



ИТ ШКОЛА SAMSUNG

ИНДИВИДУАЛЬНЫЙ ПРОЕКТ

Управление

микроконтроллерами:

Arduino IO Control

Город: Иркутск

Площадка: МБОУ лицей №2

Учащийся: Вершинин Дмитрий

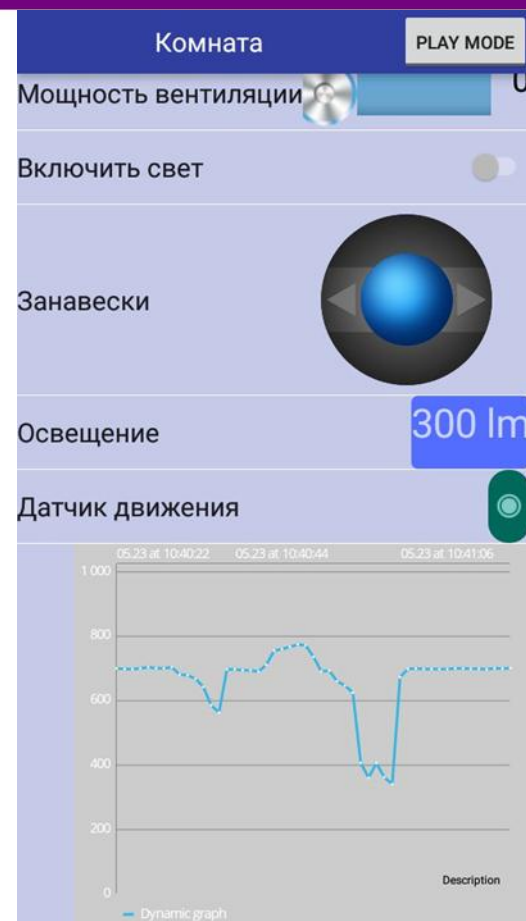
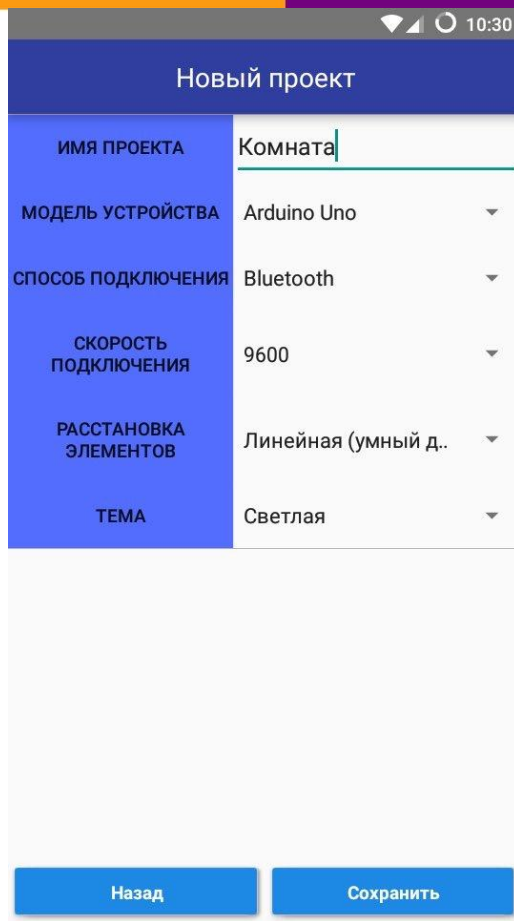
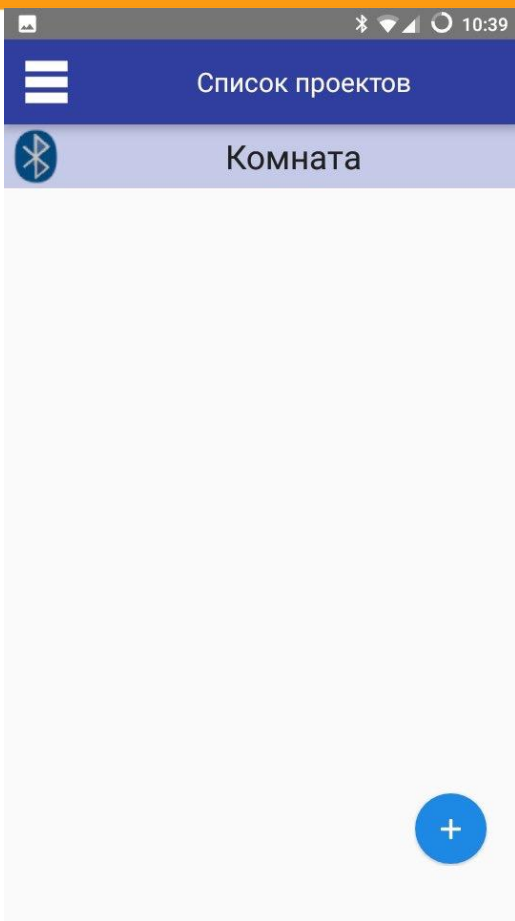
Преподаватель: Реймеров С.Ю.



