



Одним из вариантов программирования в наборе «ЗНАТОК для Arduino START» (в дальнейшем «**Arduino START**») является самый простой вариант языка, разработанный компанией «ЗНАТОК», для перехода от визуального программирования к программированию проектов уже на «серьезном» языке **C++\Arduino**. Составив программу из простых блоков-действий можно увидеть, как эта же программа выглядит на **C++**.

Выполнив ряд проектов, вы сможете понимать, как изменение в алгоритме влияет на текст программы, и сможете уже самостоятельно писать код программ.

СОВМЕСТИМОСТЬ «ARDUINO START» И «GRAPH Z»

В отличие от использования «классического» подхода с **Arduino IDE**, где текстовая программа пишется на компьютере и потом загружается на плату и выполняется автономно, использование графического языка **Graph Z** подразумевает составление программ при помощи простых блоков-действий, которые потом трансформируются в код C++ и уже потом последующую загрузку программ.

Graph Z не является полноценным средством для программирования Arduino, но является первым шагом в обучении и обеспечивает плавный переход к программированию на языке C++.

Не все проекты, которые можно выполнить при помощи C++ можно выполнить в среде Graph Z, зато это может показаться вам проще и веселее!

УСТАНОВКА

1. Скачайте последнюю версию обучающей среды Graph Z с сайта <https://znatok.ru/link/?graphZ-installer>
2. Следуя инструкциям установщика, установите Graph Z на ваш компьютер
3. Установите драйвер для модуля Arduino (см. бумажную инструкцию стр. 22)
4. Запустите среду Graph Z

