

**НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ  
АГРАРНЫЙ УНИВЕРСИТЕТ**

**Инженерный институт**

**ЖУРНАЛ**

Для выполнения лабораторно-практических работ  
По дисциплине «Автоматизированные технологические линии пищевых  
производств»

***19.04.03 Продукты питания животного происхождения***, профиль  
*Продукты для функционального питания*

Ф.И.О. \_\_\_\_\_

группа \_\_\_\_\_

**Новосибирск 2023**

**Кафедра механизации животноводства и переработки  
сельскохозяйственной продукции**

Автоматизированные технологические линии пищевых производств  
Журнал лабораторно-практических работ: / Новосиб. гос. аграр. ун-т. Инж.  
ин-т; Сост.: А.А. Мезенов, Блескин С.С.. – Новосибирск, 2023 - 46 с.

Рабочий журнал предназначен для студентов по направлению **19.04.03**  
***Продукты питания животного происхождения***, профиль *Продукты для*  
*функционального питания*

Утверждено и рекомендовано к изданию методическим советом  
Инженерного института (протокол № 5 от 22 декабря 2023 г.).

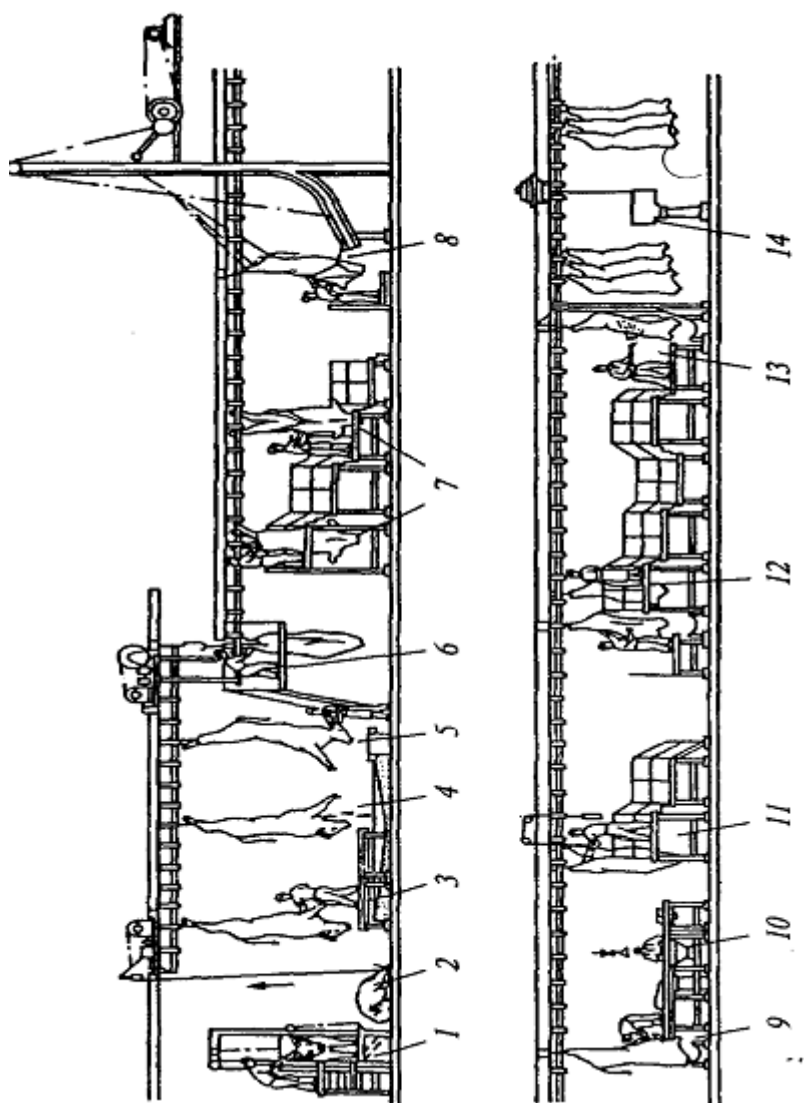
© Новосибирский государственный  
аграрный университет, 2023

Лабораторно-практическое занятие №1

По каждой из рассмотренных в журнале линий необходимо ответить на следующие вопросы:

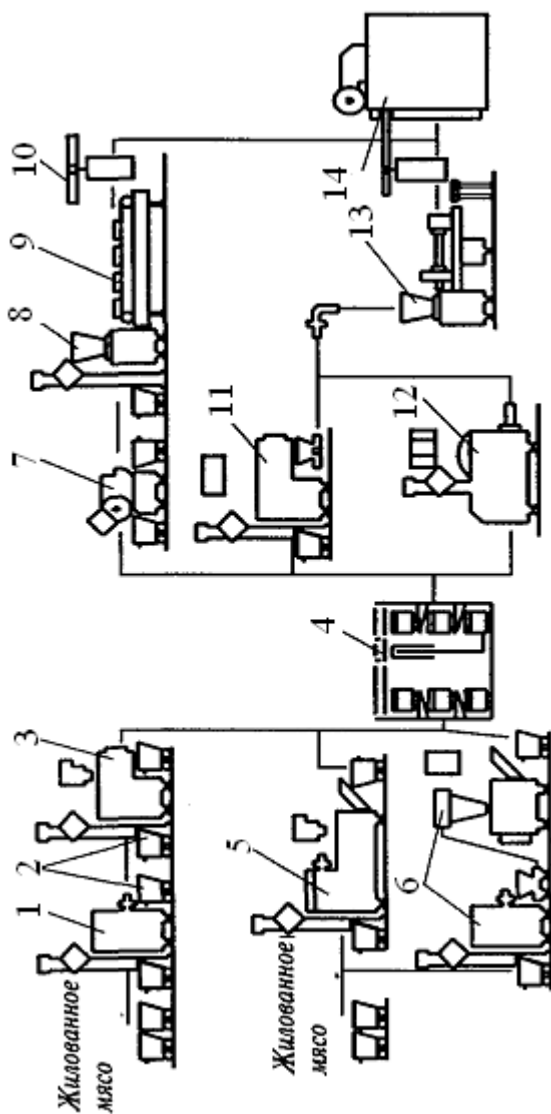
1. Для рассматриваемой поточно технологической линии заполнить таблицу классификации.
2. Разделить технологический процесс на стадии автоматизации.
3. Для выделенных стадий сформировать производственные ячейки и всю автоматизированную линию.

# ТЕХНОЛОГИЧЕСКАЯ ЛИНИЯ ПЕРВИЧНОЙ ПЕРЕРАБОТКИ КРС



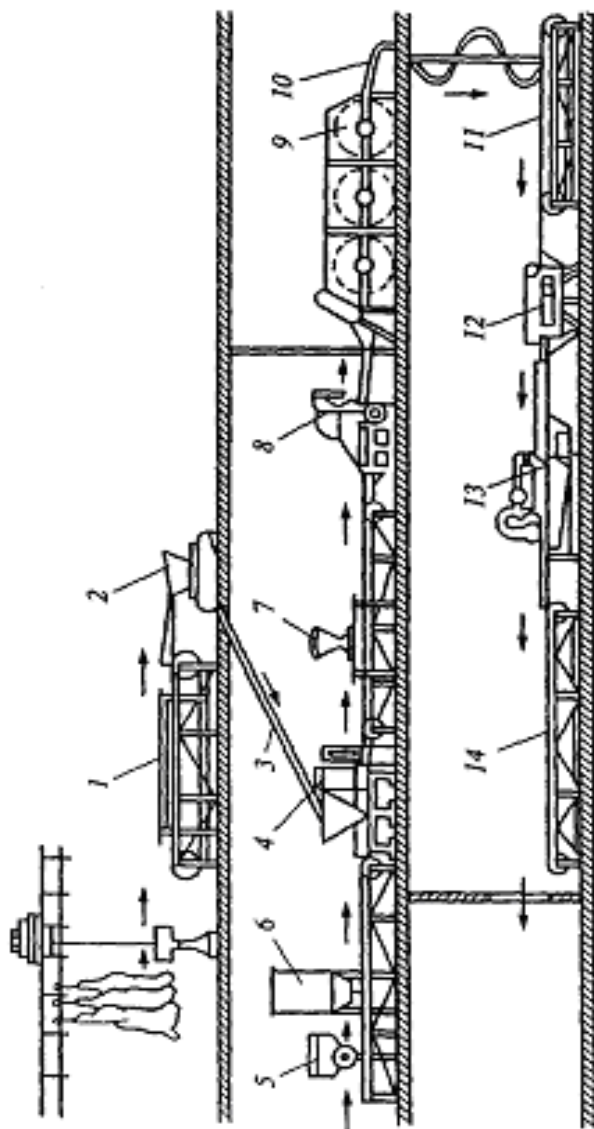
Признаки классификации	Поточные линии
Функциональное назначение	
Зависимость от номенклатуры вырабатываемых изделий	
Ритм работы	
Вид связи между машинами	
Степень механизации и автоматизации	
Структура потока	
Компоновка	

