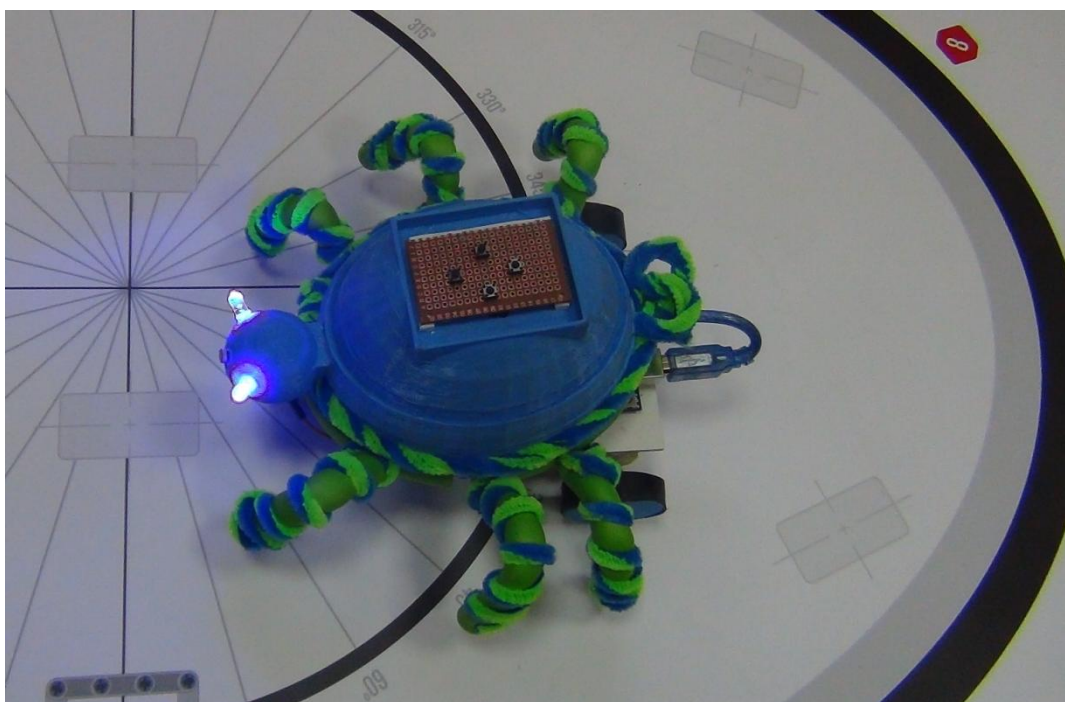


## Проект «Муравей»

Авторы проекта:

1. Колесников Всеволод
2. Дмитрук Юра
3. Каширина Светлана

Научный руководитель: Корягин Андрей Владимирович



Цель: разработка и создание мобильного роботизированного устройства и программного обеспечения для отработки инженерных компетенций и применения в образовательной и исследовательской деятельности.

## **Задачи:**

1. Разработать и воплотить в реальность проект командой детей из разных технических направлений: программирование, 3d моделирование и анимация, электротехника и робототехника.
2. Приблизить этапы создания к реальным производственным этапам разработки и внедрения.
3. Разработать робота с практическими функциональными возможностями: образовательными и исследовательскими.

## **Структура проекта:**

1. Робот – муравей. Выполняет образовательные задачи в области робототехники, математики и биологии
2. Программа «РКонтроль» - программа по управлению роботом с помощью кнопок (образовательный модуль).
3. Программа «Фотопоиск» - программа автономной функциональности робота на примере реакции на свет (исследовательский модуль из жизни насекомых)
4. Программа «Лпоиск» - программа автономной функциональности робота на распознавание меток (исследовательский модуль из жизни насекомых)
5. Программа «ВКонтроль» - программа по дистанционному управлению роботом с использованием USB и Bluetooth соединений. Для Windows и Android (образовательный модуль).
6. Программа «ПиСимулятор» - программа симулятор жизни муравья (образовательный и исследовательский модуль).