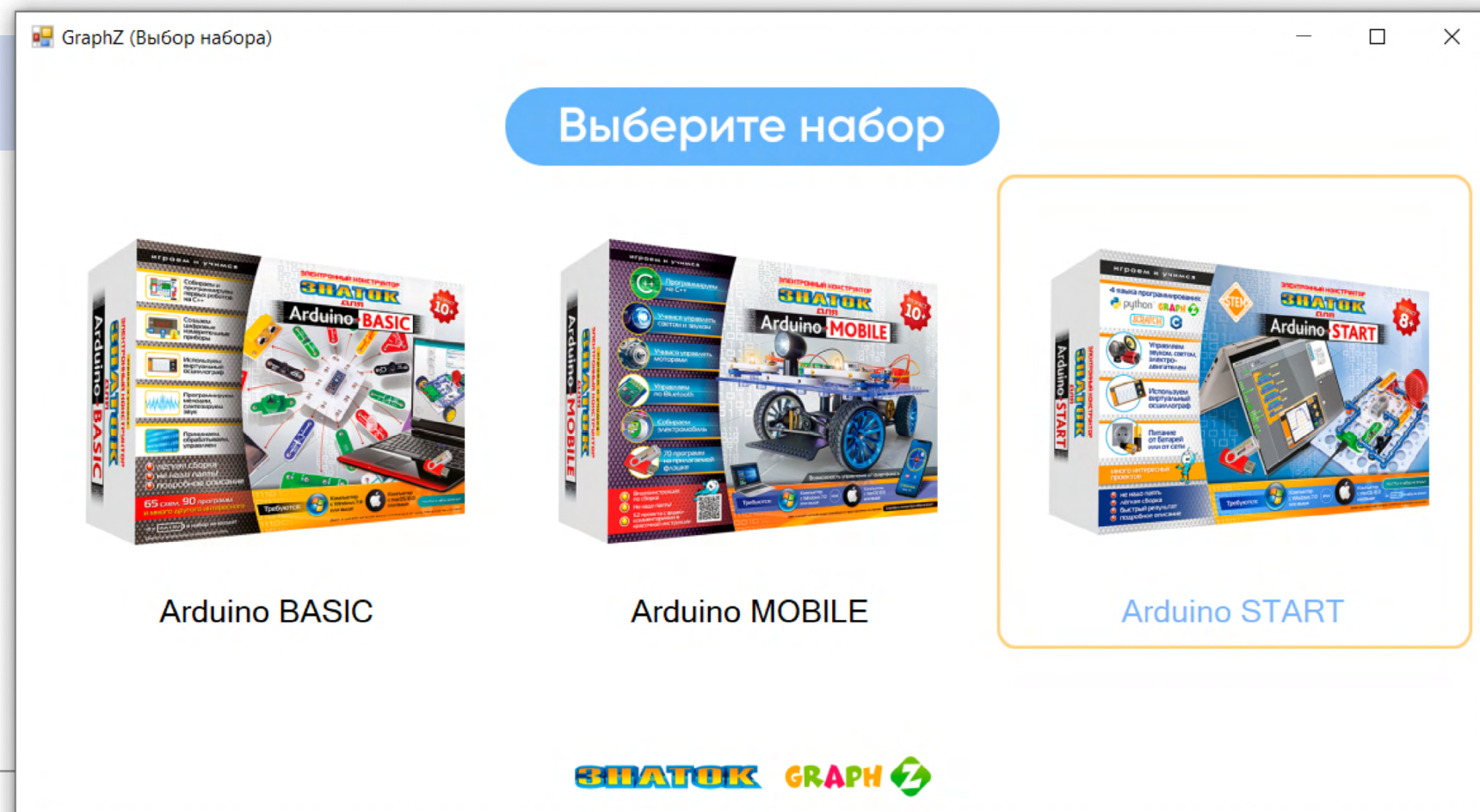
ОПИСАНИЕ ЯЗЫКА ПРОГРАММИРОВАНИЯ

Данное руководство позволит управлять проектами с использованием модуля **111** при помощи составленных алгоритмов-программ на языке Graph Z.



ЗАПУСК

При запуске программы Graph Z будет предложен выбор набора, который у вас есть.



ОПИСАНИЕ ЯЗЫКА ПРОГРАММИРОВАНИЯ



ЗАПУСК (ПРОДОЛЖЕНИЕ)

После, вы сможете выбрать схему, с которой вы хотите поработать.

Обратите внимание – доступны 2 режима работы – «Начальный», который рекомендуется для делающих самые первые шаги в изучении программирования, и «Продвинутый» – для тех кто научился работать с «Начальным» и хочет создавать что то своё.

Вам откроется рабочее поле, на котором можно выполнять проекты или создавать собственные блок-схемы и изучать программирование.

Выбор схемы

Выберите схему

- Control_LED-1-5 стр. 27-29
- Control_LED-7-8 стр. 33-34
- Control_LED-13 стр.38
- RGB_Mix-1 стр. 45
- RGB_Mix-2-3 стр. 47-48**
- Motor_Control-1 стр. 53
- Motor_Control-2 стр. 55
- Lamp_Control-1 стр. 56

Выберите режим

ПРОДВИНУТЫЙ

НАЧАЛЬНЫЙ

ЗНАТОК GRAPH



ИСПОЛЬЗОВАНИЕ

ИНТЕРФЕЙС РАБОЧЕГО ПОЛЯ

1. Строка меню – необходима для настройки проекта, а так же предоставляет возможность Сохранять\Загружать проект
2. Текущая схема – схема, с которой выполняется в данный момент работа. Возможно кликнуть на неё – откроется увеличенное изображение.
3. Список доступных блоков – все элементы для создания алгоритма, которые можно использовать в данной собранной схеме
4. Рабочее поле – поле, на котором расположены все блоки создаваемой программы
5. Код на языке С++ – код, который получается из блоков, которые размещены в рабочем поле. Динамично изменяется в зависимости от того как соединять блок с другими.





ИСПОЛЬЗОВАНИЕ (ПРОДОЛЖЕНИЕ)