# ТЕХНОЛОГИЧЕСКАЯ ЛИНИЯ ПРОИЗВОДСТВА ВАРЕННЫХ КОЛБАС



Признаки классификации	Поточные линии
Функциональное назначение	
Зависимость от номенклатуры вырабатываемых изделий	
Ритм работы	
Вид связи между машинами	
Степень механизации и автоматизации	
Структура потока	
Компоновка	

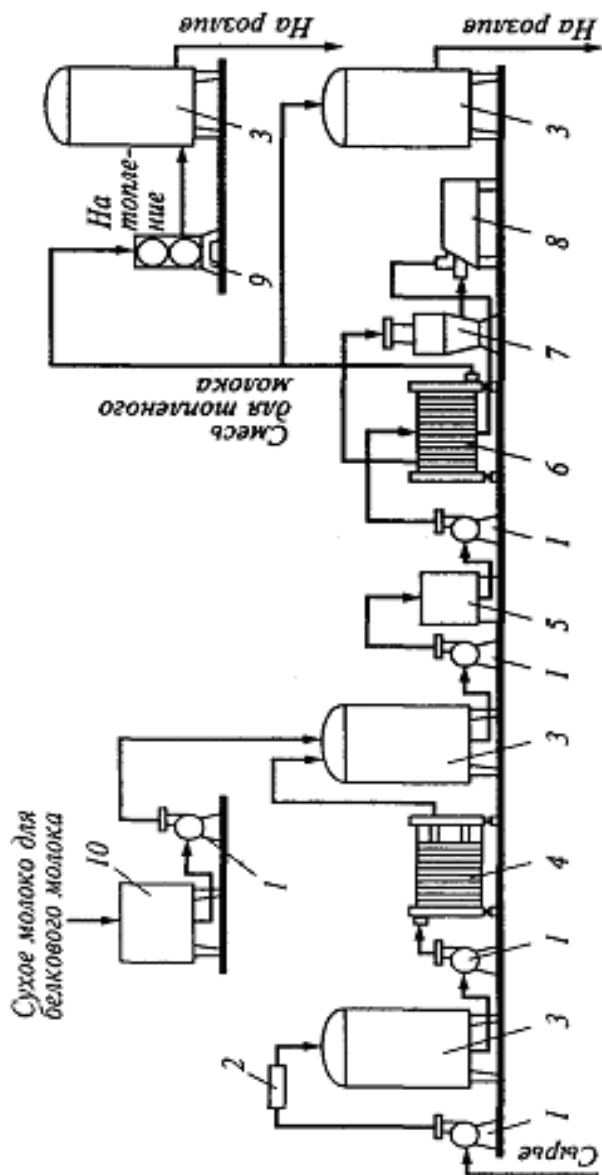
## ТЕХНОЛОГИЧЕСКАЯ ЛИНИЯ ПРОИЗВОДСТВА МЯСНЫХ КОНСЕРВОВ





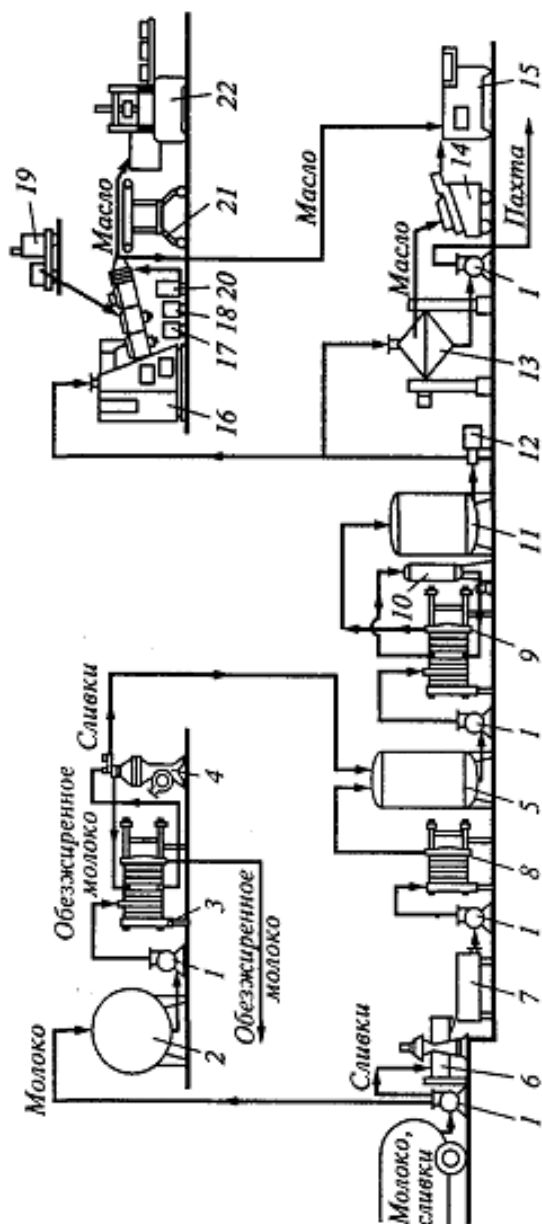
Признаки классификации	Поточные линии
Функциональное назначение	
Зависимость от номенклатуры вырабатываемых изделий	
Ритм работы	
Вид связи между машинами	
Степень механизации и автоматизации	
Структура потока	
Компоновка	

## ТЕХНОЛОГИЧЕСКАЯ ЛИНИЯ ПРОИЗВОДСТВА ПАСТЕРИЗОВАННОГО МОЛОКА



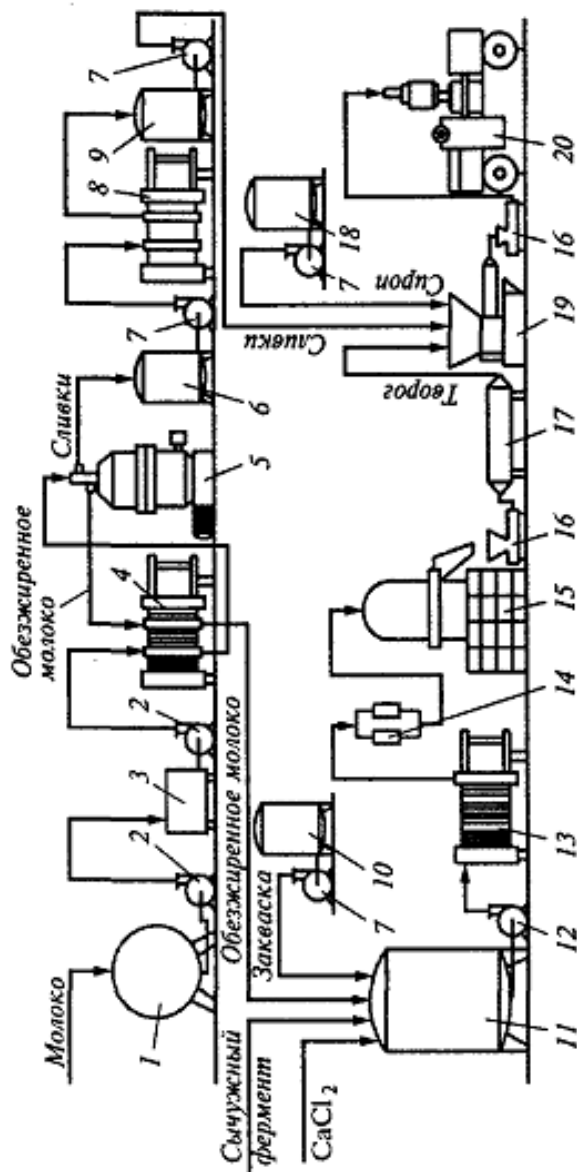
Признаки классификации	Поточные линии
Функциональное назначение	
Зависимость от номенклатуры вырабатываемых изделий	
Ритм работы	
Вид связи между машинами	
Степень механизации и автоматизации	
Структура потока	
Компоновка	

