

Министерство науки и высшего образования Российской Федерации

**Федеральное государственное бюджетное образовательное учреждение
высшего образования**

**«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ» (ТУСУР)**

Заочный и вечерний факультет (ЗиВФ)

А. В. Коновальчиков, Д. А. Кондрашов, К. В. Савенко

ПРОГРАММИРОВАНИЕ ВСТРАИВАЕМЫХ СИСТЕМ

Методические указания

**по изучению дисциплины и организации самостоятельной работы для
студентов, обучающихся на заочной или вечерней форме обучения**

Томск 2022

Корректор: А. Н. Миронова

А. В. Коновальчиков, Д. А. Кондрашов, К. В. Савенко

Программирование встраиваемых систем: методические указания по изучению дисциплины и организации самостоятельной работы для студентов, обучающихся на заочной или вечерней форме обучения / А. В. Коновальчиков., Д. А. Кондрашов, К. В. Савенко – Томск: ЗиВФ, ТУСУР, 2022. – 18 с.

© Коновальчиков А. В.,
Кондрашов Д. А.,
Савенко К. В., 2022
© Оформление.
ЗиВФ, ТУСУР, 2022

ОГЛАВЛЕНИЕ

Введение	4
1 Общая характеристика изучаемой дисциплины	5
2 План изучения дисциплины	7
3 Примеры тестовых заданий для зачета	11
Литература	12
Приложение А (справочное) Начало работы в среде Tinkercad.....	13

ВВЕДЕНИЕ

Внедрение автоматизации необходимо для уменьшения присутствия человека или полного его исключения из ряда процессов, которые не нуждаются в его контроле. Одним из примеров такой автоматизации является «умный дом», в котором регулированием температуры комнаты, яркости освещения, нагревом воды, открытием / закрытием окон управляют встраиваемые системы.

Навык программирования является ценным сам по себе, так как развивает логику, последовательность мышления, умение разбивать задачи на подзадачи и комбинировать различные подходы к компоновке конечного решения на основе частичных решений.

Программирование встраиваемых систем требует знания основ схемотехники, т. е. понимания простейших элементов электрической схемы, работы электрической цепи в целом, законов и правил, действующих в этой цепи, назначения простейших элементов схемы и эффектов от комбинации этих элементов.

Обладая основами понимая программирования и схемотехники, инженер способен сконструировать и запрограммировать такую систему, которая будет делать то, что запланировано.

В курсе, к изучению которого вы приступаете, закладываются основы, необходимые для развития специалиста в области программирования встраиваемых систем.

1 ОБЩАЯ ХАРАКТЕРИСТИКА ИЗУЧАЕМОЙ ДИСЦИПЛИНЫ

Цели дисциплины:

- формирование системы знаний о программировании встраиваемых систем в целом и в среде разработки Tinkercad в частности;
- получение первичных навыков программирования на языке C/C++;
- получение первичных навыков в проектировании простых схем встраиваемых систем.

Задачи дисциплины:

- усвоение основ программирования;
- изучение общих понятий программирования встраиваемых систем: синтаксис языка программирования C/C++;
- усвоение основ схемотехники.

Требования к результатам освоения дисциплины

Процесс изучения дисциплины направлен на формирование компетенций, заключающихся в способности:

- составлять схемы в среде Tinkercad в соответствии с поставленной задачей;
- создавать алгоритмы, направленные на решение задач;
- программировать алгоритмические решения на языке C/C++;
- анализировать готовые решения для существующих задач.

В результате изучения дисциплины обучающийся должен:

- **знать** общую характеристику основ программирования встраиваемых систем, базовые понятия синтаксиса языка программирования C/C++, базовые понятия схемотехники встраиваемых систем;

- **уметь** применять на практике полученные знания по программированию встраиваемых систем, составлять структурные схемы решений поставленных задач и реализовывать их в среде Tinkercad, разрабатывать, проектировать, исследовать и эксплуатировать радиоэлектронные средства и технологии, обеспечивающие передачу, обработку и прием информации по сетям радиосвязи различного назначения;
- **владеть** навыками самостоятельной работы со средой разработки Tinkercad, умением составлять алгоритмы решения задач, умением составлять схемы, реализующие поставленную задачу, с использованием компонентов среды Tinkercad.