ИСПОЛЬЗОВАНИЕ РАБОЧЕГО ПОЛЯ

МАСШТАБИРОВАНИЕ

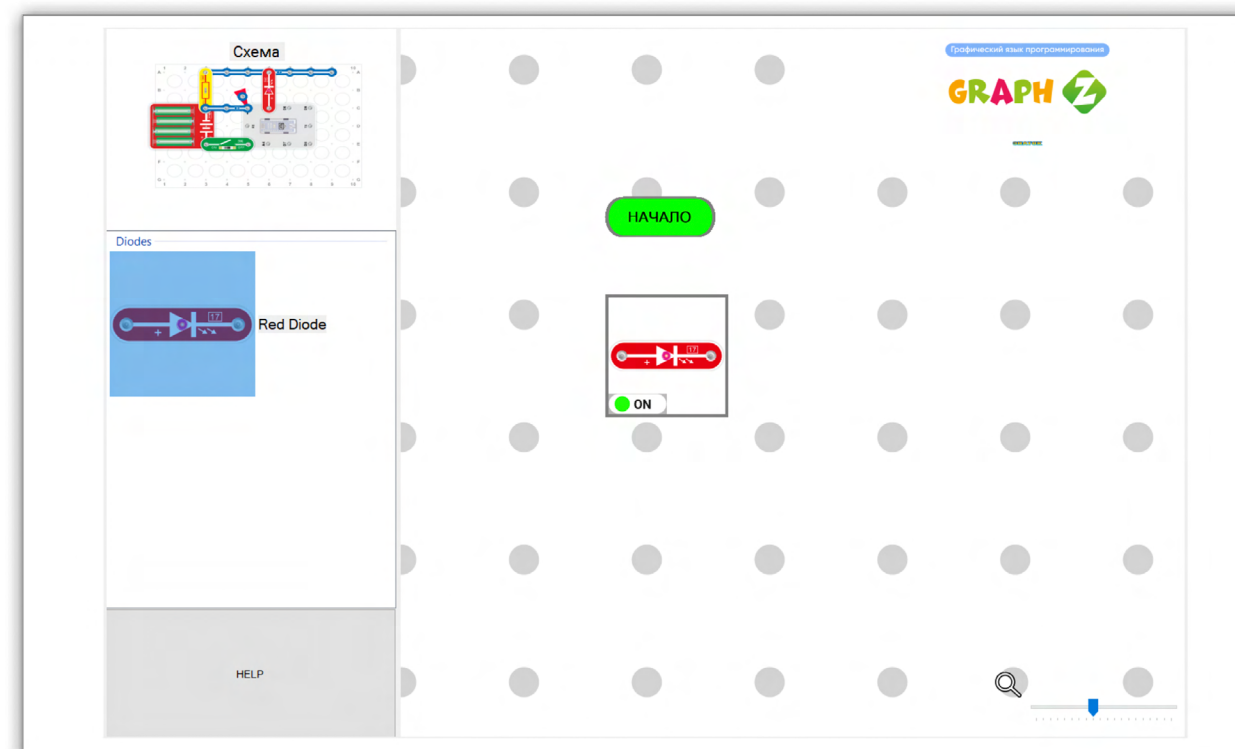
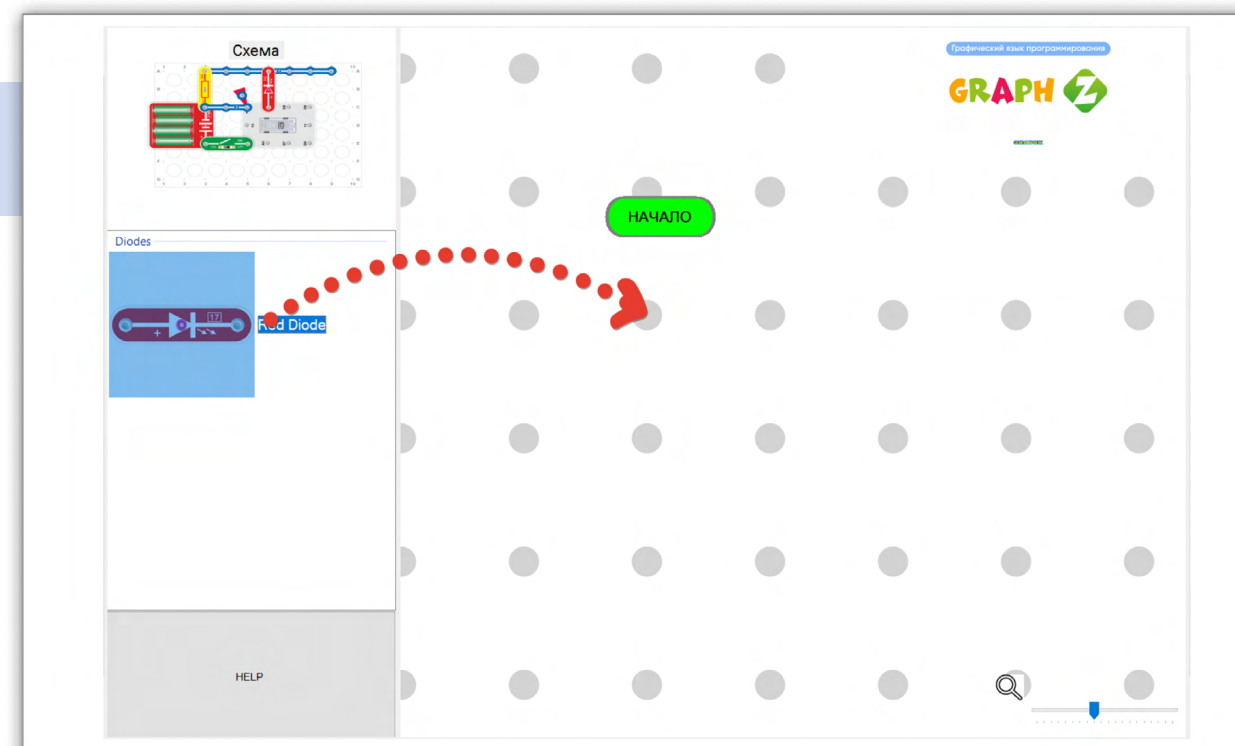
Для масштабирования рабочего поля может быть использован ползунок в нижнем правом углу рабочего поля. Так же это действие может выполняться при помощи колесика мыши (прокрутка вниз – «Отдалить», прокрутка вверх – «Приблизить»).