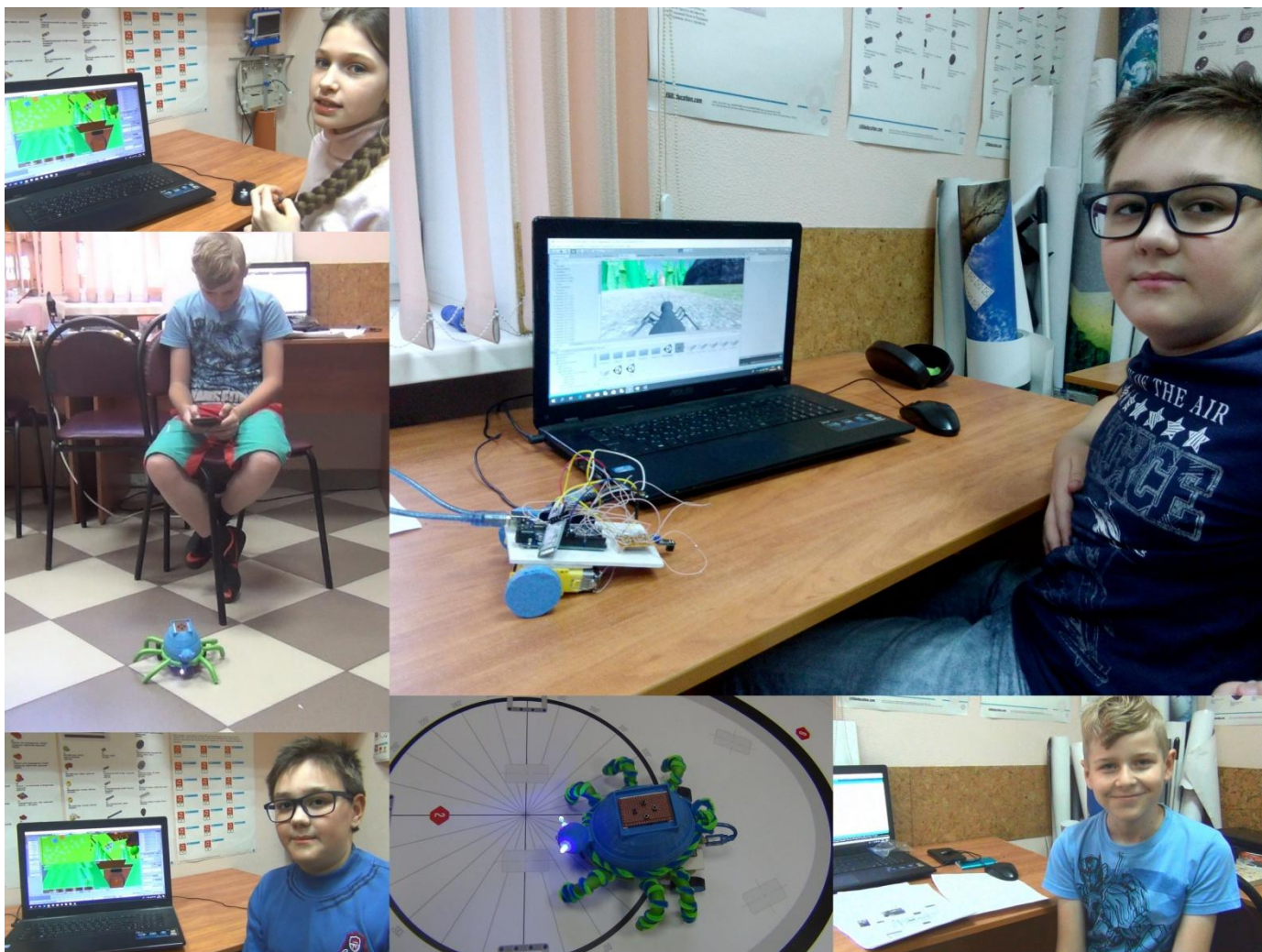
В команде чётко распределены задачи, которые решались в ходе создания робота.

**Колесников Всеволод** – капитан команды. Разработал «ПиСимулятор» и дизайн робота муравья (прототипирование корпуса и головы, дизайн внешности муравья).

