

Научно-исследовательская работа

Предмет Информатика

## **РАЗРАБОТКА И ПРОГРАММИРОВАНИЕ ТАНЦУЮЩЕГО РОБОТА**

**Выполнил:**

*Красовский Станислав Александрович  
учащийся 3 Б класса Муниципального  
автономного общеобразовательного  
учреждения средней школы №5  
г. Павлово Нижегородской области*

**Учитель:**

*Горшкова Елена Евгеньевна  
МАОУ СШ №5 г.Павлово  
Нижегородской области*

**Павлово**

**2020**

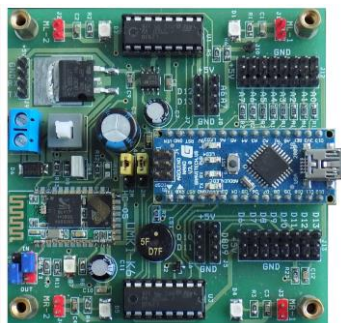
## Оглавление

Введение .....	1
Основная часть. Что такое робот. Какие бывают роботы.....	2
Как работает сервомотор .....	3
Программируем четыре сервомотора .....	5
Программирование мелодии .....	8
Программирование танцующего робота .....	11
Корпус робота с помощью 3D ручки.....	12
Как программировать один светодиод. ....	13
Как программировать цепочку светодиодов. Бегущие огни .....	15
Заключение.....	19
Литература.....	20
Приложение 1. Среда программирования Snap! .....	21
Приложение 2. Как работает 3D ручка.....	24

## Введение

Я люблю заниматься робототехникой. Я немного знаю язык программирования Scratch, на котором можно создавать компьютерные игры. Во втором классе я научился программировать робота-машинку, которая ездит по программе. Но настоящие роботы передвигаются не только на колесах, и мне захотелось сделать робота, который умеет ходить, как человек – ногами.

В книжке про роботов и я прочитал, что для того чтобы автоматически управлять разными частями робота нужно специальное устройство – контроллер. Контроллер находится на специальной плате, его можно запрограммировать, и программа останется в памяти.



Плата с контроллером который называется Arduino у меня уже была. Именно к ней я подключал моторчики, которые вращали колеса машинки, но как же заставить робота ходить ногами? Я узнал что для такого управления нужны другие моторчики, которые называются «Сервомоторы».

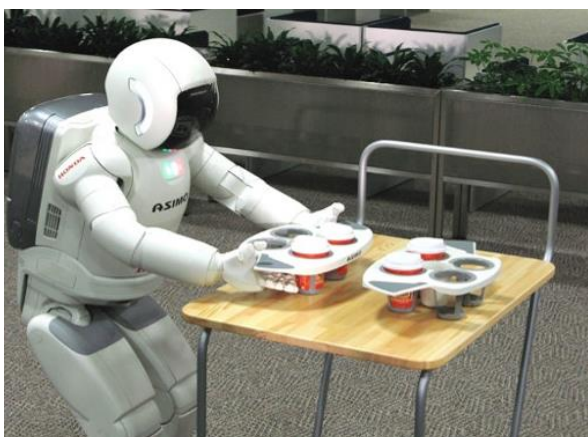
Но в книжке программы для роботов были очень сложные, на том языке программирования который я не знал. И бабушка мне сказала, что для контроллера Arduino программы можно попробовать составить и на том языке на котором я уже умею программировать, но только там обязательно должны быть команды которые понимает робот. Такой ящик с командами для робота был в программе Snap4Arduino (Приложение 1).

Так как книжек про программирование роботов в программе Snap! нет, поэтому мне пришлось разбираться в создании программ для робота самому, а бабушка мне помогала.

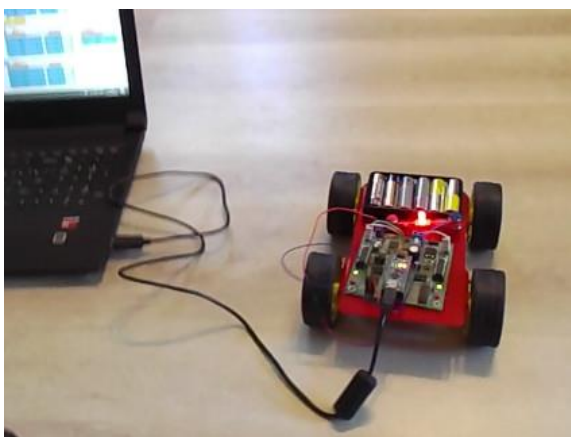
## Основная часть.

### Что такое робот. Какие бывают роботы

Роботы применяются в самых разных областях нашей жизни: они помогают лечить людей и делать автомобили, роботы летают в космос и опускаются под воду, работают на полях и строят дома. Еще существуют роботы, которые помогают нам в быту и развлекают нас. Такие роботы часто похожи на людей и животных и могут по-стоящему шагать



Но, если к роботу подключить обычный мотор и нацепить на него колеса, то робот будет ездить как обычная машинка.



Но, если мы хотим робота который ходит, то нам не нужно чтобы элементы на которых робот перемещается постоянно крутились, как колеса – ведь наши ноги не вращаются. Оказывается для управления конечностями

нужны особые моторы. Подходящий нам моторчик называется сервопривод или сервомотор.