ПЕРЕМЕЩЕНИЕ ПО РАБОЧЕМУ ПОЛЮ

Для перемещения следует разместить курсор мыши на рабочем поле, зажать колесико мыши и передвигать мышь в сторону желаемого перемещения

РАЗМЕЩЕНИЕ БЛОКОВ

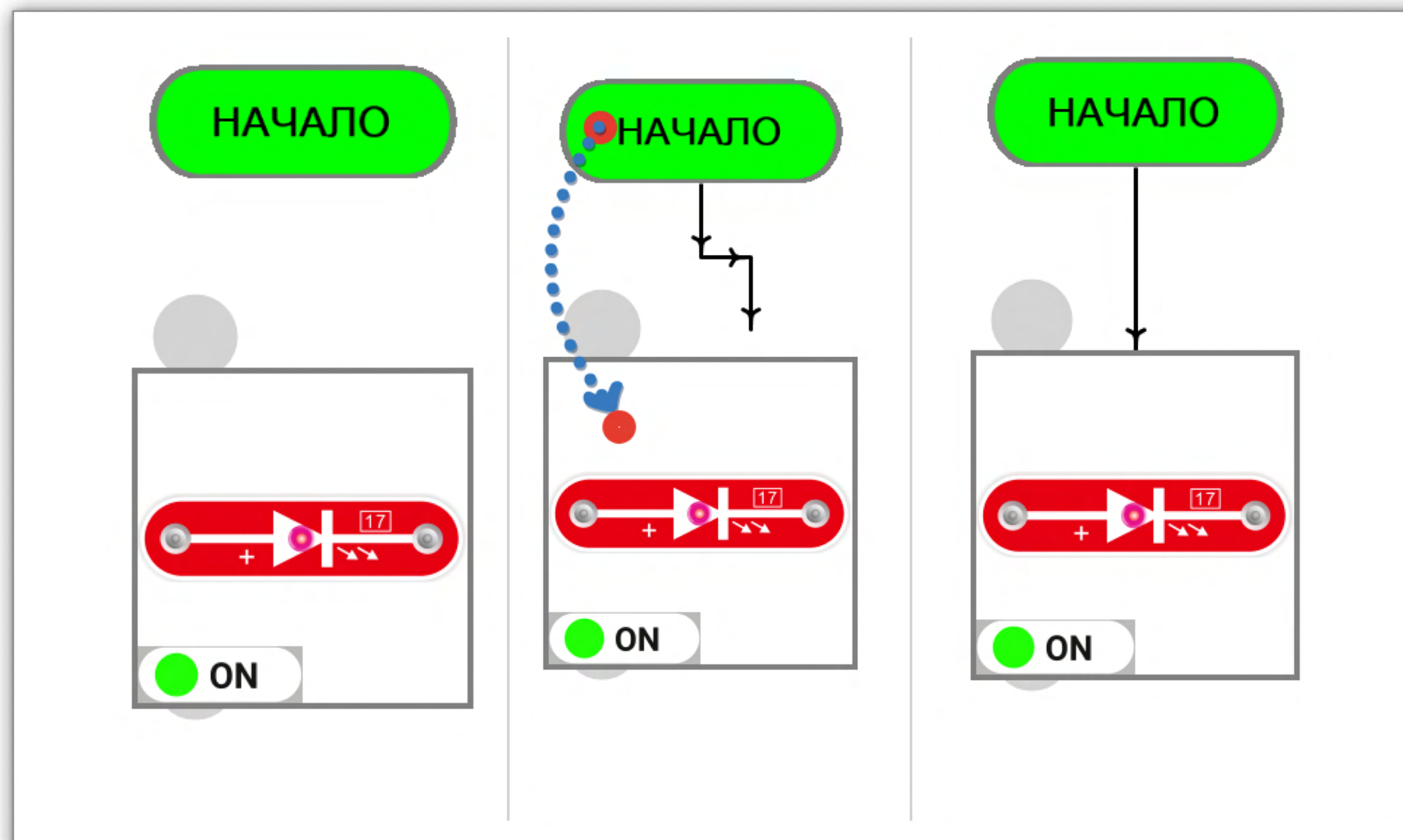
Для размещения блока, необходимо нажать на левую кнопку мыши на нужном блоке из Списка доступных блоков и «перетащить» блок на Рабочее поле.





СОЕДИНЕНИЕ БЛОКОВ - СОЗДАНИЕ СВЯЗЕЙ

Для соединения блоков между собой, необходимо кликнуть правой кнопкой мыши по блоку от которого должна идти связь, и потом кликнуть на блок к которому эта связь должна "дойти". В процессе, будет изображаться стрелочка, которая эту связь будет обозначать. Для того, чтобы связь зафиксировать, необходимо кликнуть по второму блоку правой кнопкой мыши.