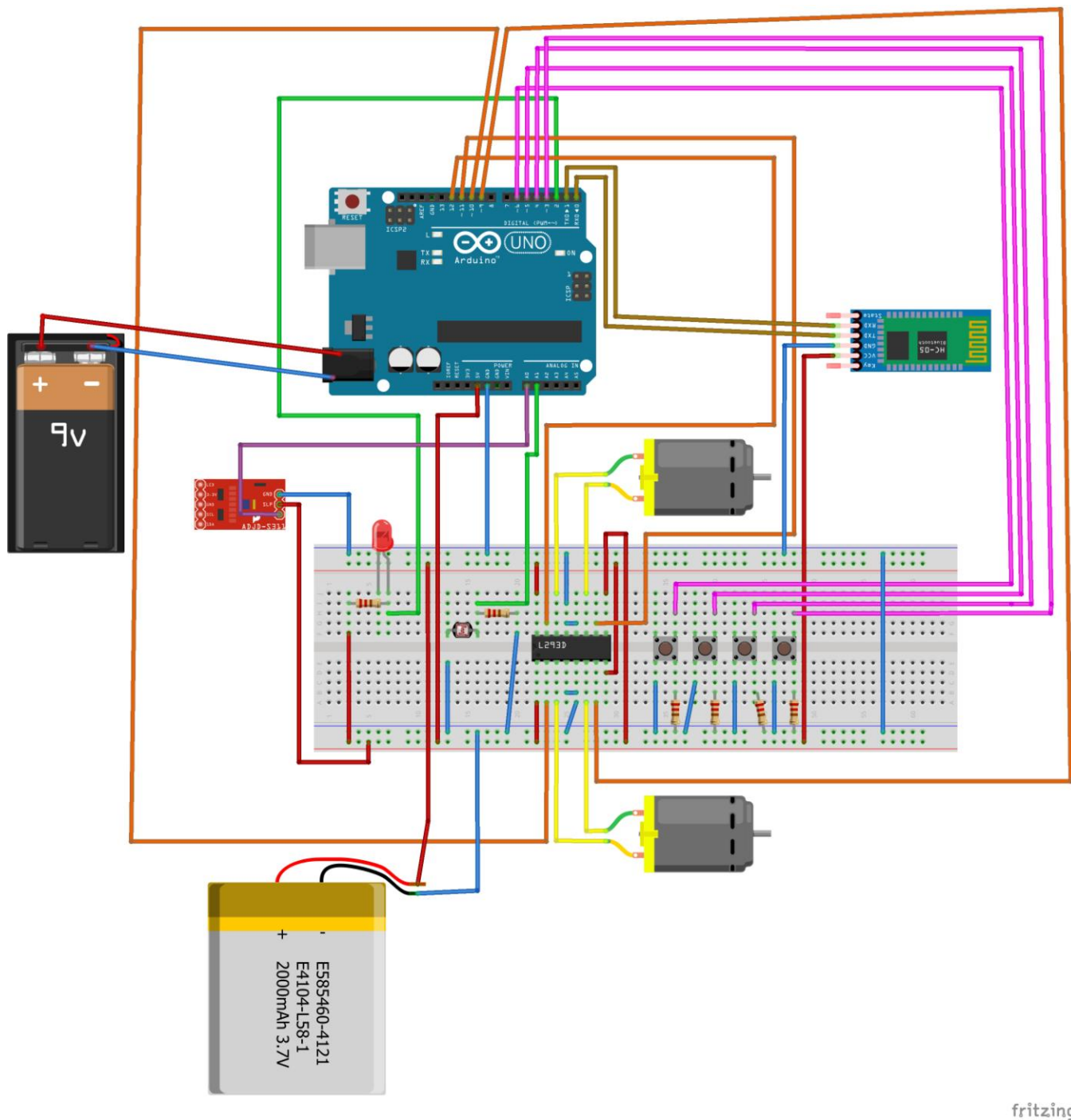
**Каширина Светлана** – программирование. Разработала программы для взаимосвязи программ и устройств управления с роботом «Муравей».