# ТЕХНОЛОГИЧЕСКАЯ ЛИНИЯ ПРОИЗВОДСТВА СЛИВОЧНОГО МАСЛА



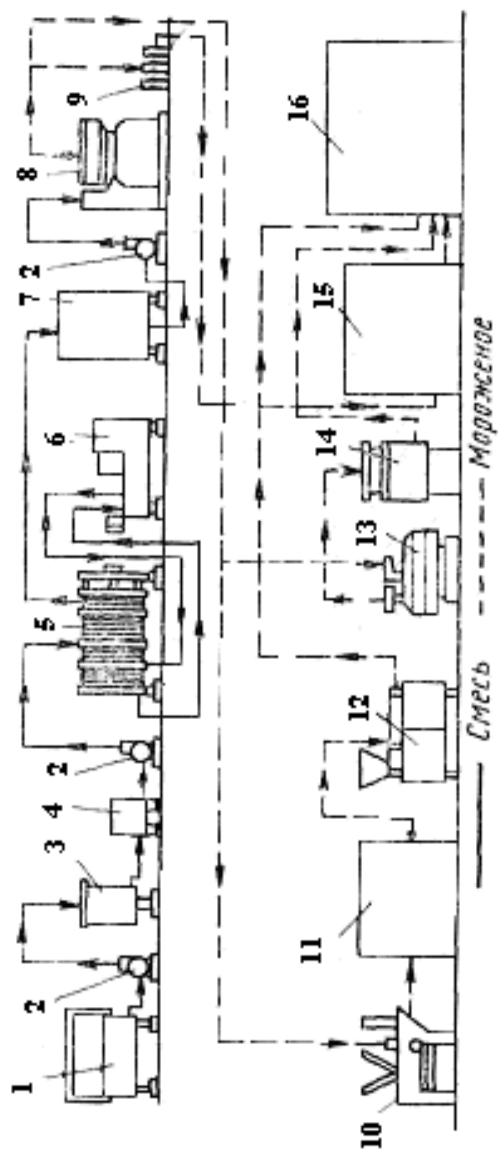
Признаки классификации	Поточные линии
Функциональное назначение	
Зависимость от номенклатуры вырабатываемых изделий	
Ритм работы	
Вид связи между машинами	
Степень механизации и автоматизации	
Структура потока	
Компоновка	

# ТЕХНОЛОГИЧЕСКАЯ ЛИНИЯ ПРОИЗВОДСТВА ТВОРОГА



Признаки классификации	Поточные линии
Функциональное назначение	
Зависимость от номенклатуры вырабатываемых изделий	
Ритм работы	
Вид связи между машинами	
Степень механизации и автоматизации	
Структура потока	
Компоновка	

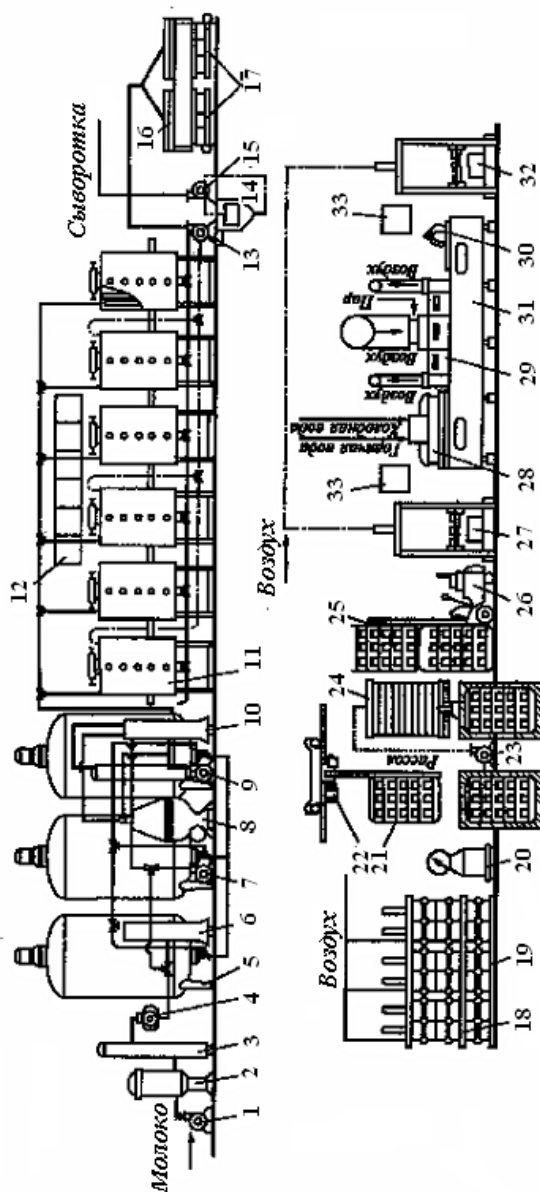
# ТЕХНОЛОГИЧЕСКАЯ ЛИНИЯ ПРОИЗВОДСТВА МОРОЖЕНОГО





Признаки классификации	Поточные линии
Функциональное назначение	
Зависимость от номенклатуры вырабатываемых изделий	
Ритм работы	
Вид связи между машинами	
Степень механизации и автоматизации	
Структура потока	
Компоновка	

## ТЕХНОЛОГИЧЕСКАЯ ЛИНИЯ ПРОИЗВОДСТВА ТВЕРДОГО СЫРА



Признаки классификации	Поточные линии
Функциональное назначение	
Зависимость от номенклатуры вырабатываемых изделий	
Ритм работы	
Вид связи между машинами	
Степень механизации и автоматизации	
Структура потока	
Компоновка	

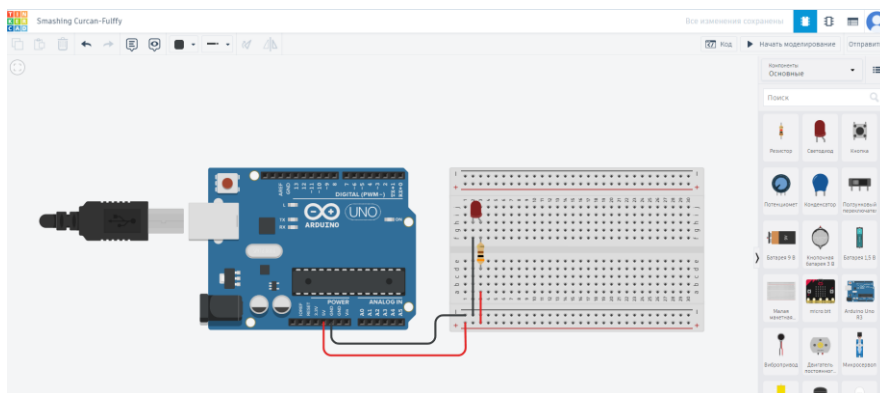
## Лабораторная работа № 2

**Название:** Работа с макетной платой устройство и использование Arduino в качестве источника питания.

**Цель:** Понять принцип работы макетной платы, а также распиновку питания Arduino.

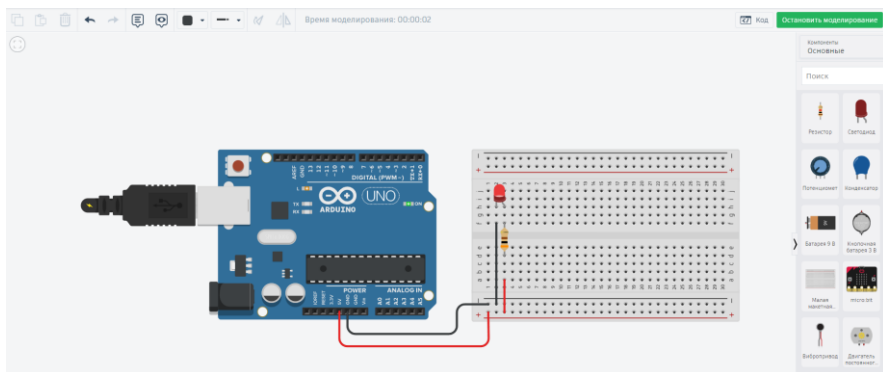
**Оборудование:** Макетная плата, светодиод, резистор, питание Arduino, плата Arduino UNO.