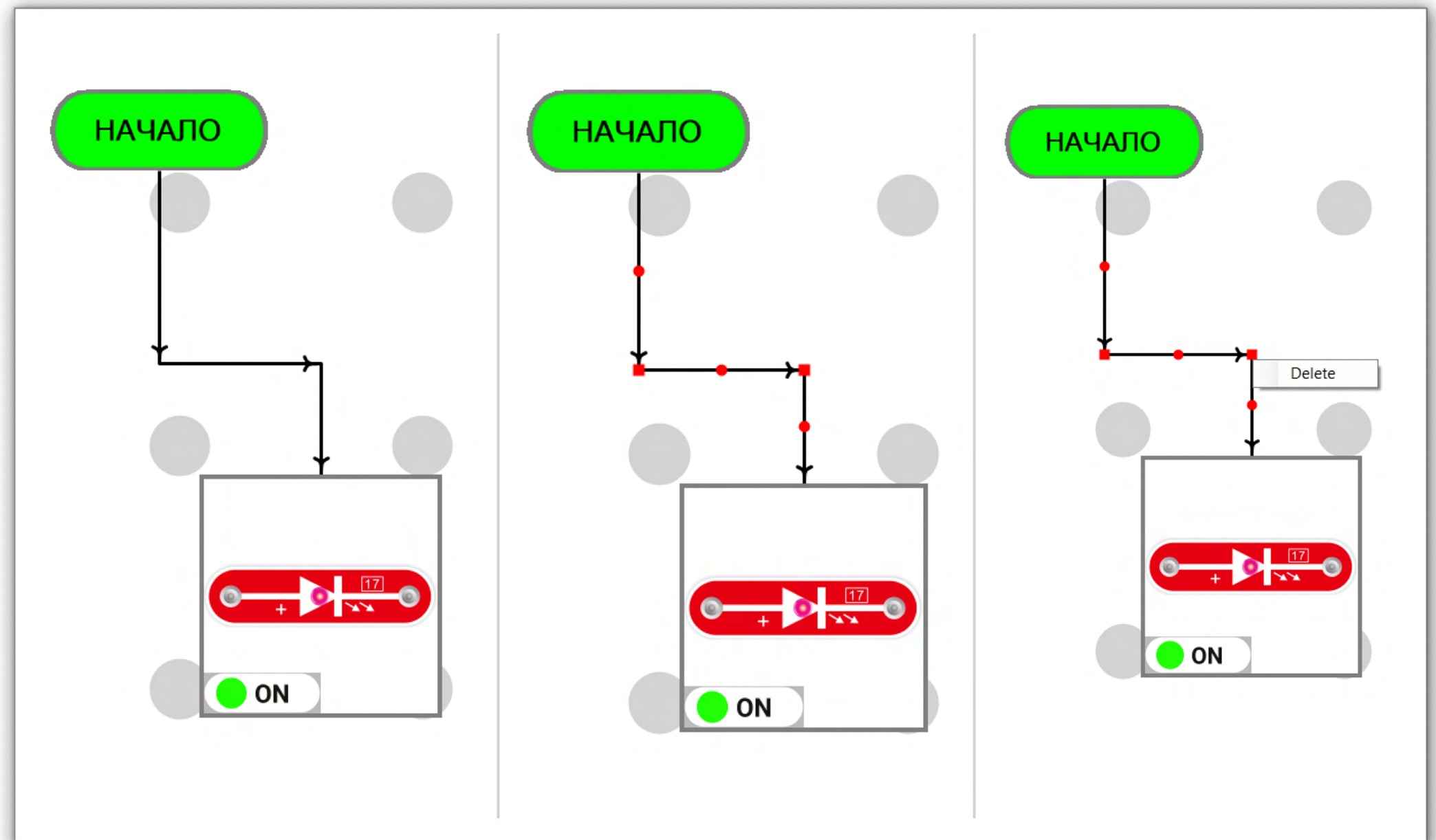




УДАЛЕНИЕ СВЯЗЕЙ-СТРЕЛОЧЕК

Для удаления связи-стрелочки, необходимо:

1. Выделить стрелочку – кликнуть на неё
2. Нажать на стрелочку правой кнопкой мыши – вызвать контекстное меню
3. Нажать на пункт «Delete»\«Удалить»