Arduino IO Control (AIO) – универсальное приложение для управления контроллерами. Фактически, это конструктор, с помощью которого вы создаете удобную и индивидуальную панель управления устройствами, подключенными к контроллеру. Упор в приложении делается на простоте использования и возможности полной настройки под свои нужды.



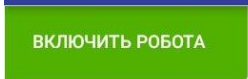
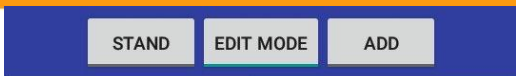
СКРИНШОТЫ



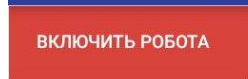
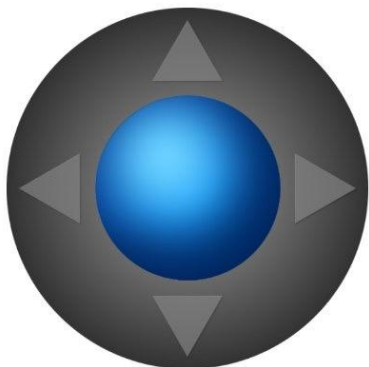
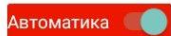


СКРИНШОТЫ

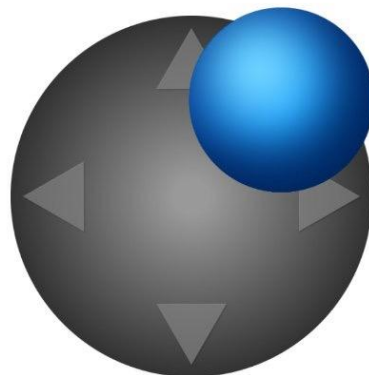




Заполнено: 50%



Заполнено: 50%





ЦЕЛЬ СОЗДАНИЯ

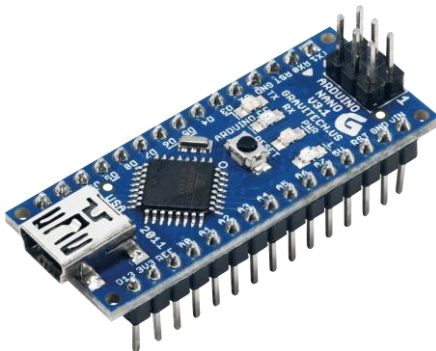
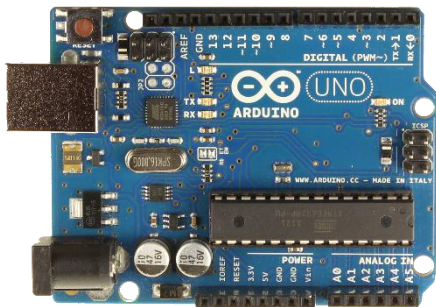
- Упрощение управления контроллерами
- Предоставление возможности каждому без программирования создать пульт для управления Arduino контролером
- Возможность создания сложных проектов, управляемых с помощью смартфона



ДЛЯ ЧЕГО?

Допустим, что у вас есть плата линейки Arduino...
Что вы можете с ее помощью сделать?

- Компоненты умного дома
- Роботы
- Метеостанции
- Музыкальные инструменты
- Сигнализации





КАК ЭТО СДЕЛАТЬ?