## Как работает сервомотор

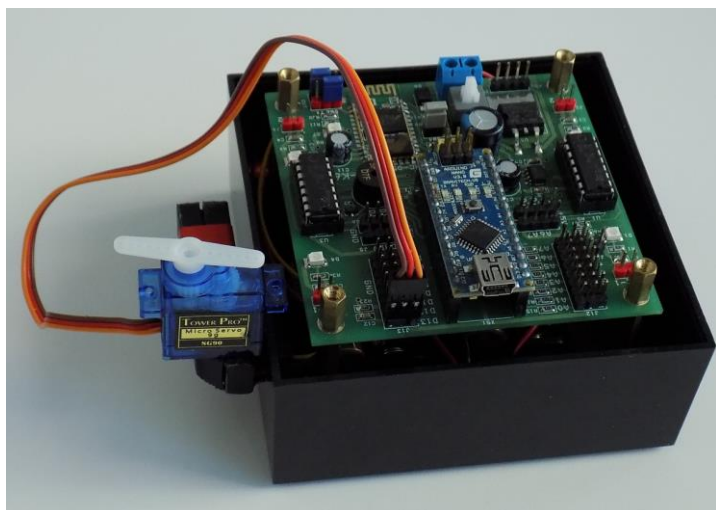
Сервомотор, в отличие от обычного мотора не вращается постоянно, а только поворачивается на угол, который мы можем изменить из программы. То есть мы можем управлять положением вала мотора. Сервоприводы используются для моделирования различных механических движений роботов.

Выводы моторчика имеют три цвета:

- коричневый или черный – подключение к земле (gnd)
- красный – питание (+5 вольт)
- оранжевый желтый или белый – подача управляющего сигнала,

Все выводы соединены вместе и нужно только внимательно смотреть на разъем при подключении.

Сначала, я научился управлять таким сервомотором.



Я подключил сервомотор к шестому контакту контроллера и стал подавать сигналы на этот контакт и менять угол поворота:

при нажатии клавиши пробел ▾

изменить статус разъема серво 6 ▾ на 0 ▾

при нажатии клавиши 1 ▼

изменить статус разъема серво 6 ▼ на 45 ▼

при нажатии клавиши 2 ▼

изменить статус разъема серво 6 ▼ на 90 ▼

При нажатии клавиши 1 – сервомотор поворачивался совсем чуть-чуть, при нажатии клавиши 2 – сервомотор поворачивался на прямой угол, при нажатии на клавишу «Пробел» сервомотор возвращался в исходное состояние.

Также я провел эксперименты, подключая этот сервомотор к другим контактам: шестому, седьмому, восьмому, девятому.

## Программируем четыре сервомотора

Я взял четыре сервомотора и подключил их к тем контактам с которыми уже экспериментировал. При нажатии на клавишу «Пробел» все сервомоторы встают в исходное положение.

при нажатии клавиши пробел ▾

изменить статус разъема серво 6 ▾ на 0 ▾

изменить статус разъема серво 7 ▾ на 0 ▾

изменить статус разъема серво 8 ▾ на 0 ▾

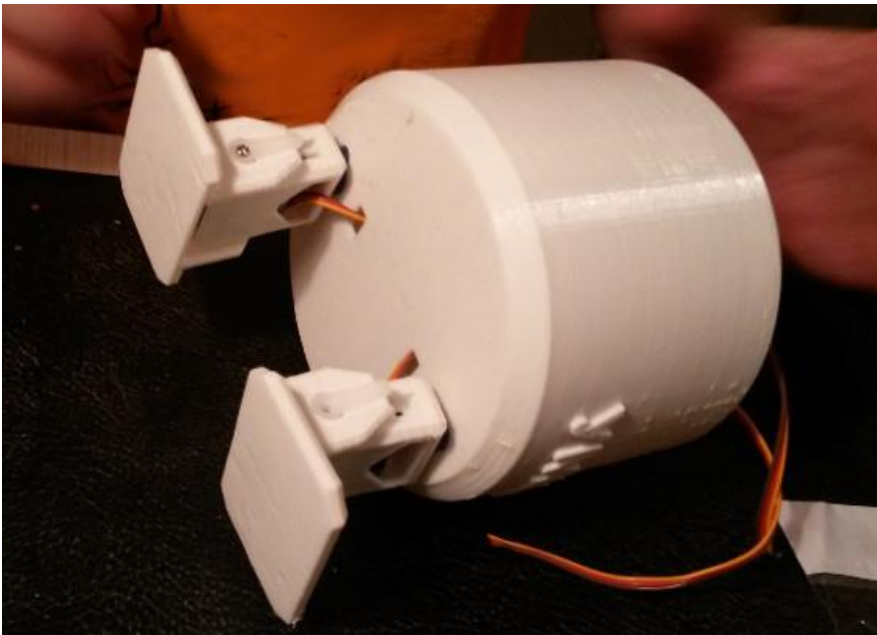
изменить статус разъема серво 9 ▾ на 0 ▾

Мне подарили вот такой наборчик для крепления сервомоторов.



И я собрал эти сервомоторы в виде ног робота.





И составил программу, чтобы робот двигал ступнями

при нажатии клавиши 1 ▾

изменить статус разъема серво 7 ▾ на 45 ▾

изменить статус разъема серво 9 ▾ на 45 ▾

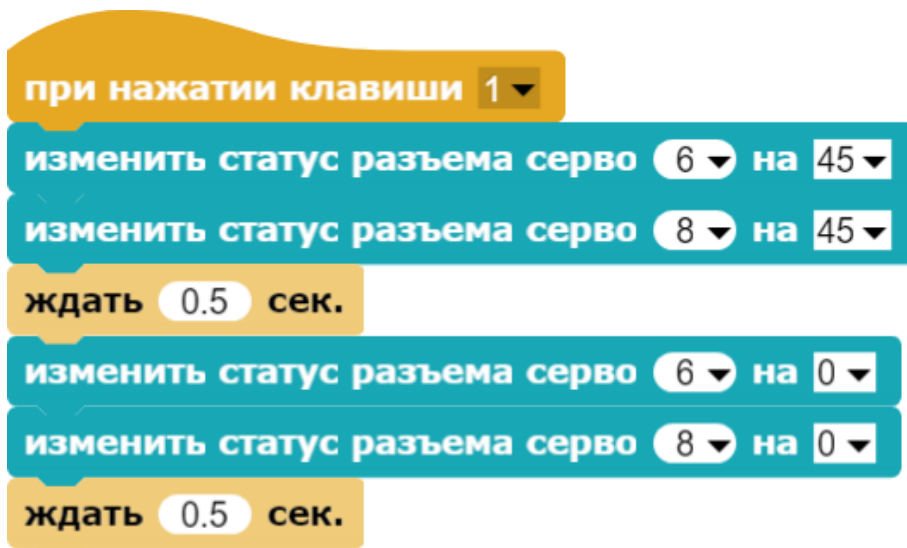
ждать 0.5 сек.