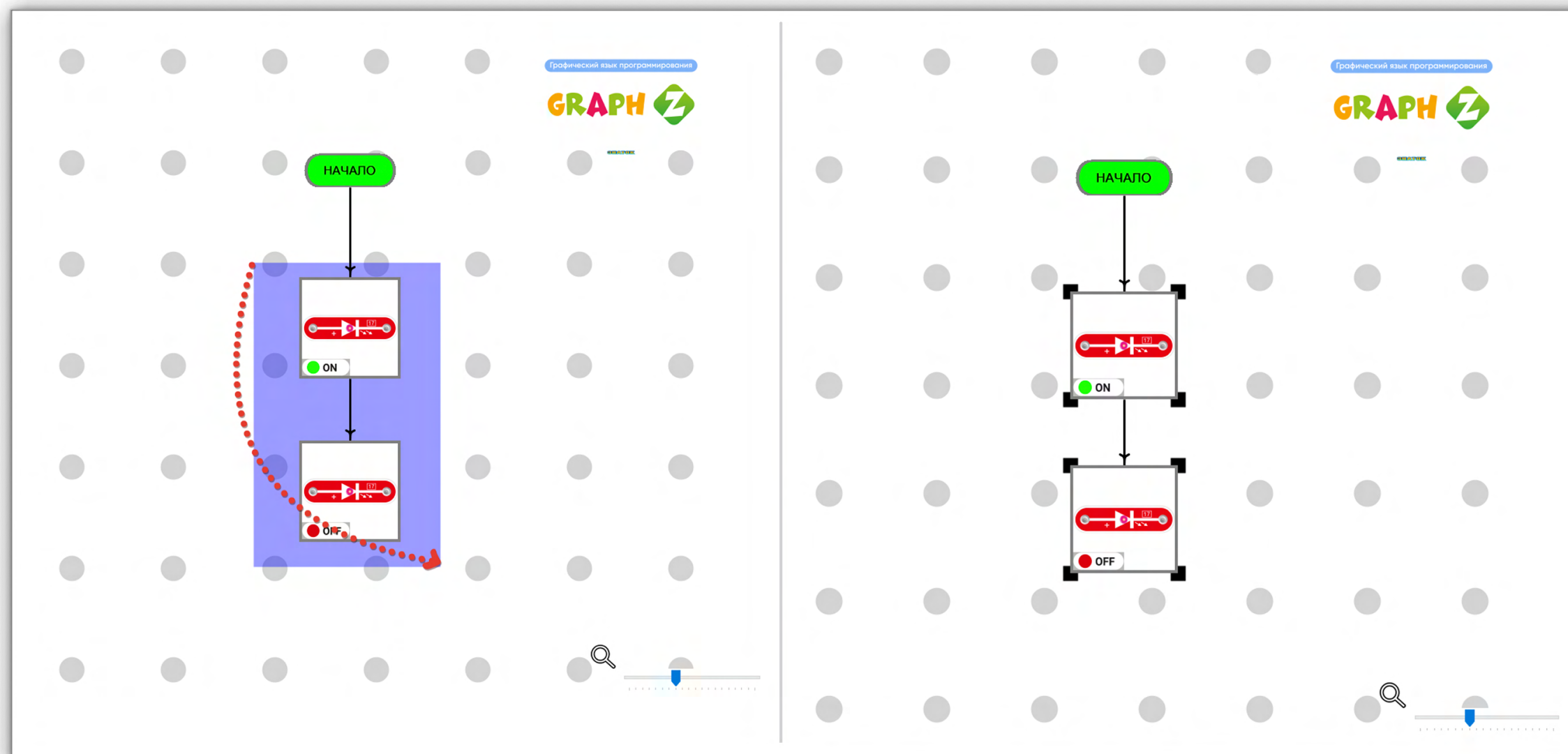
ИСПОЛЬЗОВАНИЕ (ПРОДОЛЖЕНИЕ)

ВЫДЕЛЕНИЕ БЛОКОВ

Для того, чтобы выделить блок или несколько блоков, необходимо на Рабочем поле зажать левую клавишу мыши на свободной области и выделить прямоугольник со всеми нужными элементами.

УДАЛЕНИЕ БЛОКОВ

Для удаления блоков, необходимо выделить блоки, которые нужно удалить, кликнуть на свободной области правой мыши, вызвав меню и нажать на пункт «Delete»\«Удалить».