2 ПЛАН ИЗУЧЕНИЯ ДИСЦИПЛИНЫ

Тема 1. Введение. Краткое введение в Arduino

Источник: [1], разделы 1.1, 1.2.

Понятие микроконтроллера. Устройство микроконтроллера. Введение в программирование для платформы Arduino. Что такое Arduino. Языки Processing, Wiring. Установка Arduino IDE.

Тема 2. Конструкции языка Wiring

Источник: [1], раздел 1.3.

Основные конструкции языка программирования Wiring. Этапы исполнения программы. Константы и переменные. Функции и процедуры. Синтаксис. Типы данных и функции обработки этих типов. Объявления переменных. Массивы. Операторы и циклы.

Тема 3. Начало работы в среде Tinkercad

Настройка среды Tinkercad. Описание основных шагов настройки среды Tinkercad приведено в приложении А.

Тема 4. Структура программы для Arduino

Источник: [1], раздел 1.4.

Основные структурные блоки программы. Файлы с расширением *.ino*. Ключевые слова `#include` и `#define`. Объявление глобальных переменных. Назначение функций `setup` и `loop`.

Тема 5. Порты ввода/вывода

Источник: [1], раздел 1.5.

Конфигурация портов ввода/вывода. Цифровые порты. Аналоговый ввод/вывод. Дополнительные функции ввода/вывода. Функции `digitalRead`, `digitalWrite`, `analogRead`, `analogWrite`.

Тема 6. Функции стандартной библиотеки

Источник: [1], раздел 1.6.

Функции работы со временем. Математические функции. Тригонометрические функции. Генераторы случайных значений. Операции с битами и байтами. Функции `delay`, `min`, `max`, `abs`, `random`.

Тема 7. Внешние прерывания

Источник: [1], раздел 1.7.

Понятие прерывания и его свойства. Обработчик прерываний. Функции `attachInterrupt` и `detachInterrupt`. Условие прерывания.

Тема 8. Многозадачность Arduino

Источник: [1], раздел 1.9.

Понятие многозадачности. Параллельность и последовательность исполнения инструкций. Функция `millis`. Библиотека `Servo`.

Тема 9. Управление светодиодом

Источник: [2], задача 1.

Применение полученных теоретических знаний на примере программы управления светодиодом в среде `Tinkercad`. Применение функций `setup` и `loop`.

Тема 10. Обработка нажатия кнопки

Источник: [2], задача 2.

Подключение кнопки к входу. Конфигурация контактов. Применение функций `digitalWrite` и `digitalRead`.

Тема 11. Регулирование яркости светодиода

Источник: [2], задачи 3 и 4.

Подключение потенциометра к плате. Конфигурация потенциометра. Чтение данных потенциометра и его использование для манипуляций яркостью светодиода.

Тема 12. Широтно-импульсная модуляция.

Вывод аналоговых данных

Источник: [2], задача 5.

Понятие ШИМ и RGB-светодиода. Конфигурация RGB-светодиода. Написание и применение собственных вспомогательных функций.

Тема 13. Семисегментный светодиодный индикатор

Источник: [2], задача 6.

Обзор семисегментного светодиодного индикатора. Маркировка светодиодных контактов. Отображение цифр на семисегментном индикаторе. Принцип формирования байт-записи цифр на экране. Битовые операции языка Arduino.

Тема 14. Матрица из светодиодных индикаторов

Источник: [2], задача 7.

Обзор четырехразрядной матрицы из семисегментных светодиодных индикаторов. Понятие динамической индикации. Подключение матрицы к плате. Одновременный вывод цифр. Применение Arduino-функции `millis()`.

Тема 15. Фоторезистор. Обработка освещенности

Источник: [2], задача 13.

Обзор фоторезистора. Назначение фоторезистора. Принцип действия фоторезистора. Схема подключения фоторезистора. Применение функции `map()`. Применение функции `constrain()`.