изменить статус разъема серво 7 ▾ на 0 ▾

изменить статус разъема серво 9 ▾ на 0 ▾

ждать 0.5 сек.

или поворачивался



Повторяя в разных комбинациях такие кусочки программы можно заставить робота изобразить танец. Но это будет очень скучный танец, если не будет музыки.

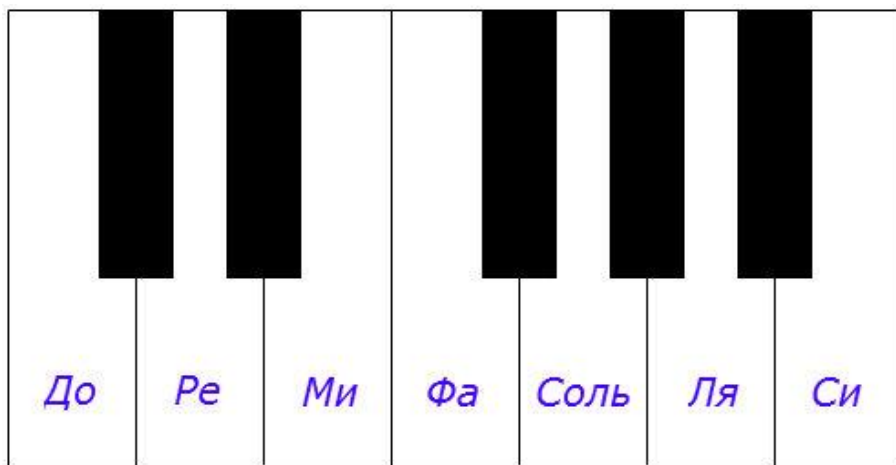
Поэтому мне пришлось познакомиться с нотами.

## Программирование мелодии

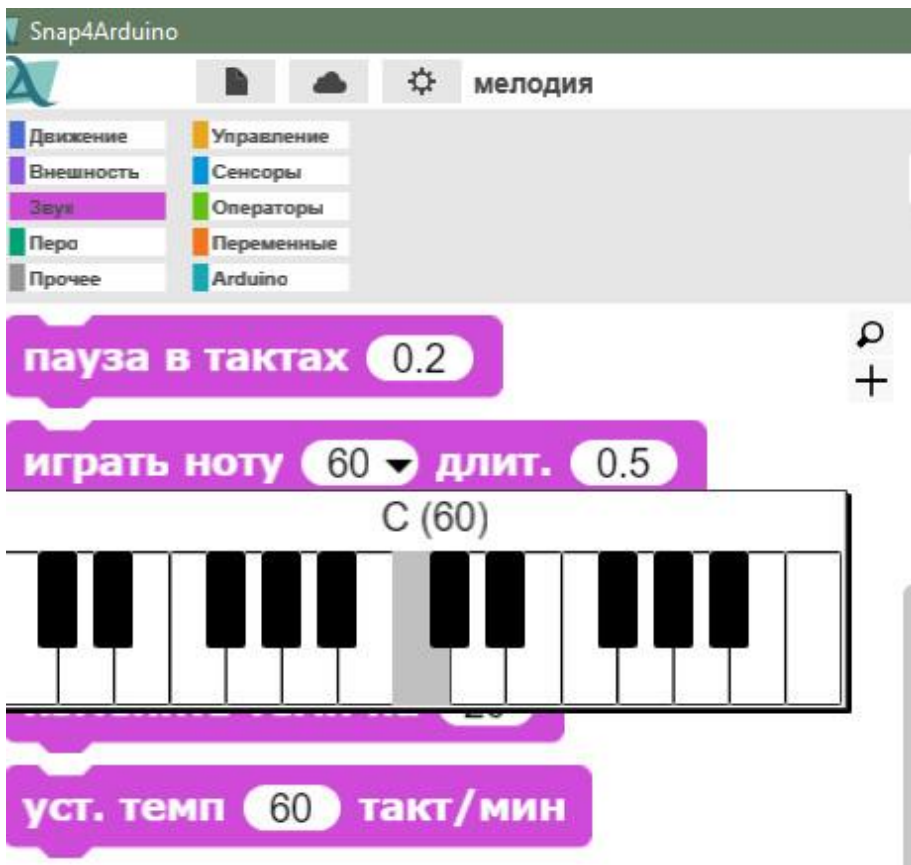
Я узнал, что ноты могут быть разной длительности: целые, половинные, четвертные восьмые и шестнадцатые. Записываются ноты на специальном нотном стане, из пяти линеек, и от того на какой линейке записана нота, зависит ее звучание и название.



Я разобрался какая нота играется на какой клавише клавиатуры и научился подбирать на компьютере несложные песенки.



Для того чтобы запрограммировать ноты мне потребуются команды из ящика «Звуки»



Оказывается, когда выбираем параметр команды играть ноту, открывается такая же клавиатура. Я запрограммировал несколько песенок, и больше всего мне понравилась про маленькую елочку.

4/4

Ма-лень-кой ё-лоч-ке хо-лод-но зи-мой

Из ле-су ё-лоч-ку взя-ли мы до-мой

Маленькой елочке  
Холодно зимой.  
Из лесу елочку  
Взяли мы домой.  
Из лесу елочку  
Взяли мы домой.

Сколько на елочке  
Шариков цветных,  
Розовых пряников,  
Шишек золотых!  
Розовых пряников,  
Шишек золотых!

