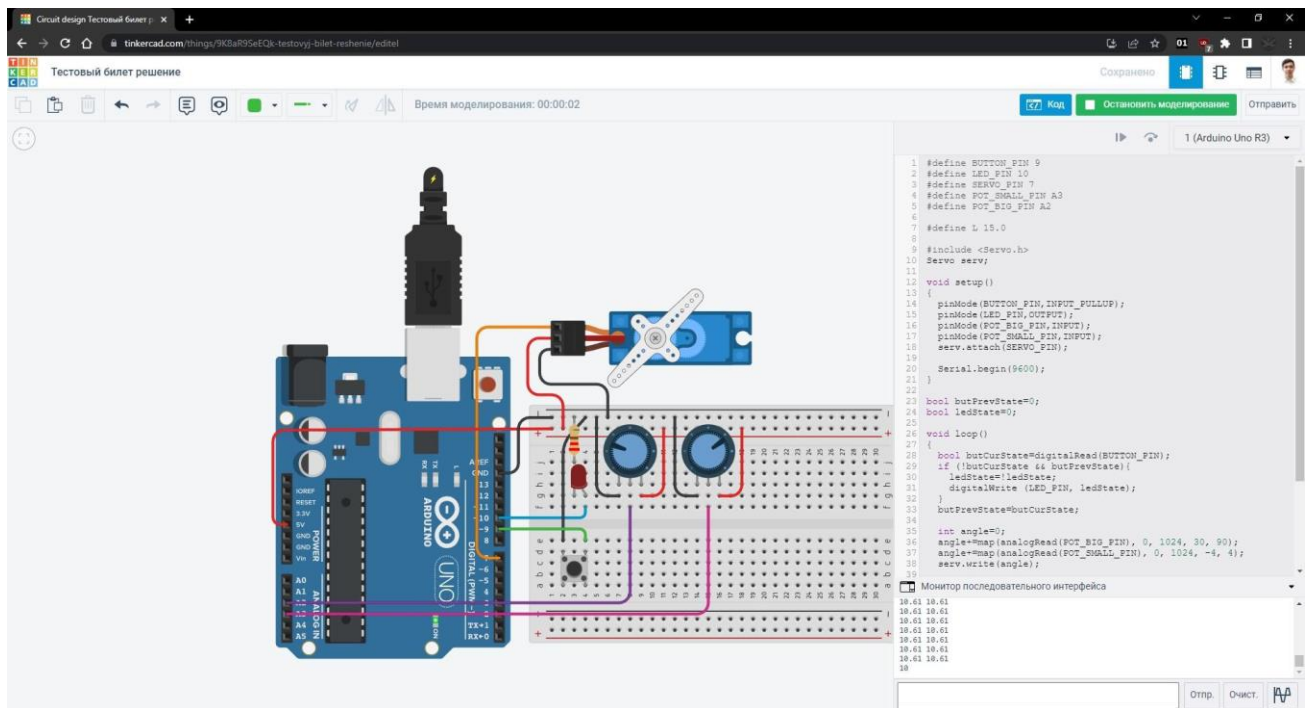
Элементы можно соединять проводами между собой, для создания провода необходимо нажать левой кнопке мыши на возможной точке старта – контакте элемента или макетной платы, после этого провод можно подвести к второй точке подключения и вновь нажать левую кнопку мыши. На проводе можно создавать точки сгиба нажатием левой кнопки мыши на части провода, после его создания.

Для написания программ для микроконтроллеров используется вкладка код, в правом верхнем углу. Tinkercad поддерживает три способа программирования – Scratch-подобными блоками, текстовой программой на языке Си и смешанный формат, при котором код на блоках также показывается в виде кода на Си.

Переключение режима находится в верхней левой части окна с программой.

3.4. Эмуляция и отладка

После создания схемы и написания программы можно провести отладку системы, запустив её с помощью кнопки “Начать моделирование” в правом верхнем углу. В режиме моделирования пользователь может взаимодействовать с контроллером через Serial интерфейс, экран расположен в нижней части блока кода. Пользователь также может взаимодействовать с различными устройствами, например нажимать на кнопки или вращать потенциометры, нажимая на них левой кнопкой мыши.



Более сложные устройства, например датчик расстояния, предоставляют собственные интерфейсы взаимодействия с ними, интерфейсы активируются при выборе устройства левой кнопкой мыши.