1. Собрать схему показанная на рисунке 13, на сайте <https://www.tinkercad.com>(если нет учетной записи необходимо зарегистрироваться) , изменить сопротивление резистора на 300 Ом, нажатием на него после установки в цепь.



*Рис. 12 – Схема светодиода*

2. Нажать на кнопку начать моделирование в верхнем правом углу, получить результат показанном на рисунке 13.



*Рис. 13 – Схема горящего светодиода*

3. Самостоятельно подключить 3 светодиода к ардуино используя макетную плату, сделать выводы по устройству макетной платы.
4. После успешного подключения повторить на лабораторных наборах и оформить в MS Office Word.

### **Защита:**

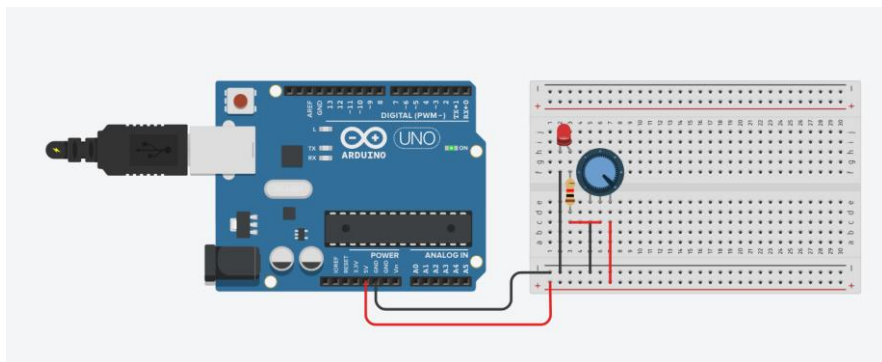
### **Лабораторная работа № 3**

**Название:** Работа с макетной платой устройство и использование Arduino в качестве источника питания с добавлением переменного резистора.

**Цель:** Понять принцип работы переменного резистора, а также научиться его подключать.

**Оборудование:** Макетная плата, светодиод, резистор, питание Arduino, плата Arduino UNO, переменный резистор.

1. Собрать схему показанная на рисунке 14, на сайте <https://www.tinkercad.com>, изменить сопротивление резистора и переменного резистора на 300 Ом, нажатием на него после установки в цепь.



*Рис. 14 – Схема светодиода*

2. Нажать на кнопку начать моделирование в верхнем правом углу, получить результат показанном на рисунке 14, перемещая положение переменного резистора получить изменения яркости светодиода.

3. Самостоятельно подключить 3 светодиода к ардуино используя макетную плату, для регулирования яркости всех трех светодиодов, сделать выводы по устройству макетной платы.

1. После успешного подключения повторить на лабораторных наборах и оформить в MS Office Word.

### **Защита:**

## **Лабораторная работа №4**

**Название:** Установка Arduino

**Цель:** научиться устанавливать Arduino IDE.

**Оборудование:** компьютер.

Ознакомиться с инструкцией по установке IDE системы Arduino на компьютер для продолжения следующих практических работ:

1. <https://wiki.iarduino.ru/page/ustanovka-nastroyka-programmnoy-obolochki-arduino-ide-dlya-windows>

После установки ознакомиться с интерфейсом ПО, выбором порта подключения платы, выбора платы

необходимой для компиляции кода, установкой библиотек, а так же основными настройками.

Оформить в MS Office Word основные моменты.

Основные строки в коде:

`void setup { }` - функция настройки платы;

`void loop { }` – бесконечно повторяющаяся; функция;

Таких строчек в каждом коде должны быть только одна, в которых выполняется команда всего один раз, настройки платы первоначальные, а во втором циклично построчно каждый раз.

**Защита:**

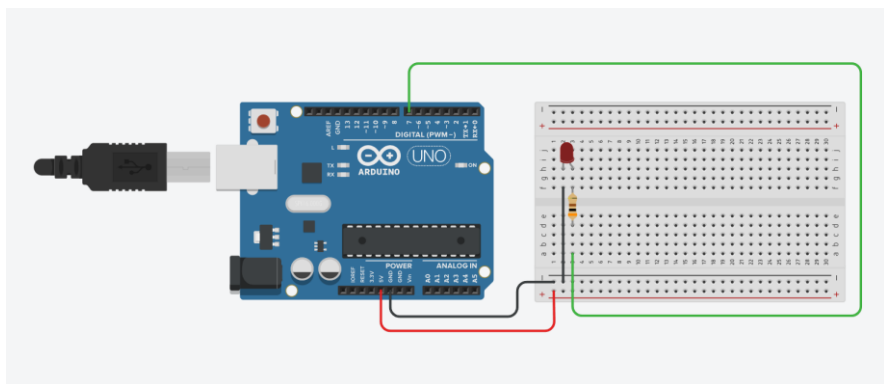
### Лабораторная работа №5

**Название:** Работа с Arduino с добавлением элементов программирования.

**Цель:** Научиться использовать блоки для создания кода, на примере включения светодиода.

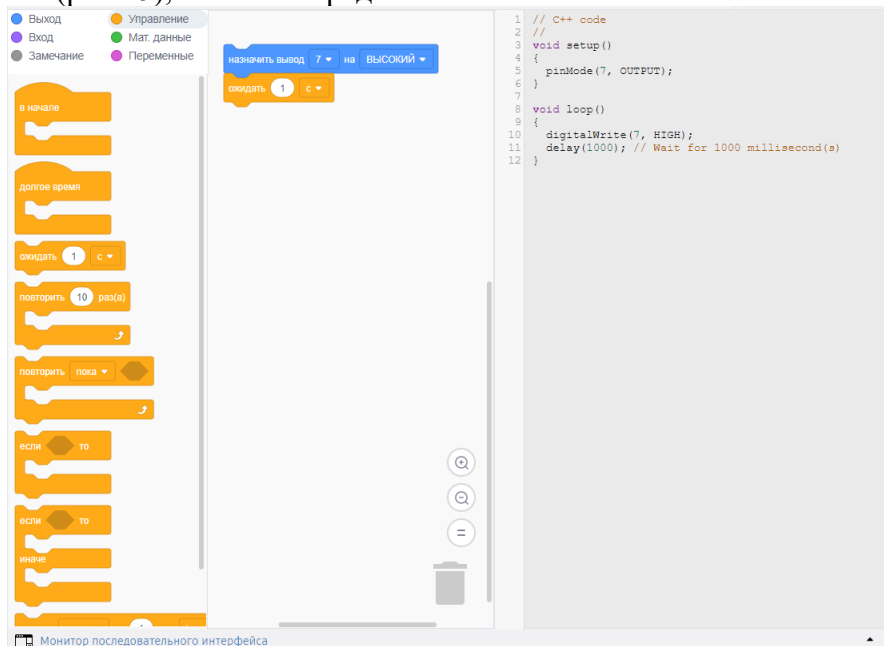
**Оборудование:** Макетная плата, светодиод, резистор, питание Arduino, плата Arduino UNO.

1. Собрать схему показанная на рисунке 15, на сайте <https://www.tinkercad.com>, изменить сопротивление резистора на 300 Ом, нажатием на него после установки в цепь.



*Рис. 15 – Схема светодиода*

2. Перейти в элемент «Код» в верхнем правом углу и выбрать в поле «блоки», поле «блоки с текстом» (рис.15), очистить среднее поле блоков



*Рис. 16 – элементы программирования*

3. Собрать схему из блоков (рис. 16), вставляя блоки прописывается код в правом крайнем окне.
4. Запустить процесс моделирование получить в результате светящийся светодиод.
5. Самостоятельно подключить 3 светодиода к портам 5,6,7, прописать схему блоков для включения светодиодов программное.
6. После успешного подключения повторить оба опыта на лабораторных наборах и оформить в MS Office Word.
7. Для загрузки кода на плату необходимо использовать ПО Arduino IDE.

## **Защита:**



### **Лабораторная работа №6**

**Название:** Работа с Arduino с добавлением элементов программирования без блоков.

**Цель:** научиться читать код с дальнейшим его изменением.

**Оборудование:** Макетная плата, светодиод, резистор, питание Arduino, плата Arduino UNO.

1. Используя полученный код в предыдущей работе, проанализировать основные строки и прописать код для 4 светодиодов, мигающих поочередно, сначала на сайте затем на лабораторных наборах.