Тема 16. Индикатор LCD. Вывод информации на ЖК-индикатор

Источник: [2], задача 15.

Обзор жидкокристаллического (ЖК) индикатора Winstar. Принцип действия ЖК-индикатора. Схема подключения ЖК-индикатора. Назначение выводов контроллера ЖК-индикатора. Библиотека `LiquidCrystal`. Функции библиотеки `LiquidCrystal`.

Тема 17. Датчик температуры DS18B20

Источник: [2], задача 20.

Знакомство с цифровым термометром DS18B20. Шина 1-Wire. Возможности датчика температуры DS18B20. Схема подключения датчика температуры. Конфигурация датчика температуры. Чтение данных с датчика и передача данных на экран. Использование библиотеки `OneWire`.

Тема 18. Датчик влажности и температуры DHT11

Источник: [2], задача 21.

Знакомство с датчиком влажности и температуры DHT11. Шина 1-Wire. Возможности датчика температуры DHT11. Схема подключения датчика DHT11. Конфигурация датчика DHT11. Чтение данных с датчика и передача данных на экран. Использование библиотеки `DHT`.

3 ПРИМЕРЫ ТЕСТОВЫХ ЗАДАНИЙ ДЛЯ ЗАЧЕТА

1. Что такое компьютерная программа?

- Набор команд, понятных процессору.
- Процесс выполнения известного заранее алгоритма.
- Перевод команд языка программирования на машинный язык.

2. Где хранятся данные во время выполнения программы?

- На жестком диске.
- В оперативной памяти.
- На съемном диске.
- В процессоре.

3. Встраиваемая система – это ...

- совокупность процессора, памяти и периферийных устройств, сконфигурированных для выполнения задачи.
- набор файлов с программным кодом, сконфигурированным для выполнения задачи.
- любая компьютерная система, готовая к встраиванию в уже функционирующую систему.

ЛИТЕРАТУРА

Основная литература

1. Кудрявцев, Н. Г. Элементарные основы программирования встраиваемых систем [Электронный ресурс] : учеб. пособие / Н. Г. Кудрявцев. – Горно-Алтайск : ГАГУ, 2021. – 148 с. // «Лань» : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/178005> (дата обращения: 24.06.2022).

2. Петин, В. В. Практическая энциклопедия Arduino [Электронный ресурс] : энциклопедия / В. В. Петин, А. А. Биняковский. – 2-е изд., испр. и доп. – М. : ДМК Пресс, 2020. – 166 с. // «Лань» : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/131675> (дата обращения: 24.06.2022).

Рекомендуемая литература

1. Рацеев, С. М. Программирование на языке Си [Электронный ресурс] : учеб. пособие для вузов / С. М. Рацеев. – СПб. : Лань, 2022. – 332 с. // «Лань» : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/193320> (дата обращения: 27.06.2022).

2. Рысин, М. Л. Основы программирования на языке C++ [Электронный ресурс] : учеб. пособие / М. Л. Рысин, М. В. Сартаков, О. В. Макеева. – М. : РТУ МИРЭА, 2022. – 118 с. // «Лань» : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/239957> (дата обращения: 27.06.2022).

ПРИЛОЖЕНИЕ А

(справочное)

Начало работы в среде Tinkercad

Для того чтобы начать проект, перейдите на сайт Tinkercad (<https://www.tinkercad.com/>) (рис. А.1).