Вы придумали, что хотите сделать. Далее, вам необходимо приобрести контроллер, Bluetooth модуль и периферийные устройства, например, модуль реле, сервопривод и датчик температуры. После этого нужно залить скетч (программу на C-подобном языке) на Arduino, который будет обрабатывать команды, приходящие с Android устройства и отправлять данные с определенной периодичностью (пример в конце презентации). Осталось лишь скачать приложение Arduino IO и настроить его так, как вам удобно. Пример такой настройки есть в видеообзоре приложения.



- Приложение выполняет, отправляет и получает данные с и на Arduino-устройство.



- Обмен осуществляется с помощью Bluetooth соединения.

- Главными достоинствами приложения является универсальность и модифицируемость панели управления.



- Полная настройка проекта занимает менее 7 минут.



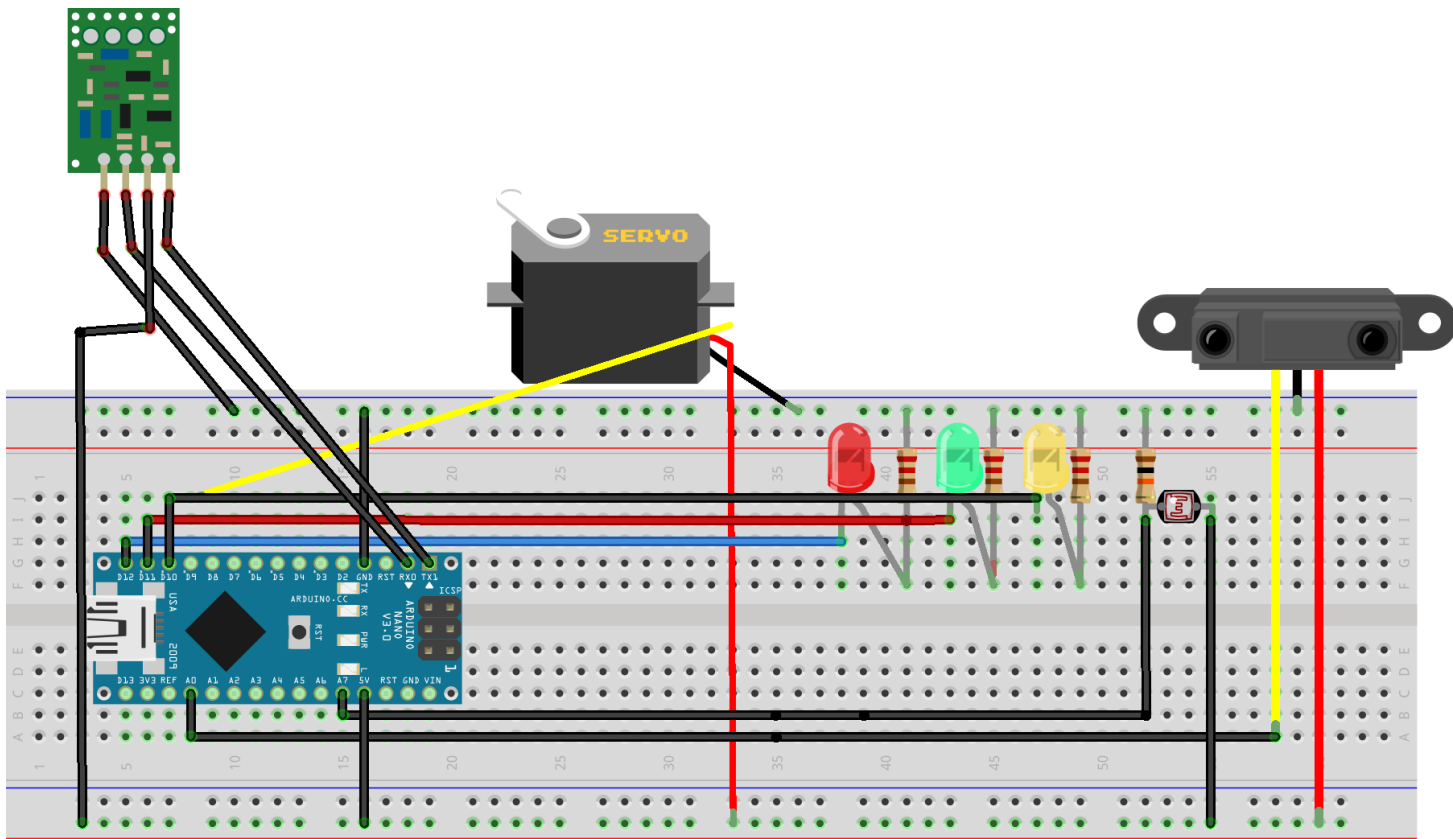
- В приложении 48 классов, из них 13 активностей, более 8000 строчек кода
- 2 таблицы БД: для хранения настроек панелей управления и для хранения их компонентов
- Частичная реализация материального дизайна
- Приложение переведено на английский язык
- Приложение доступно в Google Play (<https://goo.gl/3lr7fF>)