2. На сайте для ввода кода изменить «блоки с текстом» на «текст» в верхнем правом углу.

3. Изменить код для мигания в обе стороны 4 светодиодов «волной»

4. После успешного подключения повторить оба опыта на лабораторных наборах и оформить в MS Office Word.

**Защита:**

### **Лабораторная работа №7**

**Название:** Работа с Arduino с добавлением элементов программирования с переменным резистором.

**Цель:** Научиться передать сигналы с помощью аналоговых входов на цифровые и проверять правильность схем.

**Оборудование:** Макетная плата, светодиод, резистор, кабель подключения, плата Arduino UNO, переменный резистор.

1. Собрать схему на сайте (рис. 17), проанализировать её и исправить не верно подключенный элемент.

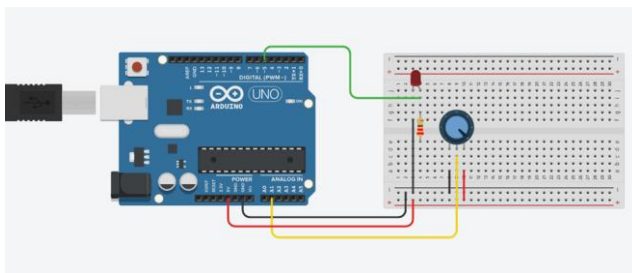


Рис.17 – переменный резистор

2. В поле «блоки с текстом» исправить на «текст» и ввести следующий код.

```
sketch_oct12a | Arduino 1.8.15
Файл Правка Скетч Инструменты Помощь

sketch_oct12a$

void setup() { // функция настройки работы платы
  pinMode(5,OUTPUT); // объявления 5 цифрового пина с ШИМ, как выход
  pinMode(A1, INPUT); // объявления аналогового пина A1, как вход
}

void loop() { // бесконечная повторяющаяся функция
  int bright; // объявляем переменную bright (яркость) как целочисленную
  bright = analogRead(A1); // присваиваем переменной bright значение, считываемое с пина A1 (потенциометра)
  bright = bright / 4; // делим значение переменной bright на 4, так как значение равно числу от 0-1023, а необходимо 0-255
  analogWrite(5,bright); // передаем на светодиод новое значение переменной bright
}
```

3. Проанализировать код, прочитав комментарии, а также основные команды:

void setup { } - функция настройки платы;  
 void loop { } – бесконечно повторяющаяся; функция;  
 pinMode(пин, OUTPUT или INPUT) – объявление режима порта;  
 digitalWrite(пин, 1 или 0) – передача цифрового сигнала;  
 analogWrite(пин, 0-255) – передача аналогового сигнала;  
 digitalRead(пин) – прием цифрового сигнала;  
 analogRead(пин) – прием аналогового сигнала;  
 delay(мс) – задержка между выполнением команд в секундах

4. После успешного подключения повторить опыт на лабораторных наборах и оформить в MS Office Word.

**Защита:**

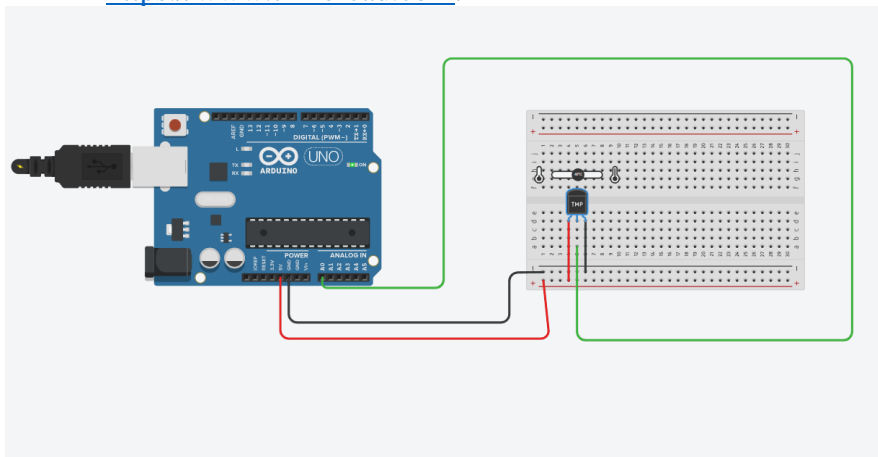
## Лабораторная работа №8

**Название:** Подключение датчика температуры к Arduino UNO.

**Цель:** научиться читать подключать датчики, на примере датчика температуры и вывод показаний в монитор порта.

**Оборудование:** Оборудование: Макетная плата, кабель подключения, плата Arduino UNO, датчик температуры.

1. Собрать схему показанная на рисунке 18, на сайте <https://www.tinkercad.com>.



*Рис. 18 – подключение датчика температуры*

2. Прописать код в окне для кода:



```
1 // C++ code
2 //
3 void setup()
4 {
5     Serial.begin(9600);
6     pinMode(A0, INPUT);
7 }
8
9
10 void loop()
11 {
12     float temp = ((analogRead(A0)*(5.0/1024))-0.5)/0.01;
13     Serial.println(temp);
14     delay(200); // Delay a little bit to improve simulation performance
15 }
```

3. Проанализировать код, основные команды и уравнение.
4. Запустить моделирование и в окне «монитор последовательного интерфейса» проследить изменение температуры изменяя ее ручным способом нажатием на датчик.



#### Монитор последовательного интерфейса

5. Посмотреть маркировку датчика температуры в лабораторных наборах, найти уравнение в глобальной сети Интернета.
6. После успешного подключения повторить опыт на лабораторных наборах и оформить в MS Office Word.

### Защита:

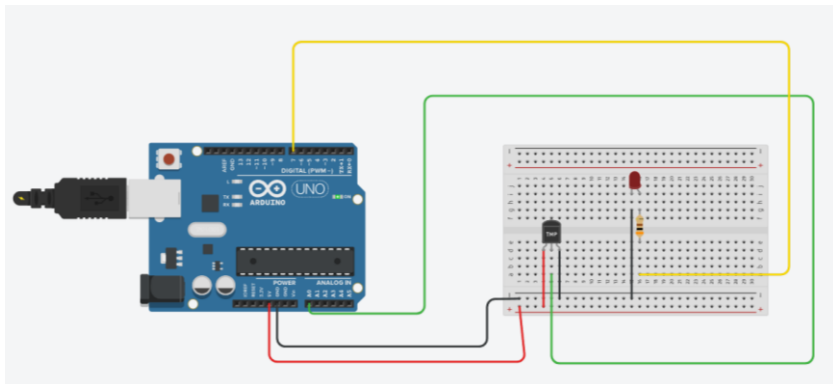
### Лабораторная работа №9

**Название:** Добавление индикаций превышение допустимой температуры.

**Цель:** Научиться логическим операциям для автоматического запуска электроприборов, на примере датчика температуры и светодиода.

**Оборудование:** Макетная плата, светодиод, резистор, кабель подключения, плата Arduino UNO, датчик температуры.

Собрать схему показанная на рисунке 19, на сайте <https://www.tinkercad.com>.



*Рис. 19 – Индикация температуры*

1. Прописать код в окне для кода:

```
1 int temp;
2 void setup()
3 {
4   Serial.begin(9600);
5   pinMode(A0, INPUT);
6   pinMode( 7, OUTPUT);
7 }
8
9 void loop()
10 {
11   float temp = ((analogRead(A0)*(5.0/1024))-0.5)/0.01;
12   Serial.println(temp);
13   delay(200); // Delay a little bit to improve simulation performance
14   if (temp > 30) {
15     digitalWrite(7, HIGH);
16   } else {
17     digitalWrite(7, LOW);
18   }
19   delay(10);
20 }
```

2. Проанализировать код, обратить внимание на функции «правда», «ложь», в коде это «if» и «else».
3. Самостоятельно добавить второй светодиод, загорающийся от опускания допустимой температуры.
4. После успешного подключения повторить опыт на лабораторных наборах и оформить в MS Office Word.