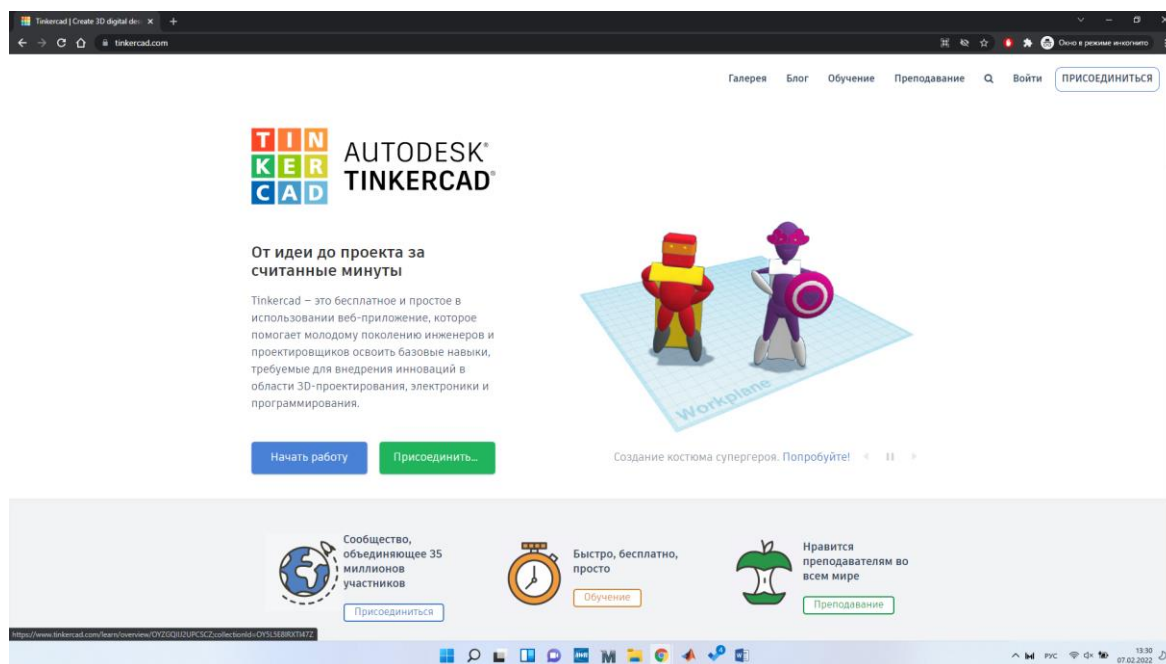


Рис. А.1 – Окно сайта Tinkercad

Далее нажмите на кнопку «Начать работу». В открывшемся окне (рис. А.2) нажмите на кнопку «Создать персональную учетную запись» и выберите удобный для вас способ регистрации.

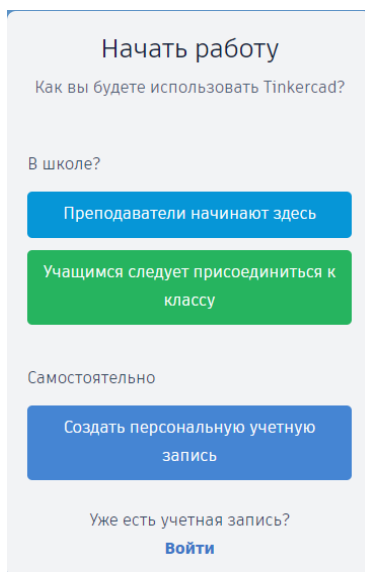


Рис. А.2 – Создание проекта

После успешной регистрации и входа в аккаунт (рис. А.3) необходимо перейти в раздел «Цепи» и создать новый проект, нажав «Создать Цепь».