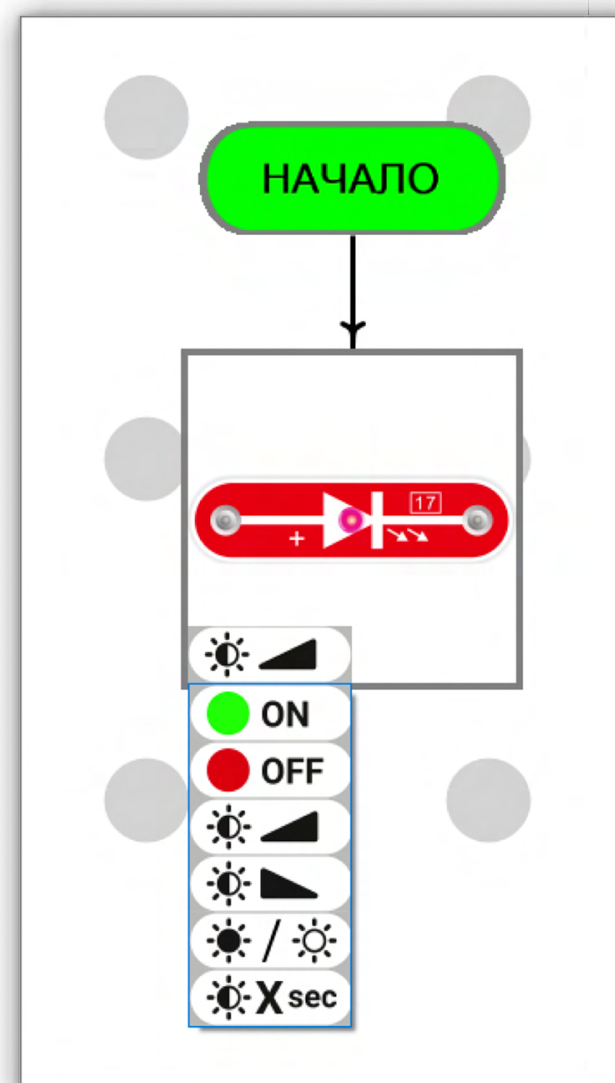
СОЗДАНИЕ БЛОК-СХЕМЫ

Каждая блок схема, состоит из блоков-действий, последовательно соединенных между собой.

ВАЖНО! Самый первый блок – блок «Старт» – создается автоматически. Это точка начала работы алгоритма.

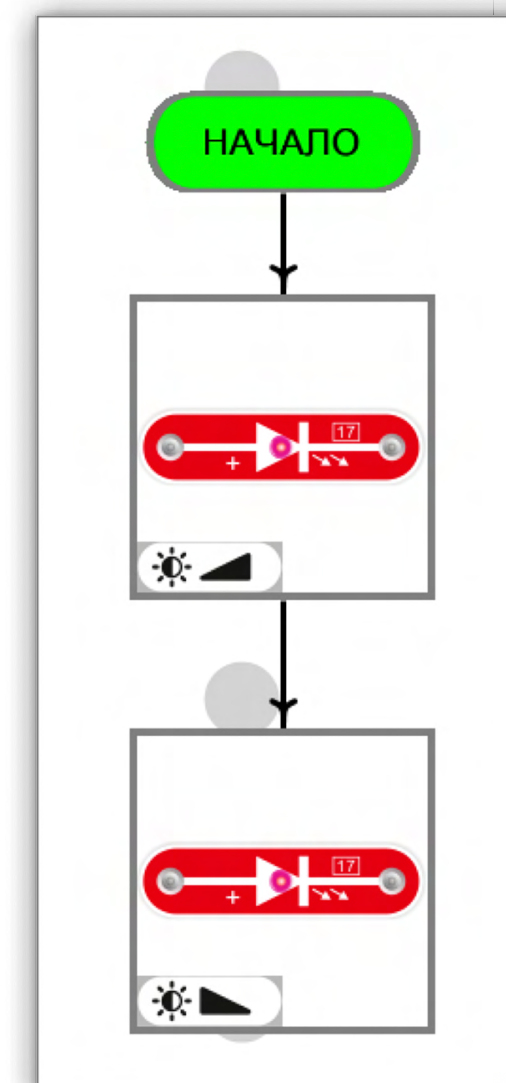
Все действия в программе будут выполняться последовательно – шаг за шагом, «идя по стрелочкам»

У каждого блока есть параметры работы. Например, блок Светодиода в первой схеме в «Базовом» режиме позволяет выбрать одно из нескольких действий.



Из выпадающего меню можно выбрать одно из действий, доступных для данного элемента: Включить, Выключить, Плавно увеличить яркость, плавно уменьшить яркость, Мигнуть, Включить на несколько секунд.