## Защита:

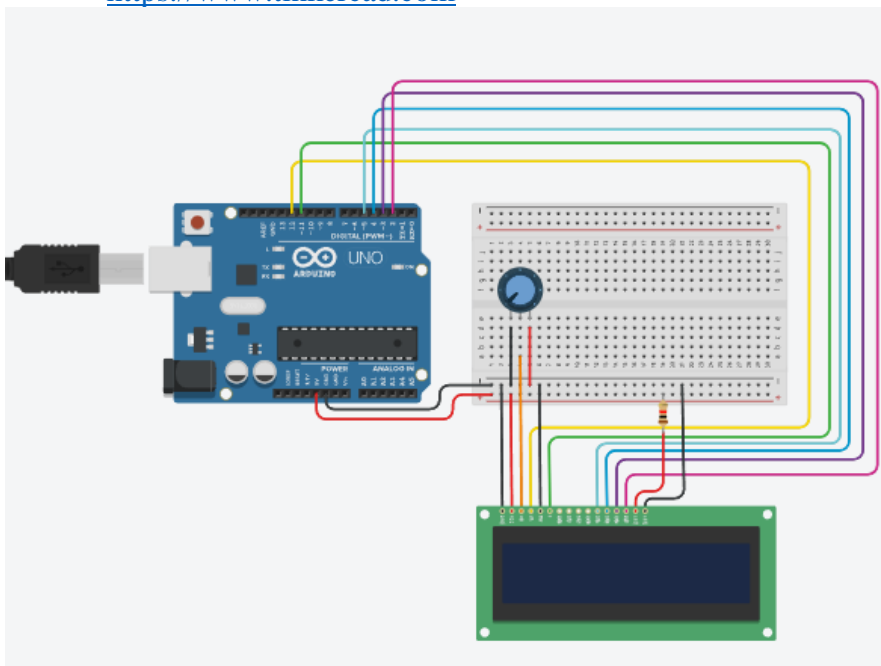
### Лабораторная работа №10

**Название:** Подключение дисплея к контролеру Arduino UNO.

**Цель:** Научиться подключать жк-дисплей и выводить на него информацию.

**Оборудование:** Макетная плата, переменный резистор, кабель подключения Arduino, плата Arduino UNO, дисплей 16x2.

1. Собрать схему показанная на рисунке 20, на сайте <https://www.tinkercad.com>



*Рис. 20 – Подключенный дисплей*

2. Прописать код в окне для ввода кода:

lesson\_01.ino

```
1  #include <LiquidCrystal.h>
2
3  LiquidCrystal lcd(12,11,5,4,3,2);
4  void setup()
5  {
6      lcd.begin(16, 2); // устанавливаем кол-во столбцов и строк
7      lcd.clear();
8      lcd.print("hello, world!"); // печать на ЖКИ сообщения
9  }
10
11 void loop()
12 {
13     lcd.setCursor(0,1);
14     lcd.print(millis()/1000);
15 }
```

3. Проанализировать код, описать что выводится на дисплей

4. Самостоятельно после анализа кода вывести на экран свое имя (на латинице) и номер группы, переместить строки ввода по центру экрана.

5. После успешного подключения повторить опыт на лабораторных наборах и оформить в MS Office Word.

## **Защита:**

### **Лабораторная работа №11**

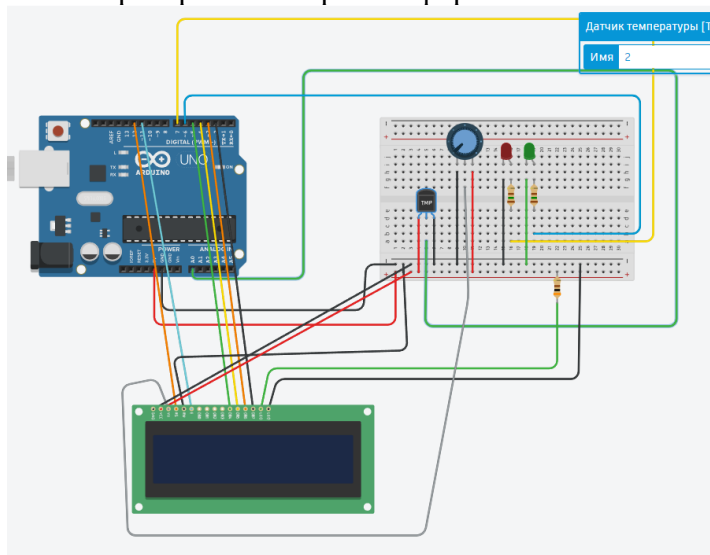
**Название:** Объединение кодов для вывода показаний температуры на экран.

**Цель:** Научиться объединять готовые коды для вывода информации на ЖК дисплей.

**Оборудование:** Макетная плата, светодиод, резистор, кабель подключения, плата Arduino UNO, датчик температуры, экран 16x2.

1. Используя схемы подключения датчика температуры (рис.19) и схему подключения ЖК дисплея (рис.20), собрать единую схему

- Используя код для чтения данных с датчика температуры и код для вывода информации на ЖК дисплей, объединить код для вывода на дисплей показаний температуры.
- После успешного подключения повторить опыт на лабораторных наборах и оформить в MS Office Word.



**Защита:**

## Лабораторная работа №12

**Название:** Добавление библиотек для чтения датчиков в ардуино на примере датчика температуры DHT11.

**Цель:** Научиться добавлять сторонние библиотеки в Arduino.

**Оборудование:** Компьютер.

1.Перейти на сайт:

[https://wiki.iarduino.ru/page/Installing\\_libraries](https://wiki.iarduino.ru/page/Installing_libraries)

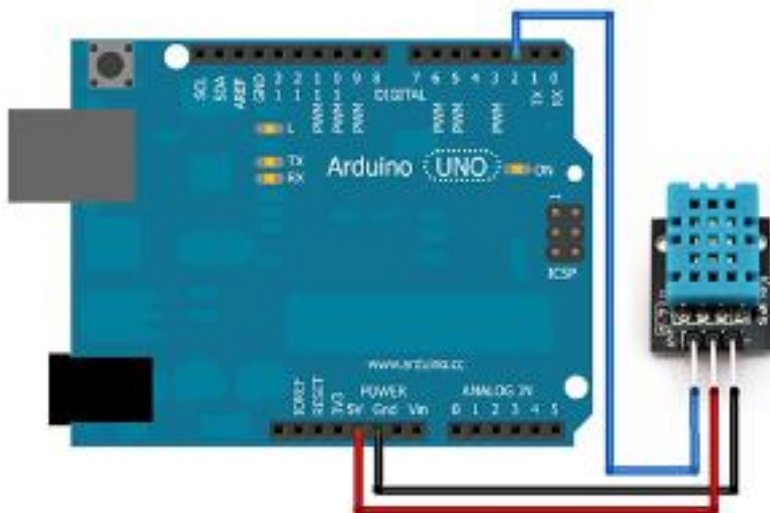
прочитать о способах установки библиотек в Ардуино.

2. Оформить в MS Office Word, основные данные, полученные из статьи.



3. Скачать и выполнить установку библиотек для датчика DHT11, с GitHub.com

4. Выполнить подключение датчика по схеме на рис.21



*Рис. 21 – Подключение DHT 11*

5.Добавить код из примера библиотек для датчика:

```
#include "DHT.h"
#define DHTPIN 2 // Тот самый номер пина, о котором
упоминалось выше
// Одна из следующих строк закомментирована. Снимите
комментарий, если подключаете датчик DHT11 к arduino
DHT dht(DHTPIN, DHT22); //Инициация датчика
//DHT dht(DHTPIN, DHT11);
void setup() {
  Serial.begin(9600);
  dht.begin();
}
void loop() {
  delay(2000); // 2 секунды задержки
  float h = dht.readHumidity(); //Измеряем влажность
  float t = dht.readTemperature(); //Измеряем температуру
  if (isnan(h) || isnan(t)) { // Проверка. Если не
  удастся считать показания, выводится «Ошибка считывания»,
  и программа завершает работу
    Serial.println("Ошибка считывания");
    return;
  }
  Serial.print("Влажность: ");
  Serial.print(h);
  Serial.print(" %\t");
  Serial.print("Температура: ");
  Serial.print(t);
  Serial.println(" *C "); //Вывод показателей на экран
}
```

5. Снять комментарий со строки для подключения именно DHT11.
6. Использую код для подключения ЖК дисплея, объединить код для вывода показаний, полученных из датчика температуры на экран (а также данных по влажности воздуха)