Бусы повесили,  
Встали в хоровод.  
Весело - весело  
Встретим Новый год.  
Весело - весело  
Встретим Новый год.

Вот такая программа у меня получилась:


The image shows a Scratch script for a musical program. It starts with a 'when green flag clicked' event block. The script then sets the instrument to 2, volume to 60%, and tempo to 95 bpm. A 0.1-second pause follows. The main sequence is enclosed in a 'repeat 3' loop. Inside this loop, there is a 'repeat 2' loop containing three 'play note' blocks: 55 (duration 1), 52 (duration 0.5), and 52 (duration 0.5). After this inner loop, there are five 'play note' blocks: 55 (0.5), 53 (0.5), 52 (0.5), 50 (0.5), and 48 (2). Finally, there is another 'repeat 2' loop at the end of the script. This second 'repeat 2' loop contains a 'repeat 2' loop with six 'play note' blocks: 57 (1), 60 (0.5), 57 (0.5), 55 (1), 52 (0.5), and 52 (0.5). This is followed by four 'play note' blocks: 55 (0.5), 53 (0.5), 52 (0.5), and 50 (0.5), and ends with a 'play note' block of 48 (2).

```
при нажатии на   
инструмент 2  
set volume to 60 %  
пауза в тактах 0.1  
уст. темп 95 такт/мин  
повторять 3  
  повторять 2  
    играть ноту 55 длит. 1  
    играть ноту 52 длит. 0.5  
    играть ноту 52 длит. 0.5  
  играть ноту 55 длит. 0.5  
  играть ноту 53 длит. 0.5  
  играть ноту 52 длит. 0.5  
  играть ноту 50 длит. 0.5  
  играть ноту 48 длит. 2  
повторять 2  
  повторять 2  
    играть ноту 57 длит. 1  
    играть ноту 60 длит. 0.5  
    играть ноту 57 длит. 0.5  
    играть ноту 55 длит. 1  
    играть ноту 52 длит. 0.5  
    играть ноту 52 длит. 0.5  
  играть ноту 55 длит. 0.5  
  играть ноту 53 длит. 0.5  
  играть ноту 52 длит. 0.5  
  играть ноту 50 длит. 0.5  
  играть ноту 48 длит. 2
```

## Программирование танцующего робота

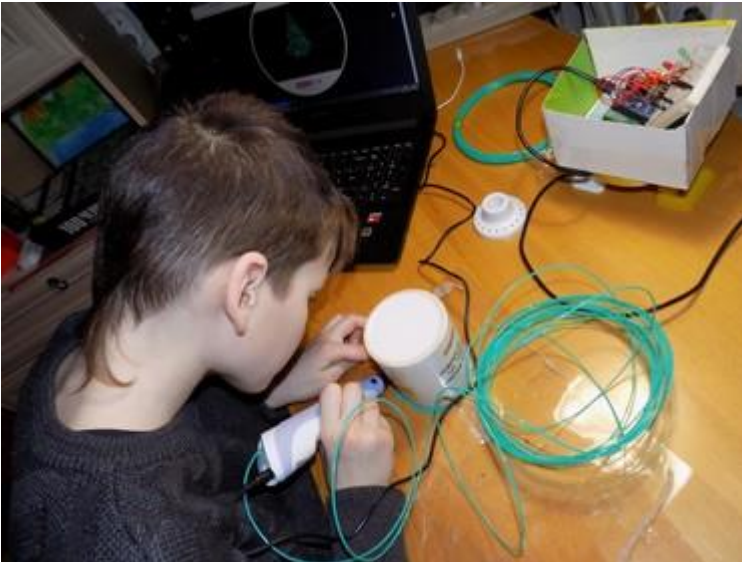
Под эту музыку я стал подбирать повороты сервомоторов для движения ног робота.

The image shows two columns of Scratch code blocks. The left column starts with a 'when green flag clicked' block, followed by four 'change servo port status' blocks (ports 6, 8, 9, 7 to 0). This is followed by a 'repeat 4' loop containing: 'change servo port status' (7 to 45), 'change servo port status' (9 to 0), 'wait 0.5 sec.', 'change servo port status' (7 to 0), 'change servo port status' (9 to 45), 'wait 0.5 sec.', and another 'change servo port status' (7 to 0), 'change servo port status' (9 to 0), 'wait 0.5 sec.' block. The right column starts with a 'repeat 4' loop containing: 'change servo port status' (6 to 45), 'change servo port status' (8 to 45), 'wait 0.5 sec.', 'change servo port status' (6 to 0), 'change servo port status' (8 to 0), 'wait 0.5 sec.', 'wait 0.5 sec.', and another 'repeat 4' loop containing: 'change servo port status' (7 to 45), 'change servo port status' (9 to 45), 'wait 0.5 sec.', 'change servo port status' (7 to 0), 'change servo port status' (9 to 0), 'wait 0.5 sec.', and another 'change servo port status' (7 to 0), 'change servo port status' (9 to 0), 'wait 0.5 sec.' block.

```
при нажатии на   
изменить статус разъема серво 6 на 0  
изменить статус разъема серво 8 на 0  
изменить статус разъема серво 9 на 0  
изменить статус разъема серво 7 на 0  
повторять 4  
изменить статус разъема серво 7 на 45  
изменить статус разъема серво 9 на 0  
ждать 0.5 сек.  
изменить статус разъема серво 7 на 0  
изменить статус разъема серво 9 на 45  
ждать 0.5 сек.  
изменить статус разъема серво 7 на 0  
изменить статус разъема серво 9 на 0  
ждать 0.5 сек.  
повторять 4  
изменить статус разъема серво 6 на 45  
изменить статус разъема серво 8 на 45  
ждать 0.5 сек.  
изменить статус разъема серво 6 на 0  
изменить статус разъема серво 8 на 0  
ждать 0.5 сек.  
ждать 0.5 сек.  
повторять 4  
изменить статус разъема серво 7 на 45  
изменить статус разъема серво 9 на 45  
ждать 0.5 сек.  
изменить статус разъема серво 7 на 0  
изменить статус разъема серво 9 на 0  
ждать 0.5 сек.
```

