Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Архитектура вычислительных систем (АВС)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту на тему

«Программирование на Arduino. Механическая рука»

Выполнил: студент группы 353501,

Губский М. Д.

Руководитель: ассистент кафедры информатики, Плехова Т. В.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 4](#_Toc438097973)

[1 Анализ предметной области 5](#_Toc438097974)

[1.1 Микроконтроллер 5](#_Toc438097975)

[2 Используемые технологии 6](#_Toc438097976)

[2.1 Java 6](#_Toc438097977)

[2.2 Android Studio 2.0 6](#_Toc438097978)

[2.3 Arduino 7](#_Toc438097979)

[2.4 Среда разработки Arduino IDE 7](#_Toc438097980)

[2.5 C++ 7](#_Toc438097981)

[3 Проектирование задачи 8](#_Toc438097982)

[1.3 Проектирование архитектуры проекта 8](#_Toc438097983)

[1.3.1 Робо-рука 8](#_Toc438097984)

[1.3.2 Программное обеспечение 8](#_Toc438097985)

[1.4 Используемое оборудование 8](#_Toc438097986)

[2 Описание применения 9](#_Toc438097987)

[2.3 Программное обеспечение 9](#_Toc438097988)

[2.4 Робо-рука 10](#_Toc438097989)

[СПИСОК ИСПОЛЬЗОВАНЫХ ИСТОЧНИКОВ 12](#_Toc438097990)

[1. Программа на Arduino: 13](#_Toc438097991)

[2.1 Класс MainActivity: 18](#_Toc438097992)

[2.2 Класс HandleButtons: 21](#_Toc438097993)

[2.3 Класс передачи данных: 23](#_Toc438097994)

[1. AndroidManifest: 24](#_Toc438097995)

[2. MainActivity: 25](#_Toc438097996)

# ВВЕДЕНИЕ

В повседневном неумолимо развивающемся мире трудно представить нашу жизнь без высоких технологий, машинного производства и удобных и теперь привычных гаджетов. Человек склонен упрощать свою жизнь, перекладывая работу которую может или не может сделать сам. Зачастую, человек по физическим или биологическим причинам не может определенную работу и тогда на помощь приходят технологии.

Робот (чеш. robot, от robota - подневольный труд, rob - раб), машина с антропоморфным (человекоподобным) поведением, которая частично или полностью выполняет функции человека (иногда животного) при взаимодействии с окружающим миром. Первые упоминания о человекоподобных машинах встречаются ещё в древнегреческих мифах. В настоящее время робототехника превратилась в развитую область промышленности: тысячи промышленных роботов работают на различных предприятиях мира, подводные манипуляторы стали непременной принадлежностью подводных исследовательских и спасательных аппаратов, изучение космоса опирается на широкое использование роботов с различным уровнем интеллекта. С развитием робототехники определились 3 разновидности Роботов: с жёсткой программой действий; манипуляторы, управляемые человеком-оператором; с искусственным интеллектом (иногда называемые интегральными), действующие целенаправленно ("разумно") без вмешательства человека. Первые Роботы ("андроиды", имитировавшие движения и внешний облик человека) использовались преимущественно в развлекательных целях. С 30-х гг. в связи с автоматизацией производства Роботы - автоматы стали применять в промышленности наряду с традиционными средствами автоматизации технологических процессов, в частности в мелкосерийном производстве и особенно в цехах с вредными условиями труда.

Промышленный Робот манипулятор имеет "механическую руку" (одну или несколько) и вынесенный пульт управления или встроенное устройство программного управления, реже ЭВМ. Он может, например, перемещать. Промышленные Роботы - автоматы имеют преимущество перед человеком в скорости и точности выполнения ручных однообразных операций. Наиболее распространены Роботы манипуляторы с дистанционным управлением и "механической рукой", закрепленной на подвижном или неподвижном основании. Роботы манипуляторы используют для работы в условиях относительной недоступности либо в опасных, вредных для человека условиях, например в атомной промышленности.

## Анализ предметной области

## 1.1 Микроконтроллер

Микроконтро́ллер (англ. Micro Controller Unit, MCU) — микросхема, предназначенная для управления электронными устройствами.

Типичный микроконтроллер сочетает на одном кристалле функции процессора и периферийных устройств, содержит ОЗУ и (или) ПЗУ. По сути, это однокристальный компьютер, способный выполнять относительно простые задачи.

Отличается от микропроцессора интегрированными в микросхему устройствами ввода-вывода, таймерами и другими периферийными устройствами.

При проектировании микроконтроллеров приходится соблюдать компромисс между размерами и стоимостью с одной стороны и гибкостью и производительностью с другой. Для разных приложений оптимальное соотношение этих и других параметров может различаться очень сильно. Поэтому существует огромное количество типов микроконтроллеров, отличающихся архитектурой процессорного модуля, размером и типом встроенной памяти, набором периферийных устройств, типом корпуса и т. д. В отличие от обычных компьютерных микропроцессоров, в микроконтроллерах часто используется гарвардская архитектура памяти, то есть раздельное хранение данных и команд в ОЗУ и ПЗУ соответственно.