**Защита:**

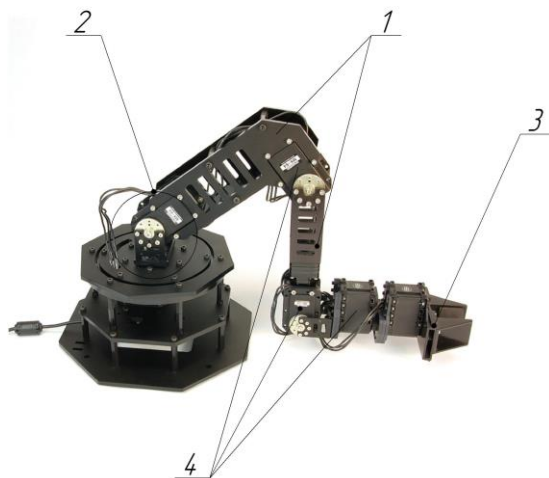
### **Лабораторная работа №13**

**Название:** Обзор конструкции робота манипулятора.

**Цель:** изучить принцип действия и конструкции робота манипулятора на примере Trossen Robotics WidowX MKII Robot Arm.

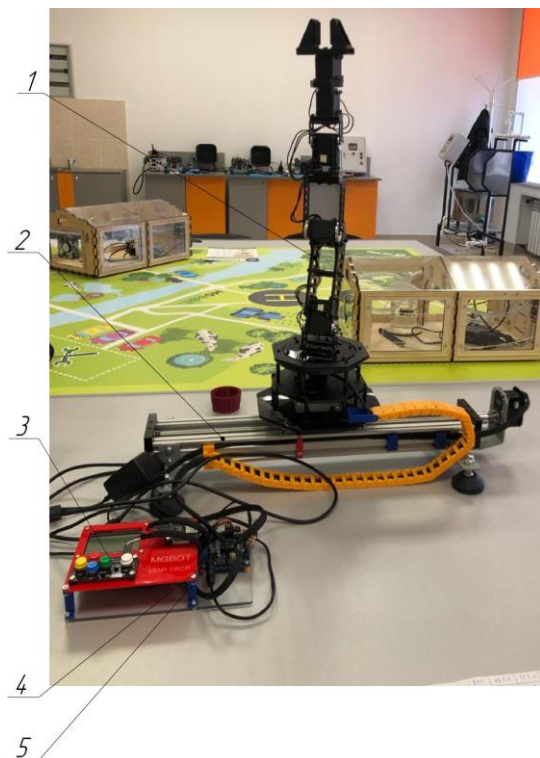
**Оборудование:** Робот манипулятор Trossen Robotics WidowX MKII Robot Arm.

1. Рассмотреть конструкцию и основные узлы робота Trossen Robotics WidowX MKII Robot Arm. (Рисунок 22)



*Рис. 22 – Trossen Robotics WidowX MKII Robot Arm*  
1) Опорные, или несущие, конструкции 2)  
Манипуляционная система 3) Рабочий орган  
манипулятора 4) Привод

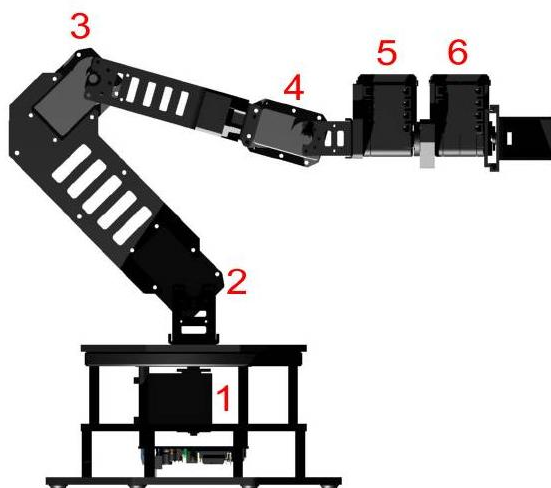
2. Определить системы робота и показать их на схеме (Рисунок 23).



*Рисунок 23 – Системы робота.*

- 1- \_\_\_\_\_
- 2- \_\_\_\_\_
- 3- \_\_\_\_\_
- 4- \_\_\_\_\_
- 5- \_\_\_\_\_

3. Изучить нахождение сервомоторов и их ограничения.



*Рисунок 24 – расположение сервомоторов.*

Для движение каждым сервомотором необходимо определить точки каждого с помощью монитора порта.

Максимальные ограничивающие точки для каждой оси:  
SERVO LIMITS :

1 : 10 - 3950

2 : 1000 - 2600

3 : 1000 - 3000 (в сторону 1000 сгибание к 2ой оси.

4 : 800 - 3200

5 : 100 - 1000

6 : 250 - 750 (захват)

4. Изучить основные команды для перемещения робота манипулятора.

SetPosition (X , Y); -Движение Приводов Робота

Где X-номер сервопривода

Y- координата перемещения (учитывая лимиты перемещения)

Set\_EXT7\_position(X); - Движение 7ой осью

Где X- процентное соотношение перемещение оси (рис.25).

Для возврата робота в исходное положение:  
`MoveExt7_HomePos();`



*Рисунок 25 – Схема 7ой оси.*

**ВАЖНО** (для работы 7ой оси, каждый запуск робота, необходимо проводить её калибровку)

5. Оформить в MS Office Word, основные данные, полученные из статьи

**Защита:**

### **Лабораторная работа №14**

**Название:** Изучение строения линейного привода на примере шарико-винтовой передачи.

**Цель работы:** ознакомиться с принципом работы ШВП, научиться работать с ШВП.

**Оборудование:** Робот манипулятор Trossen Robotics WidowX MKII Robot Arm.

**Теория:**

Шарико-винтовая передача (ШВП) применяется для передачи усилия и преобразования вращения в поступательное движение. Устройство имеет несколько элементов: стержень с винтовой канавкой, гайка, подходящая по резьбе и размерам.

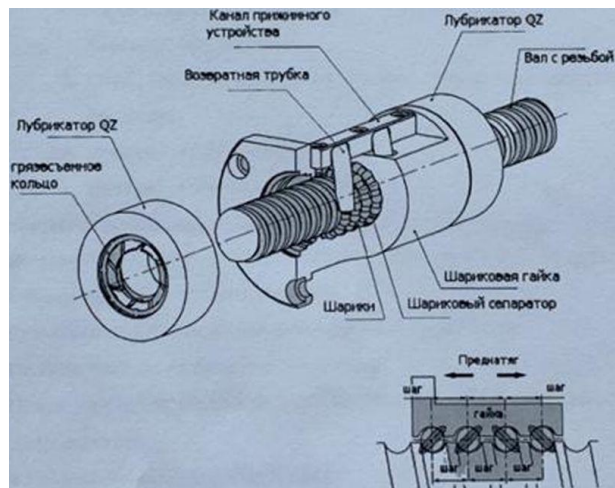


Рисунок №24 Структурная схема ШВП

ШВП имеет простую конструкцию и работает следующим способом:

1. На момент вращения гайки шарики катаются по имеющимся каналам.
2. Шарики могут поступательно перемещать гайку, выталкивая из резьбы. По перепускному каналу происходит возвращение шариков в исходное положение.

Имеются различные варианты исполнения конструкции в зависимости от условий эксплуатации и предназначения.

ШВП имеет следующие основные характеристики:  
Протяженность стержня.

- 1) Протяженность стержня
- 2) Линейное скоростное передвижение.
- 3) Диаметр и шаг винта.

Преимущества ШВП:

- 1) Низкий коэффициент трения.
- 2) Высокое значение КПД.
- 3) Отсутствие скольжения по причине применения шариков с канавками.
- 4) Простота обслуживания.
- 5) Высокая скорость перемещения.

### Ход работы:

- 1) Внимательно изучите каждый элемент ШВП. Определите названия каждого из элементов ШВП.
- 2) Внимательно изучите встроенный в рабочий стенд ШВП.
- 3) Подайте питание на испытательный стенд.
- 4) Перейдите в меню управления ШВП, нажав «SW1».
- 5) Нажав на кнопку «SW4» проведите калибровку стенда.
- 6) Нажатием на клавиши «SW2» «SW3», убедитесь, что система ШВП двигается
- 7) Сделайте выводы



Рисунок №24. Панель управления ШВП

### Содержание отчета

Отчет по лабораторной работе содержит следующие пункты:

1. Краткое описание теории;
2. Порядок выполнения и результат измерений;
3. Ответ на контрольные вопросы.