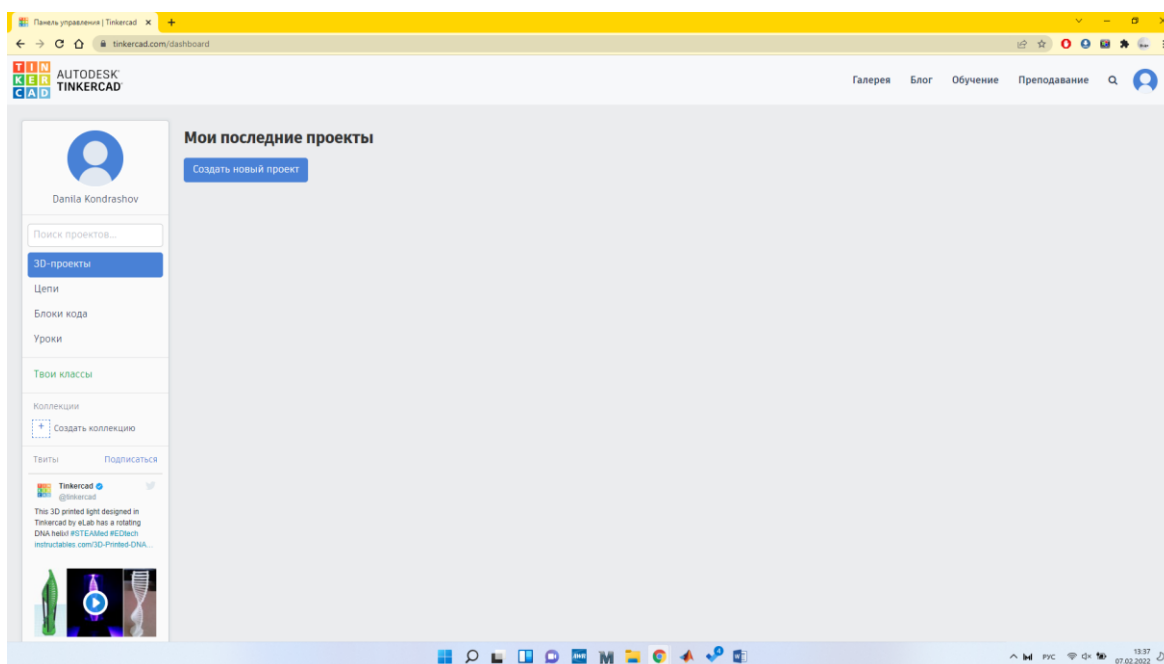


Рис. А.3 – Новый проект

После того как проект будет создан, вам будет доступно рабочее поле (рис. А.4).

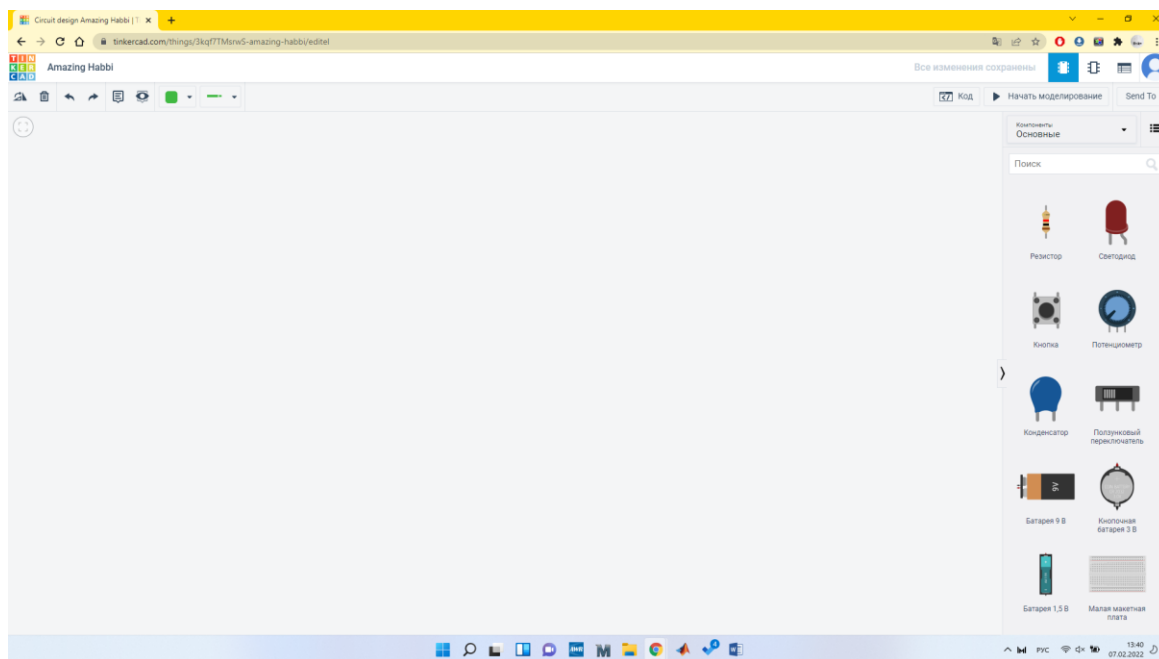


Рис. А.4 – Рабочее поле проекта

В библиотеке компонент, находящейся справа, в поисковой строке необходимо написать «Arduino Uno R3» и достать плату Arduino из библиотеки в рабочее поле путем переноса. Рабочее поле будет представлено на рисунке А.5.