ЦИРИДО «Пифаград»

Дмитрук Юрий – электротехник. Собрал электронную часть робота, создал программы для микроконтроллера arduino.



# Принципиальная схема устройства (Fritzing):



## **Структура работа:**

1. Основная платформа 85 x 120 мм – 1 шт.
2. Электромоторы на 5 Вольт – 1 шт.
3. Опорное колесо (рицинус) – 1 шт.
4. Фоторезистор керамический – 1 шт.
5. Микроконтроллер Arduino UNO – 1 шт.
6. Датчик линии arduino – 1шт.
7. Bluetooth модуль – 1 шт.
8. Кнопки – 4 шт.
9. Светодиоды – 2 шт.
10. Элемент питания (аккумулятор) – 1 шт
11. Микросхема l293D
12. Колёса
13. Корпус

## ОПИСАНИЕ:

### Электроника:

**Фоторезистор GL5528** меняет сопротивление в зависимости от силы падающего на него света. Иначе говоря, это датчик освещенности. Имеет максимальное сопротивление в темноте (т.н. «темновое сопротивление»), спадающее по мере увеличения интенсивности освещения. Может использоваться в качестве аналогового сенсора освещенности для Arduino.

**Драйвер двигателей L293D** - Микросхема включает в себя сразу два драйвера для управления слаботочными моторами. Используя данную микросхему мы можем управлять двигателями с довольно широким диапазоном питающего напряжения от 4.5 до 36 вольт, но при этом, L293D может выдать всего лишь 600mA продолжительного тока нагрузки на каждый канал.

**Arduino Uno** контроллер построен на **ATmega328**. Платформа имеет 14 цифровых вход/выходов (6 из которых могут использоваться как выходы ШИМ), 6 аналоговых входов, кварцевый генератор 16 МГц, разъем USB, силовой разъем, разъем ICSP и кнопку перезагрузки. Для работы необходимо подключить платформу к компьютеру посредством кабеля USB, либо подать питание при помощи адаптера AC/DC или батареи.

Флеш-память	32 Кб (ATmega328) из которых 0.5 Кб используются для загрузчика
ОЗУ	2 Кб (ATmega328)
Тактовая частота	16 МГц

**Светодиод**-это полупроводниковый прибор, преобразующий электрический ток непосредственно в световое излучение.

Так как светодиод является полупроводниковым прибором, то при включении в цепь необходимо соблюдать полярность. Светодиод имеет два вывода, один из которых катод ("минус"), а другой - анод ("плюс").

**Bluetooth HC-06** – модуль для беспроводной передачи и приёма данных. Данный модуль работает на частоте от 2,40 ГГц до 2,48 ГГц

**Датчик линии** - позволяет определять цвет поверхности около него. Основной фотоэлемент работает в инфракрасном спектре. Дополнительно на сенсоре установлен светодиод, который загорается когда поверхность под датчиком светлая. Это удобно для диагностики и настройки.

Переменный резистор, установленный на сенсоре, позволит регулировать чувствительность сенсора в широких пределах.

### **Функциональные возможности проекта:**

Проект «Муравей» применяется для образовательных и исследовательских целей.

В образовательном направлении робот выполняет роль дополнительного звена курса «Алгоритмизация» направления «SMART» для детей от 5 лет – 7 лет. Он является дополнением к образовательным роботам: робомышь, bee bot, blue bot, Ботли, matatalab.

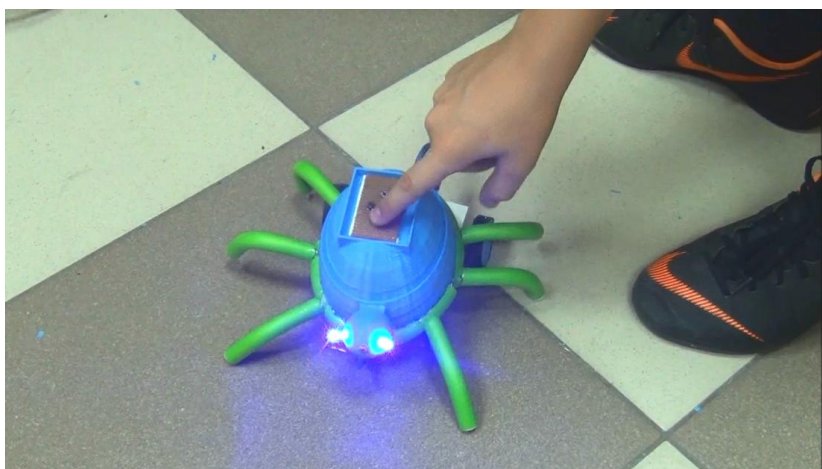
Данное направление и курс запущены в методическом центре «Пифаград» города Воронеж. Курс «Алгоритмизация» является вступлением к направлению «Робототехника».