**1.2 Обзор аналогов**

В настоящее время существует множество аналогов. В основном они применяются в промышленности, но существуют, предназначенные для бытовых целей. С их помощью можно перемещать объекты недоступные человеку по физическим или пространственным причинам. Некоторые из них имеют возможность удаленного доступа. В связи с тем, что для транспортировки грузов требуется большая грузоподъемность, что приводит к крепким и дорогостоящим материалом, также недостатком будут габаритность, а в следствии трудности при транспортировке. Данный проект обходит эти недостатки в ущерб грузоподъемности.

* 1. **Постановка задачи**

В связи с тем, что приведенные аналоги слишком дороги, габаритны и неудобны при транспортировке, было решено избавиться от этих недостатков. Цель курсовой – создание механического захвата, а так же платформы для управления им.

# 

# 2 Используемые технологии

Приложение реализовано на языке Java с использованием платформы Android и на С++ для Arduino. В качестве среды разработки были выбраны AndroidStudio 2.0 и стандартная среда разработки Arduino IDE.

### Java

Самой выдающейся чертой языка Java является то, что его приложения компилируются в специальный байт-код и выполняются в специальной виртуальной машине в любой операционной системе и на любом оборудовании. На этом языке программирования могут создаваться веб-приложения для [London escort service](http://www.supreme69.com/), мобильные приложения или программы, "удобочитаемые" на любом компьютере. К уже реализованным всемирно известным проектам на Java относятся Amazon, eBay, Yandex.

Другим важным преимуществом языка является довольно высокая безопасность. Поскольку все приложения выполняются и, соответственно, контролируются Java-машиной, то при любой попытке несанкционированного доступа или соединения программа просто прерывает свою работу. Некоторые программисты выделяют и более упрощённую саму форму написания программ Java по сравнению с её предшественниками, коими являются С, С+ и С++. Помимо этого к языку существует огромное количество бесплатных программных библиотек, написанных программистами всего мира.

К недостаткам языка следует отнести чрезмерную нагрузку на оперативную память оборудования. По некоторым данным и время выполнения одних и тех же задач в полтора-два раза у продуктов этого языка медленнее, чем на С. Тем не менее, преимущества языка Java перевешивают его недостатки, он остаётся одним из самых востребованных.

### Android Studio 2.0

**Android Studio —** это интегрированная среда разработки(IDE) для работы с платформой Android. Android Studio основана на программном обеспечении IntelliJIDEA От компании JetBrains, а также является официальным средством разработки Android приложений. Данная среда разработки доступна для Windows, OS X и Linux.

В новой версии Android Studio 2.0 был встроен новый эмулятор android. Который, по словам google, «на хорошем железе работает быстрее физического устройства». Также была введена новая функция «instant run», которая позволяет почти мгновенно обновлять запущенное под отладкой мобильное приложение при изменении исходного кода (еще не поддерживается). По словам Stephanie Cuthbertson, одной из руководителей направления Android Studio, новая версия Android Studio собирает приложения в два — два с половиной раза быстрее чем предыдущая версия. В новой версии доступна preview версия профилировщика рендера средствами GPU. С помощью этого инструмента разработчики могут анализировать время рендеринга элементов в каждом кадре и находить «узкие места».

### 

### 2.3 Arduino

Arduino — торговая марка аппаратно-программных средств для построения простых систем автоматики и робототехники, ориентированная на непрофессиональных пользователей. Программная часть состоит из бесплатной программной оболочки (IDE) для написания программ, их компиляции и программирования аппаратуры. Аппаратная часть представляет собой набор смонтированных печатных плат, продающихся как официальным производителем, так и сторонними производителями. Полностью открытая архитектура системы позволяет свободно копировать или дополнять линейку продукции Ардуино.

Arduino может использоваться как для создания автономных объектов автоматики, так и подключаться к программному обеспечению на компьютере через стандартные проводные и беспроводные интерфейсы.

### 

### 2.4 Среда разработки Arduino IDE

Среда разработки Arduino состоит из встроенного текстового редактора программного кода, области сообщений, окна вывода текста(консоли), панели инструментов с кнопками часто используемых команд и нескольких меню. Для загрузки программ и связи среда разработки подключается к аппаратной части Arduino.

### C++

C++ — компилируемый, статически типизированный язык программирования общего назначения.

Поддерживает такие парадигмы программирования как процедурное программирование, объектно-ориентированное программирование, обобщённое программирование, обеспечивает модульность, раздельную компиляцию, обработку исключений, абстракцию данных, объявление типов (классов) объектов, виртуальные функции. Стандартная библиотека включает, в том числе, общеупотребительные контейнеры и алгоритмы. C++ сочетает свойства как высокоуровневых, так и низкоуровневых языков. В сравнении с его предшественником — языком C, — наибольшее внимание уделено поддержке объектно-ориентированного и обобщённого программирования.