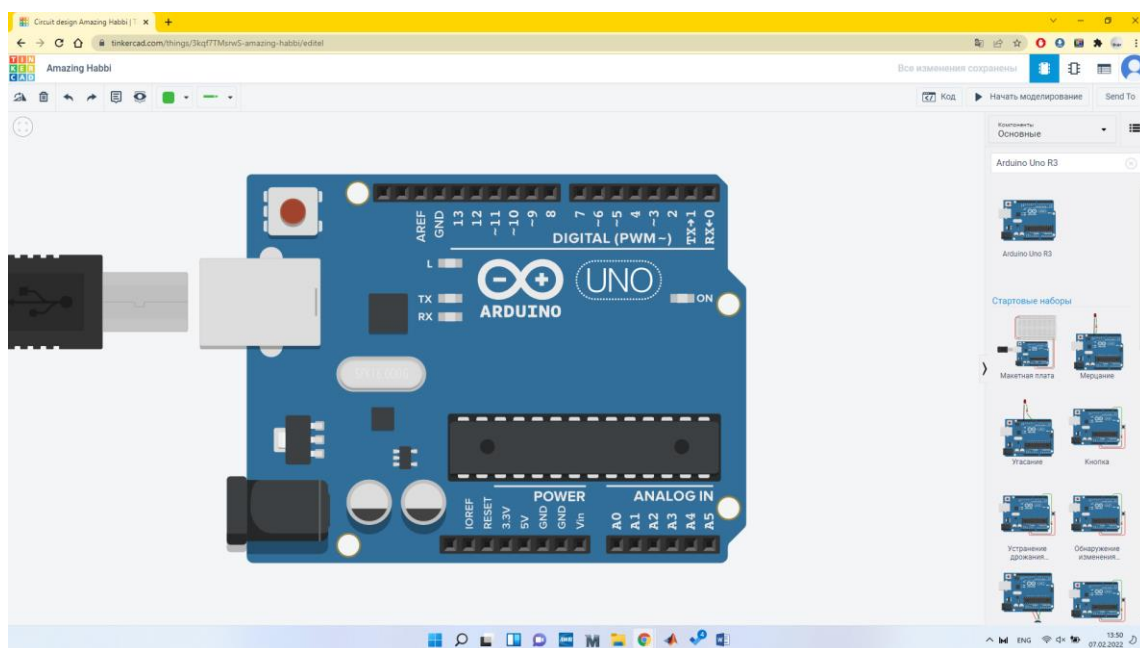



Рис. А.5 – Подготовленное рабочее поле

Далее следует нажать на кнопку  Код, находящуюся над библиотекой компонент, и очистить поле программирования (рис. А.6).