1. Добавление функции автоматического генерирования шаблона скетча
2. Введение возможности управления с помощью интернета и Wi-Fi
3. Добавление функции, которая полностью прячет реализацию
4. Доработка дизайна, включение подсказок по управлению
5. Исправление недочетов в коде
6. Установка защиты от непредвиденных ситуаций
7. Установка защиты от перехвата данных



КОНТРОЛЛЕР И ПЕРЕФЕРИЯ





```
#include <Servo.h>
Servo servo;

int FOTOREZ = A0, SERVO = 13, LED_RED = 11, LED_YELLOW = 12, LED_GREEN = 10, ECHO = 4, TRIG = 5, COOLER = 6;
unsigned long lastmillis = 0;
boolean bool1 = true;

void setup() {
  Serial.begin(9600); // инициализация порта
  servo.attach(SERVO);
  pinMode(LED_YELLOW, OUTPUT);
  pinMode(LED_RED, OUTPUT);
  pinMode(LED_GREEN, OUTPUT);
  pinMode(ECHO, INPUT);
  pinMode(TRIG, OUTPUT);
}

void loop() {
  String s = "";

  while (Serial.available() > 0){
    char c = Serial.read();
    s += c;
    delay(2); //ОБЯЗАТЕЛЬНАЯ ПАУЗА для разделения сообщений
  }

  if (s!=""){

    String key = "", value = ""; //ключ, значение

    int i = 0;

    while (i<s.length()){
      if (s[i]==' ') break;
      key += s[i];
      i++;
    }
    while (i<s.length()){
      value += s[i];
      i++;
    }
  }
}
```

```
int valueInt = value.toInt();
if (key.equals("servoj")) servo.write((valueInt+10)*9);
if (key.equals("servo")) servo.write(valueInt);
if (key.equals("yellow_on")) digitalWrite(LED_YELLOW, HIGH);
if (key.equals("yellow_off")) digitalWrite(LED_YELLOW, LOW);
if (key.equals("red_on")) digitalWrite(LED_RED, HIGH);
if (key.equals("red_off")) digitalWrite(LED_RED, LOW);
if (key.equals("all")){
  analogWrite(LED_RED, valueInt);
  analogWrite(LED_YELLOW, valueInt);
  analogWrite(LED_GREEN, valueInt);
  servo.write((int)(valueInt*0.7)); //поступает 0..255, преобразуется в 0..178
}
if (key.equals("cooler")) analogWrite(COOLER, valueInt);
}

//bool1 служит для того, чтобы отправка шла раздельно - фоторез->дальномер->фоторез->дальномер

if (millis()-lastmillis>100 && bool1){ //фоторезистор sbool1
  int foto = analogRead(FOTOREZ);
  Serial.print("fotorez ");
  Serial.println(foto);
  lastmillis = millis();
  bool1 = false;
}

if (millis()-lastmillis>100){ //дальномер
  int duration, cm;
  digitalWrite(TRIG, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG, LOW);
  duration = pulseIn(ECHO, HIGH);
  cm = duration / 58;
  Serial.print("dalnomer ");
  Serial.print(cm);
  Serial.println(" cm");
  lastmillis = millis();
  bool1 = true;
}
}
```