## Корпус робота с помощью 3D ручки

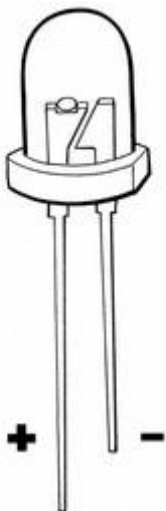
Но раз робот танцевал под новогоднюю песенку, то с помощью 3d ручки я сделал конструкцию которая изображает новогоднюю елочку. (Приложение 2)



## Как запрограммировать один светодиод.

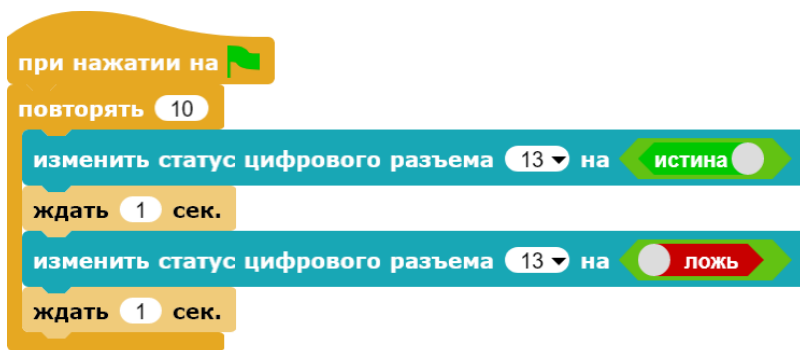
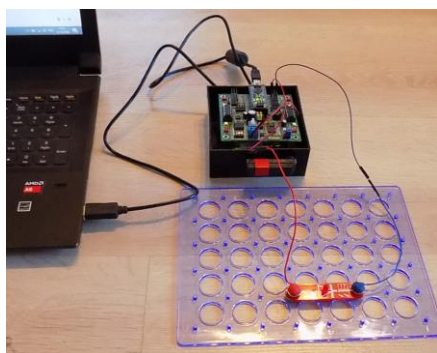
Потом я разобрался с программированием светодиодов. Светодиод это такое устройство которое излучает свет, когда через него протекает электрический ток.

У светодиода два контакта и он будет испускать свет, когда ток протекает от плюса к минусу.



Чтобы светодиод не сгорел, в цепь обязательно подключают еще один электронный компонент – резистор.

Сначала к контроллеру я подключил один светодиод. У контроллера много ножек и светодиод плюсом я подключил к контакту Arduino D13, минусом светодиод я подключил к контакту «Земля». Я составил такую программу:



По этой программе сначала на 13 контакт контроллера идет сигнал, одну секунду, в это время светодиод горит, через секунду сигнал становится

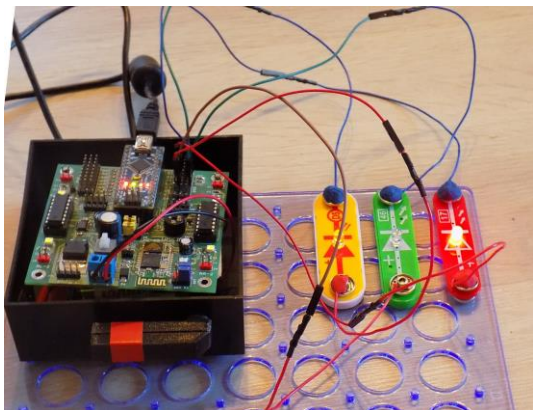


низким и светодиод гаснет на одну секунду. И мигание будет повторяться десять раз. Я менял время в команде «Ждать»: например, если задать время 3 секунды то светодиод будет мигать гораздо медленнее.

# Как запрограммировать цепочку светодиодов.

## Бегущие огни

Потом я подключил и составил программу для двух и для трех светодиодов. Светодиоды я подключал к 13, 10 и 9 контактам Arduino.



Первая программа для двух светодиодов

```
при нажатии на [флаг]
  повторить 10
    изменить статус цифрового разъема 13 на ИСТИНА
    изменить статус цифрового разъема 10 на ИСТИНА
  ждать 1 сек.
  изменить статус цифрового разъема 13 на ЛОЖЬ
  изменить статус цифрового разъема 10 на ЛОЖЬ
  ждать 1 сек.
```

Два светодиода на 10 и 13 контактах мигают одновременно.

Затем я изменил программу:

при нажатии на   
непрерывно

изменить статус цифрового разъема 13 на 

изменить статус цифрового разъема 10 на 

ждать 1 сек.