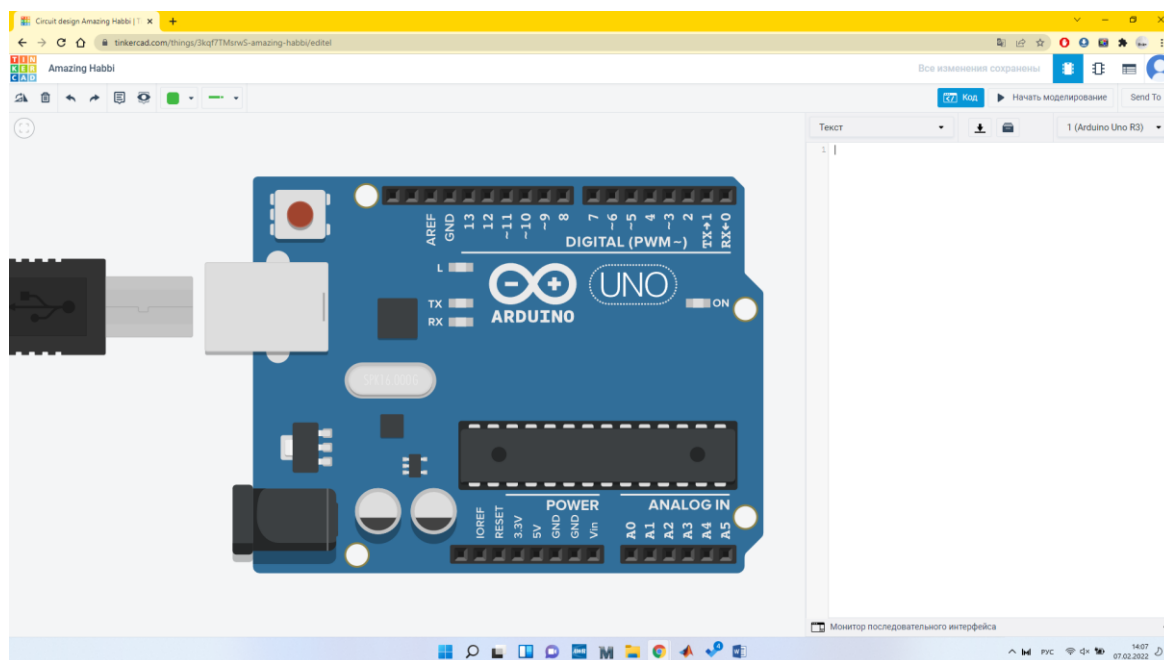


Рис. А.6 – Готовое рабочее пространство

После того как проект был создан, можно приступить к его разработке. Потребуется следующие функции:

1. `void setup()` – это функция, которая выполняется **один** раз при запуске контроллера. При запуске микроконтроллера код, который находится в этой функции, выполнится **единожды** и будет выполняться каждый раз при включении микроконтроллера.

Обычно в данной функции происходит настройка микроконтроллера, к примеру:

- настройка пинов;
- инициализация устройств.

2. `void loop()` – это функция, которая будет непрерывно вызываться каждый раз после функции `setup()`, то есть код, который содержится в данной функции, будет выполняться **циклически без перерыва**.

Комментарии могут выглядеть следующим образом:

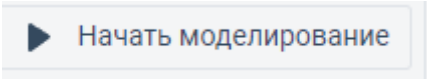

//Строчный комментарий

*/*Комментарий*

Который содержит

Несколько строк/*

Используйте комментарии в своем коде для того, чтобы лучше понимать работу программы и в данный момент, и через несколько недель или месяцев!

После того как все настройки будут выполнены, необходимо скомпилировать программу и запустить моделирование. Для это необходимо нажать кнопку . Начнется проверка скетча на ошибки и его компиляция. В случае отсутствия ошибок кнопка загорится зеленым , на плате Arduino индикатор «On» загорится зеленым цветом (рис. А.7).