**Робот «Муравей»** имеет несколько вариантов управления: ручное, дистанционное. Дополнительно, робот обладает автономным режимом работы, опираясь на показания датчиков.

### **Ручное управление:**

Управление роботом с помощью четырёх кнопок: вперёд, назад, поворот налево, поворот направо.

Есть два режима ручного управления. Первый режим управления подразумевает последовательно указывать шаг действия робота, без предварительного запоминания действий. Второй – с запоминанием действий и последовательным проигрыванием их. Они помогают детям использовать робота на разных этапах обучения курса «Алгоритмизация».



### **Дистанционное управление:**

Дистанционное управление роботом осуществляется за счёт передачи приёма данных с помощью Bluetooth. Такая технология позволяет управлять роботом на расстоянии с помощью компьютера, ноутбука, планшета и смартфона поддерживающих Bluetooth и использующих одну из операционных систем: windows, linux, android.

Робот может быть использован на любой проходимой для него поверхности. В основном он применяется совместно с образовательными полями, на которых дети должны решать определённые проблемы по перемещению робота.

Автономный режим работы применяется как в образовательном, так и в исследовательском направлении.



### **Движение по линии.**

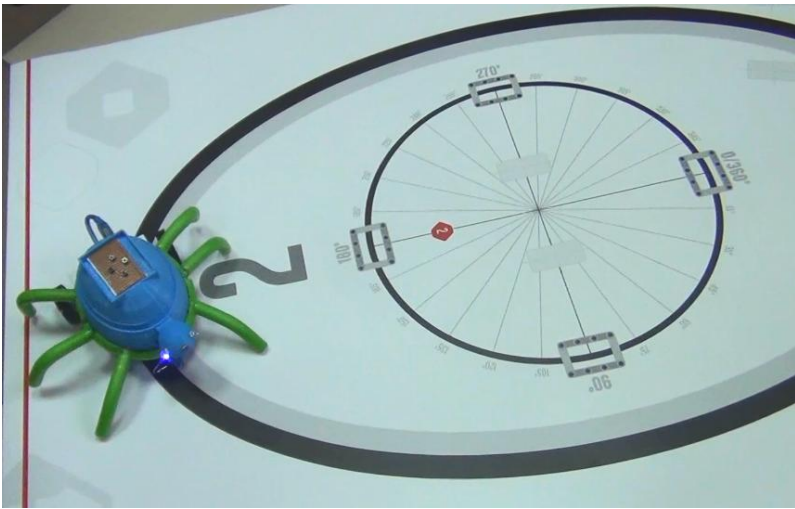
Датчик линии передаёт данные об отражающей способности поверхности на микроконтроллер, где программа обрабатывает их и принимает решения. Дети знакомятся с принципом движения по линии.

В исследовательском направлении данный датчик играет роль аналога анализатора настоящего муравья. Из курса биологии доказано, что высокоорганизованные насекомые, такие как муравьи, ориентируются на особые метки своих сородичей, которые с помощью желёз прокладывают путь, например до пищи.

Муравей разведчик исследует мир и отмечает путь, если есть что то полезное, то он отмечает сильнее. Последующие муравьи отмечают этот путь сильнее и он начинает выделяться на фоне остальных направлений.

Задача учащихся экспериментировать с чувствительностью датчика, количеством путей, их интенсивностью и использование нейронных сетей.