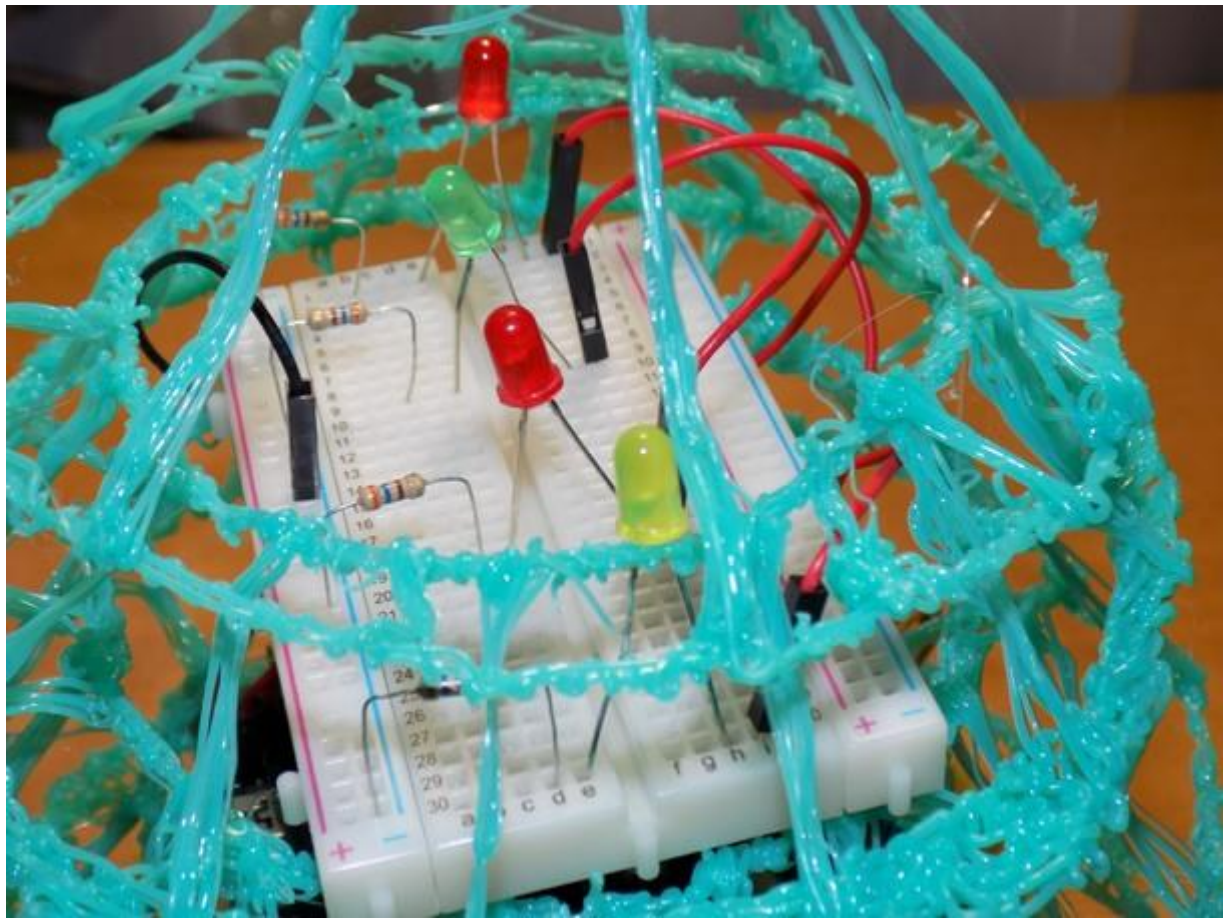
изменить статус цифрового разъема 13 на 

изменить статус цифрового разъема 10 на 

ждать 1 сек.

Два светодиода мигают на по-очереди – когда один светит второй темный и наоборот. Теперь программа будет повторяться до тех пор, пока мы не дадим команду остановить программу.

Потом я добавил еще два светодиода

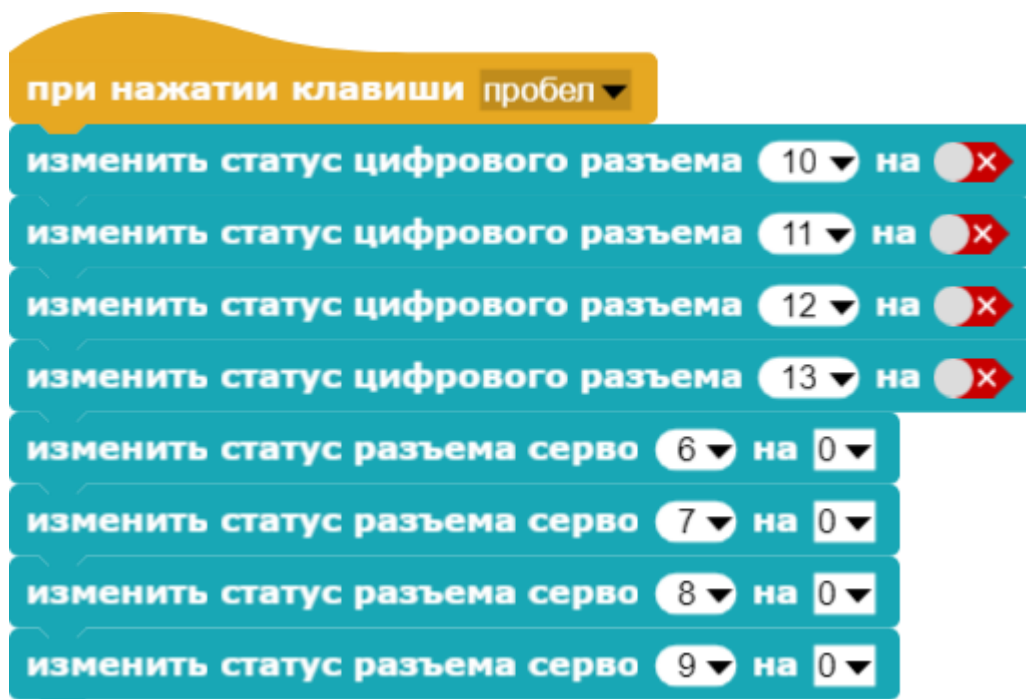


и изменил программу так:



Когда я запускаю программу, то загорается светодиод на 10 ножке контроллера, через полсекунды он гаснет, а загорится светодиод на 11 ножке, и светит половину секунды, потом он погаснет, а будет гореть светодиод на 12

ножке, а через 0.5 секунды на 13-й, и так будет продолжаться всегда. Чтобы погасить все светодиоды я нажимаю клавишу пробел – это показывает вторая часть программы, где на все ножки идет низкий сигнал и гирлянда выключается.



Вывод: я увидел, что мои светодиоды работали совсем как гирлянда с бегущими огнями, и я разместил светодиоды на новогодней елочке.