Создадим блок-схему, которая позволит красному светодиоду плавно увеличить яркость, а потом плавно уменьшить яркость:





ИСПОЛЬЗОВАНИЕ (ПРОДОЛЖЕНИЕ)

Программа, написанная по такому алгоритму, выполнится только один раз при включении. Сделаем так, чтобы она повторялась постоянно – сделаем цикл. Проведем связь от нижнего блока к верхнему (можно обратить внимание, что в правой части экрана – в зоне кода C++ – код изменился):

```
#define LED_RED 5

int DiodeSuperRedVar = 0;

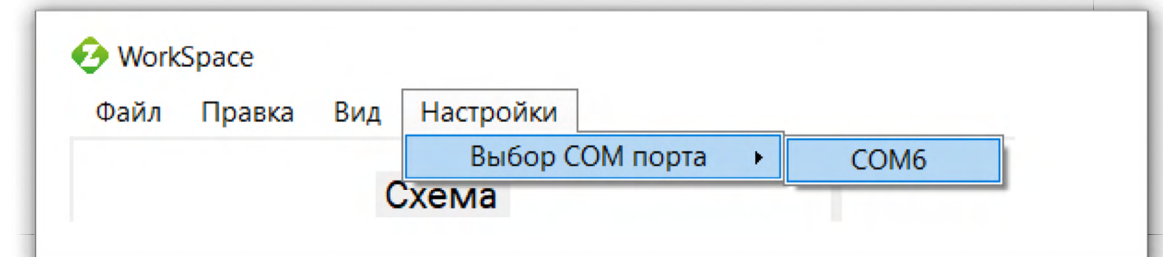
void setup()
{
  pinMode(LED_RED, OUTPUT);
}

void loop()
{
  for (int br = 0; br < 255; br += 5)
  {
    analogWrite(LED_RED, br);
    delay(5);
  }
  for (int br = 255; br > 0; br -= 5)
  {
    analogWrite(LED_RED, br);
    delay(5);
  }
}
```

ПОДКЛЮЧЕНИЕ МОДУЛЯ 111 И ЗАГРУЗКА ПРОГРАММЫ

Если вы правильно выполнили инструкцию по установке драйвера модуля Arduino, подключили модуль при помощи USB-кабеля к компьютеру и создали блок-схему программы, которую вы хотите загрузить, следует сделать следующие шаги:

1. Выберите COM-порт, к которому подключен ваш модуль Arduino в верхнем меню в разделе «Настройки» (в примере приведен порт COM6, но это может порт с другим именем).



ОПИСАНИЕ ЯЗЫКА ПРОГРАММИРОВАНИЯ



ИСПОЛЬЗОВАНИЕ (ПРОДОЛЖЕНИЕ)

2. Нажмите на кнопку «Проверить и загрузить» в нижнем правом углу программы:

The screenshot shows the graphical programming environment. On the left, there is a 'Схема' (Circuit) window showing a breadboard layout with a red diode and a resistor. Below it is a 'Diodes' component palette with a 'Red Diode' component. The main workspace contains a flowchart starting with a green 'НАЧАЛО' (Start) block, followed by two 'Red Diode' components connected in a loop. On the right, a code editor window displays the following C++ code:

```
Код
#define LED_RED 5

int DiodeSuperRedVar = 0;

void setup()
{
  pinMode(LED_RED, OUTPUT);
}

void loop()
{
  for (int br = 0; br < 255; br += 5)
  {
    analogWrite(LED_RED, br);
    delay(5);
  }
  for (int br = 255; br > 0; br -= 5)
  {
    analogWrite(LED_RED, br);
    delay(5);
  }
}
```

At the bottom right, a button labeled 'ПРОВЕРИТЬ И ЗАГРУЗИТЬ' (CHECK AND UPLOAD) is highlighted with a red box. A red arrow points from the 'НАЧАЛО' block to this button.

3. Дождитесь окончания загрузки (МОГУТ ПОЯВЛЯТЬСЯ И ПРОПАДАТЬ СИСТЕМНЫЕ ОКНА ЗАГРУЗКИ). Если все сделано верно, появится надпись «Программа успешно загружена».

The screenshot shows the same graphical programming environment as in the previous step. A dialog box with the title 'Программа успешно загружена' (Program successfully loaded) and an 'ОК' button is overlaid on the circuit diagram. The flowchart and code editor are still visible in the background.

ДОСТУПНЫЕ ПРОЕКТЫ ИЗ НАБОРА ARDUINO START

Для языка Graph Z в данном наборе доступны следующие программы – аналоги из бумажной инструкции:

| | | |
|------------------|--------------------|------------------|
| 01_Control_LED-1 | 03_RGB_Mix-3 | 23_Smart_Light-5 |
| 01_Control_LED-2 | 05_Motor_Control-1 | 23_Smart_Light-6 |
| 01_Control_LED-3 | 05_Motor_Control-2 | Synth-2a |
| 01_Control_LED-4 | 06_Lamp_Control-1 | Synth-2b |
| 01_Control_LED-5 | 06_Lamp_Control-4 | Signaling-1 |
| 01_Control_LED-7 | 06_Lamp_Control-5 | Signaling-2a |
| 01_Control_LED-8 | 06_Lamp_Control-6 | Signaling-2b |
| 03_RGB_Mix-1 | 06_Lamp_Control-7 | |
| 03_RGB_Mix-2 | 23_Smart_Light-1 | |