### **Реакция на свет.**

Суточный режим влияет на жизнь насекомых. В первую очередь это временной цикл дня и ночи. Насекомые делятся на дневных и ночных. Муравьи относятся к насекомым с дневным активным циклом. Задача учащихся экспериментировать с чувствительностью датчика с использование нейронных сетей.



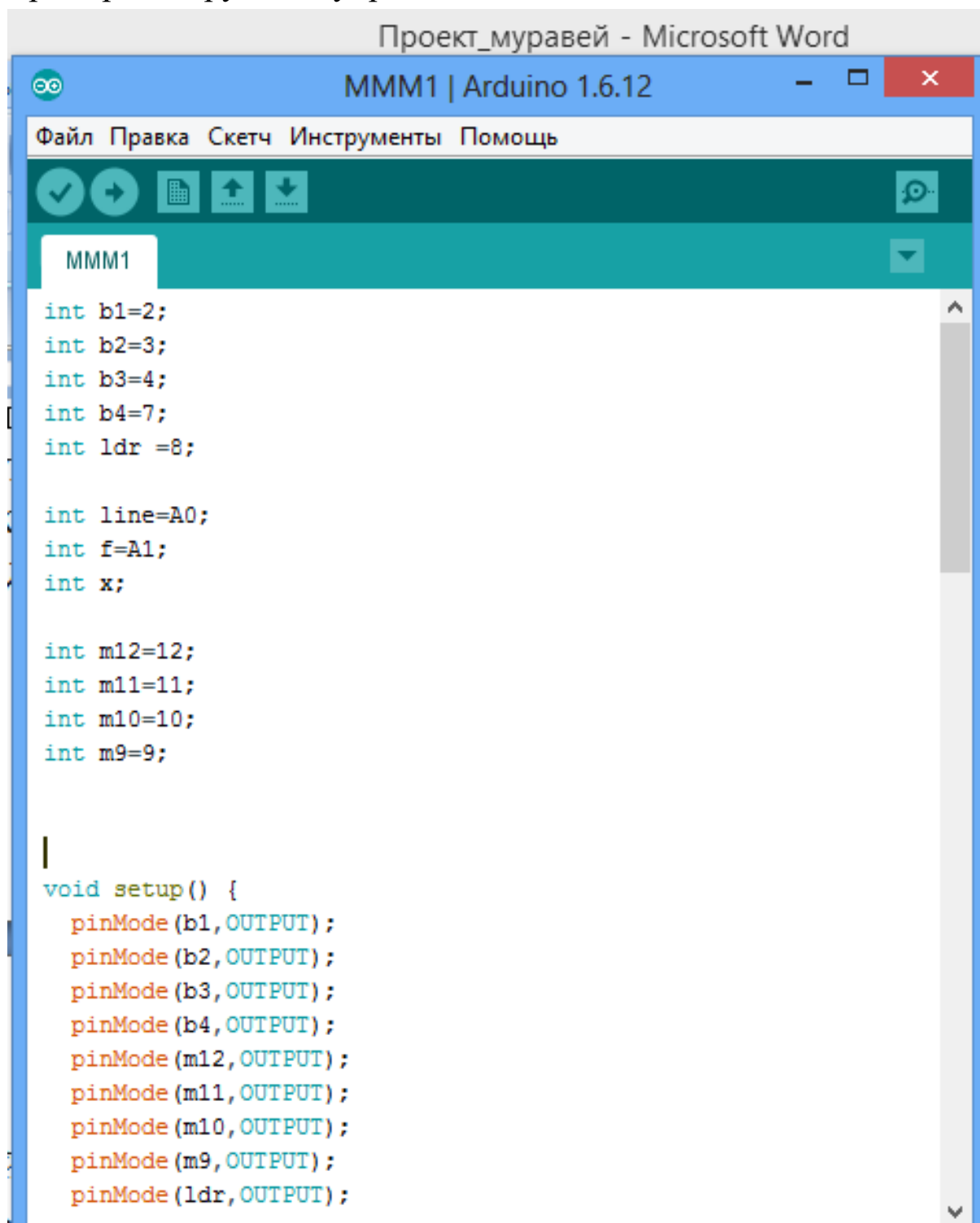
### **Симулятор жизни муравья.**

Разработана программа, создающая виртуальную модель жизни муравья. Эта программа нужна как в образовательных, так и в исследовательских целях.