## Заключение

- Так я собрал и запрограммировал в среде Snap робота, который умеет танцевать.
- Я запрограммировал для него музыку и сделал корпус в виде новогодней елочки и украсил его разноцветными огоньками, которые мигают по программе.
- Теперь я хочу сделать робота который будет выполнять другие более сложные задачи: он будет не только танцевать, но и ходить ногами, и еще сделаю плавающего робота.

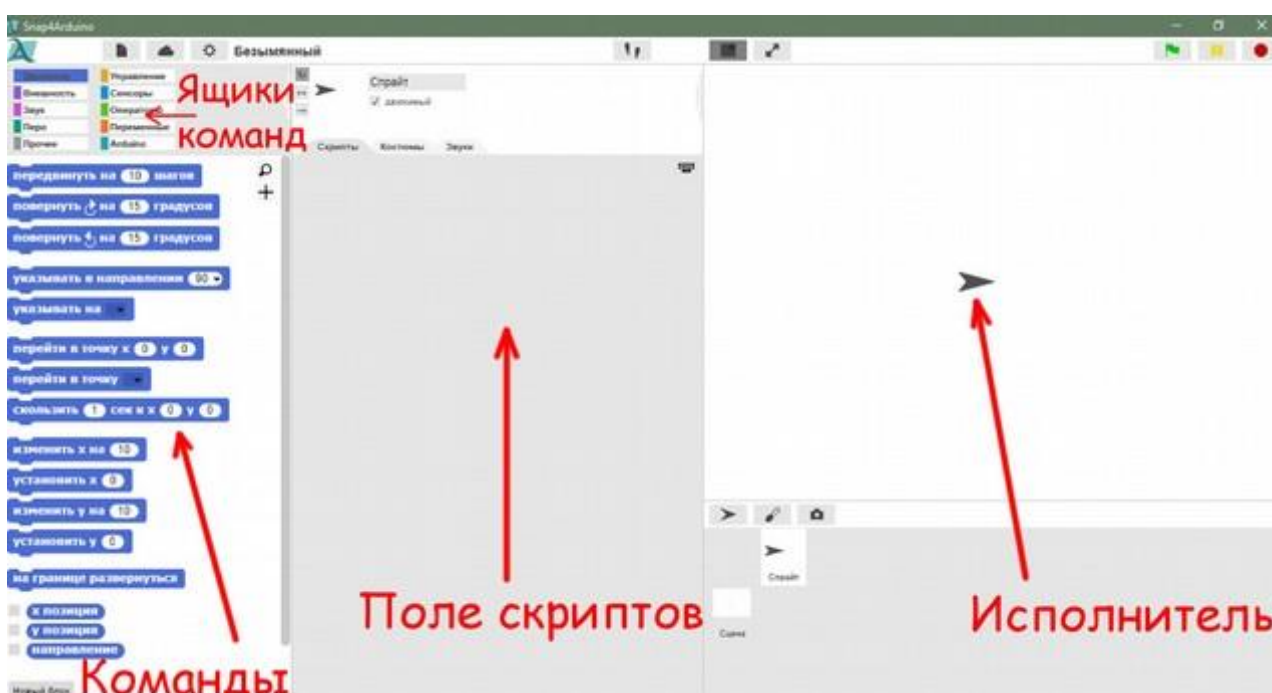
## Литература

1. И.Воронин, В.Воронина «От основ к созданию роботов». Питер. 2018
2. А.Николаева, В.Натансон, Л.Рощина «Фортепианная игра» Москва Музыка 1991
3. Знакомьтесь — Роботы! <http://umki-dist.ru/course/view.php?id=16>
4. Что такое 3d pen и зачем она нужна? <https://vk.com/@abspla3d-что-такое-3d-pen-i-zachem-ona-nuzhna>
5. 3D Лампа – Новогодняя Ёлка [http://3d-lampy.ru/index.php?route=product/product&product\\_id=82](http://3d-lampy.ru/index.php?route=product/product&product_id=82)

## Приложение 1. Среда программирования Snap!

Snap! – объектно-ориентированная среда программирования, в которой можно познакомиться с основами алгоритмизации и программирования, научиться создавать анимации и компьютерные игры, сделать первые шаги в программировании механизмов и робототехнических устройств.

Программное обеспечение относится к свободно распространяемым ресурсам, под лицензией GNU.

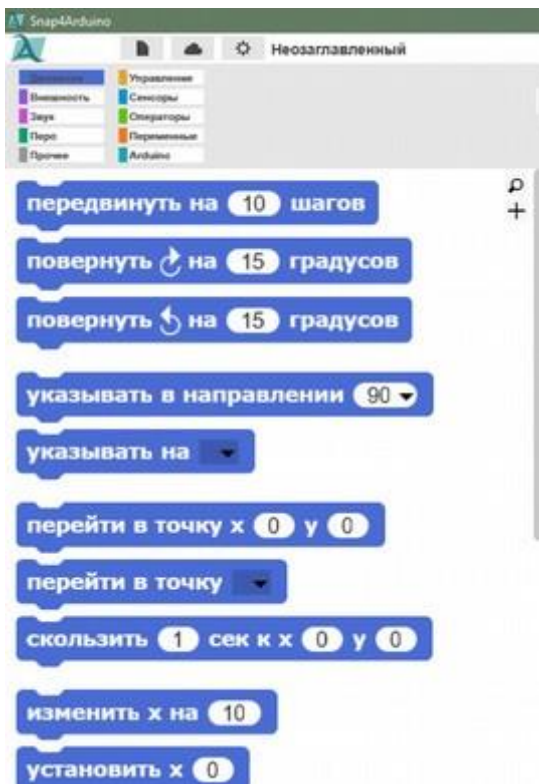


Для создания скрипта алгоритма, который будет выполнять Исполнитель, графические блоки команд из соответствующих ящиков переносятся в поле скриптов, где из них собирается программа.