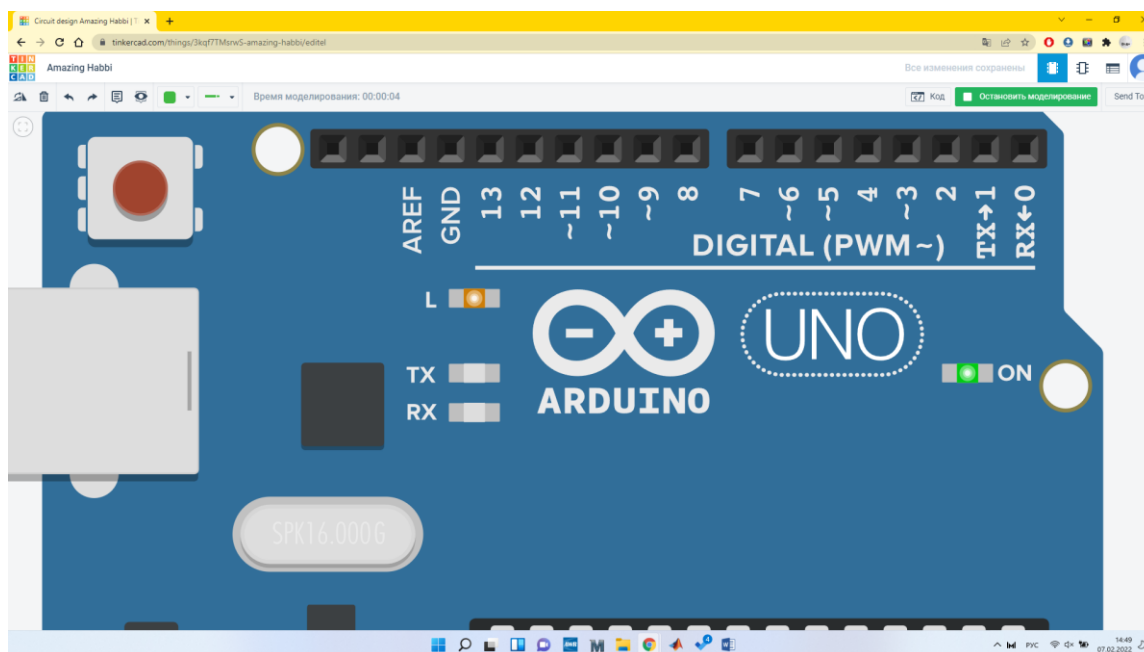
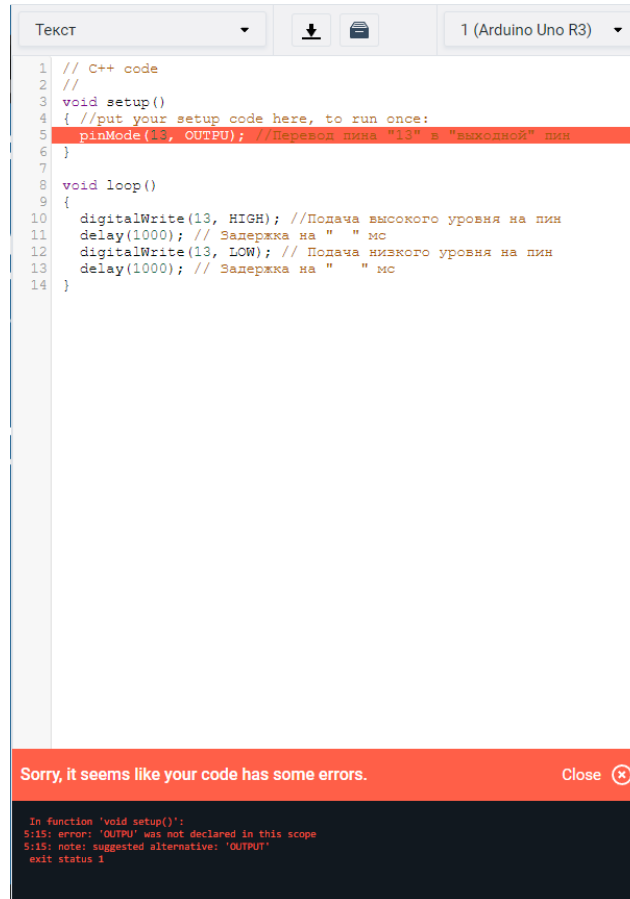


Рис. А.7 – Плата в рабочем состоянии

В случае если в коде были допущены ошибки, компилятор при попытке запуска выдаст ошибку о невозможности запуска кода со ссылками на строки, где была допущена ошибка (рис. А.8).



```
Текст 1 (Arduino Uno R3)
1 // C++ code
2 //
3 void setup()
4 { //put your setup code here, to run once:
5   pinMode(13, OUTPU); //Перевод пина "13" в "выходной" пин
6 }
7
8 void loop()
9 {
10  digitalWrite(13, HIGH); //Подача высокого уровня на пин
11  delay(1000); // Задержка на " " мс
12  digitalWrite(13, LOW); // Подача низкого уровня на пин
13  delay(1000); // Задержка на " " мс
14 }
```

Sorry, it seems like your code has some errors. Close

In function 'void setup()':
5:15: error: 'OUTPU' was not declared in this scope
5:15: note: suggested alternative: 'OUTPUT'
exit status 1

Рис. А.8 – Допущена ошибка в коде