Учащиеся знакомятся с жизнью насекомого, параллельно отработывая навыки дистанционного управления роботом-аватаром виртуального муравья.

Дети проводят исследование по созданию модели нервной системы муравья, опираясь на знания о нейронных сетях.

Пример кода ручного управления.



The image shows a screenshot of the Arduino IDE interface. The window title is "Проект\_муравей - Microsoft Word". The IDE title bar shows "MMM1 | Arduino 1.6.12". The menu bar includes "Файл", "Правка", "Скетч", "Инструменты", and "Помощь". The toolbar contains icons for a checkmark, a right arrow, a keyboard icon, an upload icon, a download icon, and a search icon. The main editor area shows the following code:

```
MMM1
int b1=2;
int b2=3;
int b3=4;
int b4=7;
int ldr =8;

int line=A0;
int f=A1;
int x;

int m12=12;
int m11=11;
int m10=10;
int m9=9;

void setup() {
  pinMode(b1, OUTPUT);
  pinMode(b2, OUTPUT);
  pinMode(b3, OUTPUT);
  pinMode(b4, OUTPUT);
  pinMode(m12, OUTPUT);
  pinMode(m11, OUTPUT);
  pinMode(m10, OUTPUT);
  pinMode(m9, OUTPUT);
  pinMode(ldr, OUTPUT);
```

```
pinMode(line, INPUT);
pinMode(f, INPUT);
Serial.begin(9600);

digitalWrite(ldr, HIGH);

digitalWrite(m12, LOW);
digitalWrite(m11, LOW);
digitalWrite(m10, LOW);
digitalWrite(m9, LOW);

}

void loop() {

  if (digitalRead(b1) == HIGH) {

digitalWrite(m12, HIGH);
digitalWrite(m11, LOW);
digitalWrite(m10, LOW);
digitalWrite(m9, HIGH);
delay(1000);
}

  if (digitalRead(b2) == HIGH) {

digitalWrite(m12, LOW);
digitalWrite(m11, HIGH);
digitalWrite(m10, HIGH);
digitalWrite(m9, LOW);
delay(1000);
```

```
}  
if (digitalRead(b3) == HIGH) {  
digitalWrite(m12, LOW);  
digitalWrite(m11, HIGH);  
digitalWrite(m10, LOW);  
digitalWrite(m9, HIGH);  
delay(1000);  
}  
if (digitalRead(b4) == HIGH) {  
digitalWrite(m12, HIGH);  
digitalWrite(m11, LOW);  
digitalWrite(m10, HIGH);  
digitalWrite(m9, LOW);  
delay(1000);  
}  
else {  
  
digitalWrite(m12, LOW);  
digitalWrite(m11, LOW);  
digitalWrite(m10, LOW);  
digitalWrite(m9, LOW);  
  
}  
Serial.println(analogRead(line));  
Serial.println(analogRead(f));  
  
delay(300);  
}
```

Пример кода для связи устройств управления и виртуальной среды с роботом.

```
File Edit Format Run Options Window Help
import serial
import time
from directkeys import ReleaseKey, PressKey, w, a, s, d, Q, E, R, T, Y, U, I, O, P

r = serial.Serial('com3', 9600)

while True:
    t=int(r.readline())
    e=time.clock()
    l=round(e,2)
    print(t)
    time.sleep(0.1)
    if t==1:
        PressKey(w)
        time.sleep(0.2)
        PressKey(Q)
    elif t==2:
        PressKey(s)
        time.sleep(0.2)
        PressKey(Q)
    elif t==3:
        PressKey(a)
        time.sleep(0.2)
        PressKey(Q)
    elif t==4:
        PressKey(d)
        time.sleep(0.2)
        PressKey(Q)
    else:
        time.sleep(0.2)
        PressKey(Q)
```

## Программное обеспечение.

1. Arduino ide
2. Python idle
3. Unity 3d
4. Blender 2.79