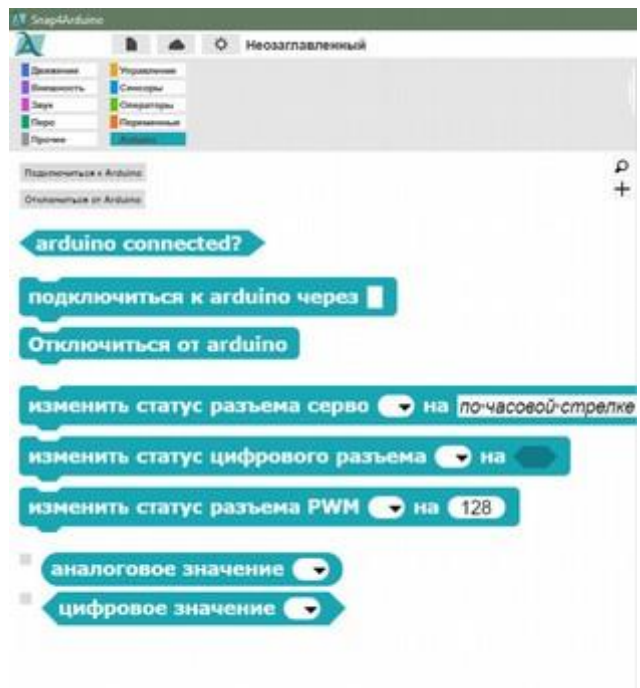
# Ящики команд среды программирования Snap4Arduino

Команды ящика Движение

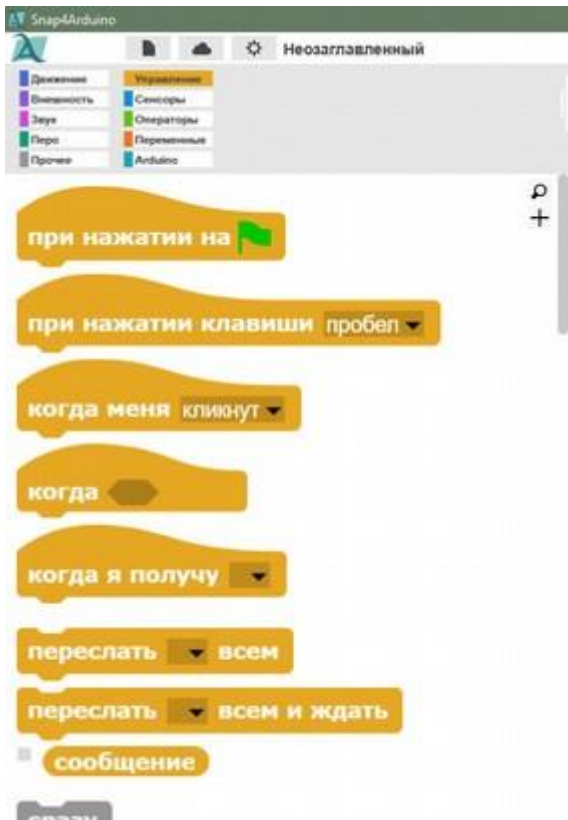


Команды ящика Управление

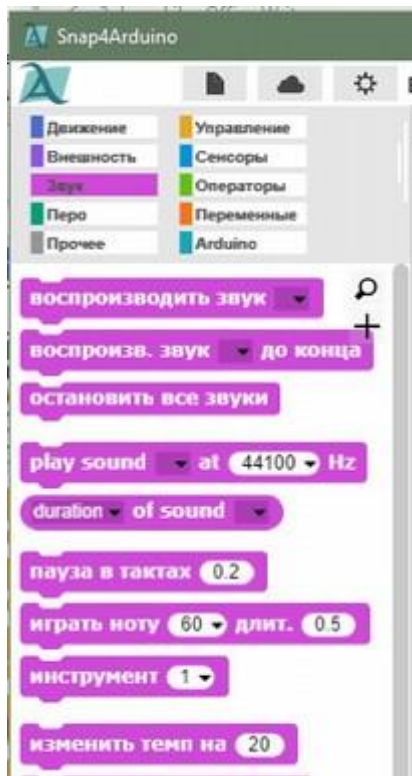
Команды ящика Arduino



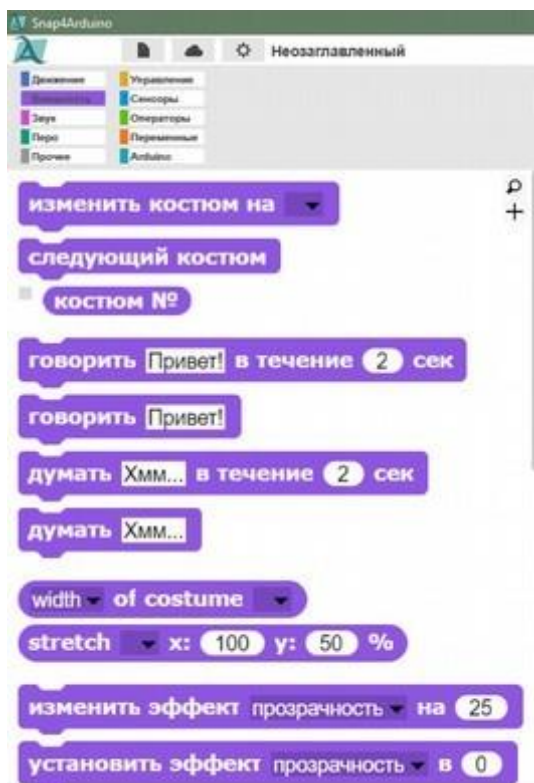
Команды ящика Звук



Команды ящика Операторы



Команды ящика Внешность



## Приложение 2. Как работает 3D ручка

Трехмерная ручка это – ручка, которая выделяет горячий пластик из сопла и может рисовать в трех измерениях. Принцип работы такой: пластиковая нить подается в ручку, под действием высокой температуры пластик расплавляется и вытекает из сопла. Ручку перемещают для создания различных фигур, а расплавленный пластик быстро твердеет.