### Контрольные вопросы:

1. Принцип работы ШВП.



## 2.Области применения ШВП.

### Защита:

### Лабораторная работа № 15

**Название:** Настройка библиотек и конфигурации для запуска примеров с контроллером Arbotix-M в Arduino IDE.

**Цель:** научиться добавлять конфигурации контролеров в Arduino.

**Оборудование:** Компьютер, контроллер.

#### Ход работы:

1) Для начала необходимо скачать Arduino IDE по ссылке: <https://www.arduino.cc/en/main/software> (Ардуино установлена)

2) Скачать и распаковать из архива примеры скетчей для

контроллера Arbotix-M по ссылке:

<https://github.com/vanadiumlabs/arbotix>

3) Скачать настройки платы и библиотеки для контроллера Arbotix-M и следовать инструкции по установке:

<https://github.com/palashn/arbotix>

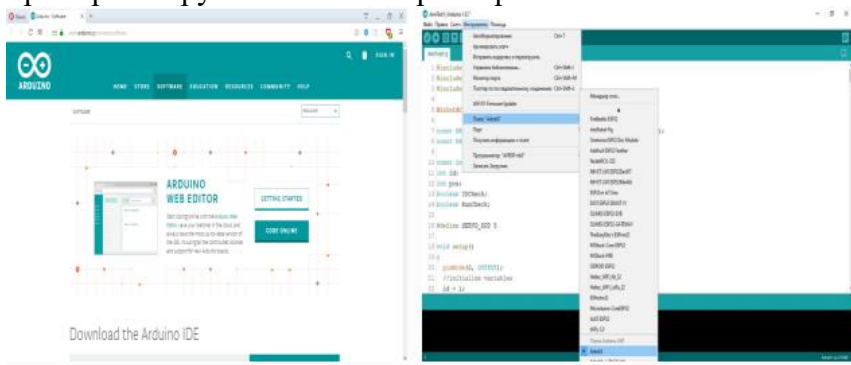
4) Если в списке плат Arduino IDE не появляется контроллер Arbotix или возникают ошибки компиляции, то необходимо скачать и установить архив по ссылке:

[https://github.com/vrxfile/Arbotix\\_V1](https://github.com/vrxfile/Arbotix_V1)

Здесь же обновленный рабочий пример для манипулятора. Распакованный архив необходимо скопировать в

C:\Users\Пользователь\Documents\Arduino\Hardware.  
Распаковка архива по принципу - «Распаковать здесь» без вложенных папок с одинаковыми именам одна в другую.

5) В Arduino IDE выбрать плату Arbotix, открыть пример и загрузить его в контроллер:



**Защита:**

## Лабораторная работа № 16

**Название:** Изучение способа программирования линейного привода.

**Цель работы:** Ознакомление с принципом программирования линейного привода.

**Оборудование:** Робот манипулятор Trossen Robotics WidowX MKII Robot Arm.

**Теория:**

Линейный привод — это привод, который создает движение по прямой линии, в отличие от кругового движения обычного электродвигателя.

Линейные приводы используются в станках и промышленном оборудовании, в компьютерных периферийных устройствах, таких как дисководы и принтеры, в клапанах и заслонках, а также во многих других местах, где требуется линейное перемещение. Гидравлические или пневматические цилиндры по своей сути создают линейное движение.

Для управления используется два скетч-файла:

ArmLearn6.ino и Ext7Control.ino, оба они работают в рамках одного проекта.

ArmLearn6.ino - содержит главный цикл, отрисовка меню для работы с роботом и с 7ой осью.

Прямое управление осью осуществляется с помощью методов в скетч-файле Ext7Control.ino

### **Описание методов :**

ResetExt7() – используется для автоматической калибровки, обнуления положения сервопривода оси.

**ВАЖНО:** перед использованием метода нужно убедиться что площадка находится физически посередине диапазона движения оси, чтобы достигать крайних положений без возможности столкновения.

*Примечание:* методы описанные ниже – не работают без проведения калибровки, после запуска необходимо выполнить ее в первую очередь.

MoveExt7\_HomePos() – используется для возврата оси в домашнее положение.

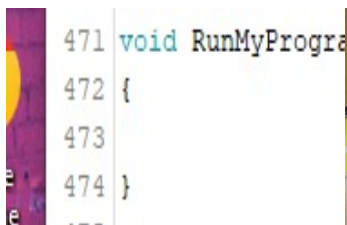
Set\_EXT7\_position() – метод для изменения положения оси в диапазоне от 0 до 100%. В качестве параметра выступает процент от максимального и минимального положения, например 0 – нижнее положение, 50 – середина, 100 – верхнее положение, диапазон значений от 0 до 100, также доступны дробные значения, например 0.001 или 99.99, актуально использовать значения в пределах тысячных долей.

*Примечание:* если дальнейшее движение невозможно (достигнут программный предел максимального или минимального положения оси) будет выведено сообщение о невозможности движения.

### **Ход работы:**

1. Подключить питание 12В к плате и USB к компьютеру;
2. Дожидаемся загрузки контроллера и отрисовки меню;
3. Нажимаем SW1 (желтая кнопка), будет выведено меню управления 7ой осью;

4. Убедитесь, что стенд находится по центру оси (рисунок 46);
5. Нажимаем SW4 – калибровка оси и ждем ее завершения;
6. Запустите в Arduino IDE скетч файл ArmLearn6.ino и Ext7Control.ino;
7. Для удобства и безопасности - не изменяйте код в Ext7Control;
8. В коде написанного в ArmLearn6.ino перейти к 472 строке (рисунок 4а);
9. Пользуясь командами напишите в 473 строку напишите команды для движения согласно вашему варианту;
10. Загрузите код и запустите движение оси;
11. Сделайте выводы

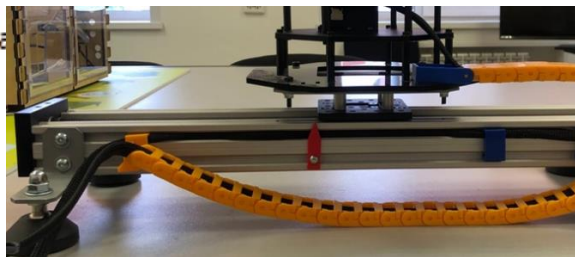


```

471 void RunMyProgram
472 {
473
474 }

```

а



б

Рисунок 25 – управление линейным приводом

Варианты заданий для обучающихся.

№ варианта	Описание движения
1	Движение на 20 процентов задержка 2с, возврат в начальное положение задержка 1с, движение на 80 процентов задержка 2с, возврат в начальное положение.
2	Движение на 14 процентов задержка 1,5 с, возврат в начальное положение задержка 1,5с, движение на 70 процентов задержка 2,5 сек,

	возврат в начальное положение.
3	Движение на 30 процентов задержка 2с, возврат в начальное положение задержка 1,5с, движение на 70 процентов задержка 2с, возврат в начальное положение.
4	Движение на 85 процентов задержка 2 с, возврат в начальное положение задержка 0,5с, движение на 13 процентов задержка 1,5 сек, возврат в начальное положение.
5	Движение на 90 процентов задержка 1,5 с, возврат в начальное положение задержка 1,5с, движение на 10 процентов задержка 1,5 сек, возврат в начальное положение.
6	Движение на 63 процентов задержка 0,5 с, возврат в начальное положение задержка 2,5с, движение на 26 процентов задержка 0,5 сек, возврат в начальное положение.
7	Движение на 16 процентов задержка 2,5 с, возврат в начальное положение задержка 0,5с, движение на 67 процентов задержка 1,5 сек, возврат в начальное положение.
8	Движение на 30 процентов задержка 1,5 с, возврат в начальное положение задержка 1 с, движение влево на 69 процентов задержка 2 сек, возврат в начальное положение.
9	Движение на 80 процентов задержка 0,5 с, возврат в начальное положение задержка 2,5с, движение на 5 процентов задержка 2 сек, возврат в начальное положение.
10	Движение на 14 процентов задержка 1,5 с, возврат в начальное положение задержка 1,5с, движение в право на 72 процентов задержка 2,5 сек, возврат в начальное положение.

**Защита:**