# Проектирование задачи

## Проектирование архитектуры проекта

Весь проект делится на две основные части: робо-рука и ПО. Рассмотрим каждую часть отдельно.

## Робо-рука??????????????

Робо-рука состоит из 6 двигателей, вращающих каждый свою ось.

### Программное обеспечение

ПО представлено в виде программы с графическим интерфейсом. Программа имеет 3 основных класса:

* MainActivity – он отвечает за отрисовку окна с пользовательским интерфейсом, а также подключение к устройству
* HandleButton – класс, отвечающий за обработку нажатия кнопок.
* ProcessingData – класс, отвечающий за передачу данных.

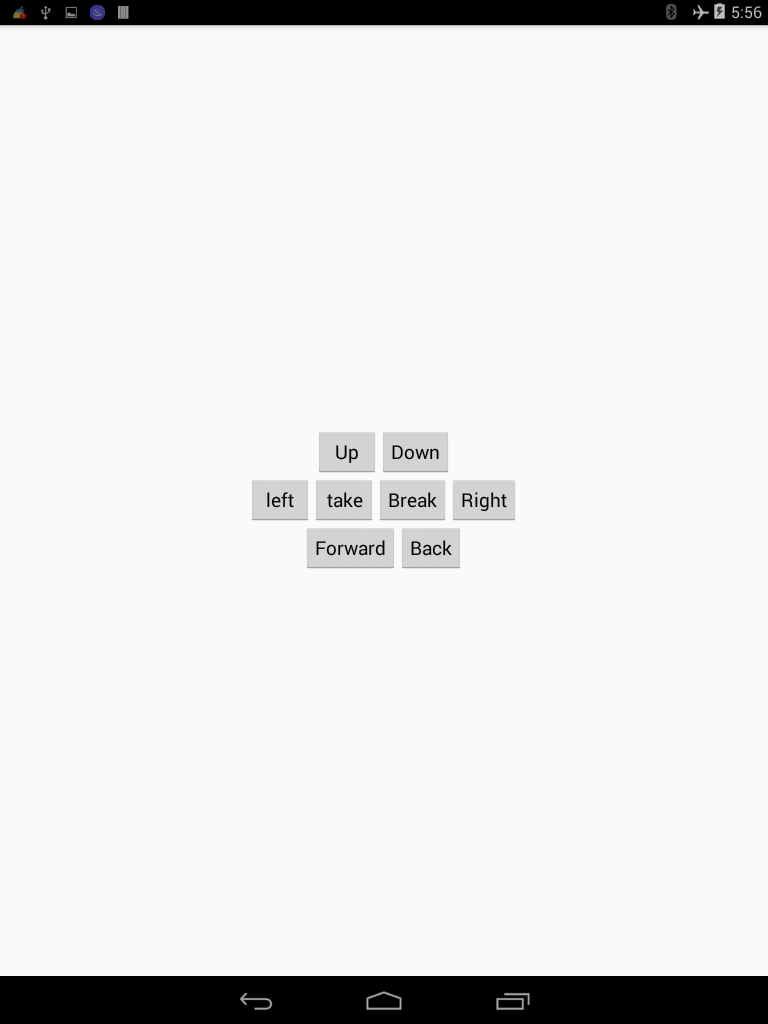
# Используемое оборудование

Для создания Руки были использованы следующие элементы:

* Микроконтроллер Arduino Uno
* USB провод для связи Arduino и ПК
* 5x Сервоприводов на 9г для управления рукой
* Микропривод для захвата
* Макетная плата
* Bluetooth modul HC-06
* Различные строительные элементы, крепежи, провода

# Описание применения

## Программное обеспечение



Из названия соответственно следует действие робо-руки. Также поддерживается мульти-тач.

## Робо-рука???????

**ЗАКЛЮЧЕНИЕ**

В результате работы над курсовым проектом была разработана механическая рука и программный продукт для управления ею. В процессе написания проекта была изучены базовые принципы программирования микроконтроллеров, на примере Arduino.

В дальнейшей работе планируется развивать проект в следующих направлениях:

1. Замена Серводвигателей более мощными.
2. Усовершенствование захвата.

# СПИСОК ИСПОЛЬЗОВАНЫХ ИСТОЧНИКОВ

[1] Startandroid [Электронный ресурс]. – документ по multi-touch. – Режим доступа : <http://startandroid.ru/ru/>

[2] Arduino [Электронный ресурс]. – документация по Arduino. – Режим доступа : <http://arduino.ru/>

[3] Habrahabr [Электронный ресурс]. – полезные статьи по созданию руки. – Режим доступа : <http://habrahabr.ru/>.

[4] Википедия [Интернет ресурс]. – Электронные данные. – Режим доступа: <http://ru.wikipedia.org/wiki/>.

[5] *Герберт Шилдт.* Java 8. Полное руководство, 9-е издание = Java 8. The Complete

Reference, 9th Edition. — М.: [«Вильямс»](https://ru.wikipedia.org/w/index.php?title=%D0%92%D0%B8%D0%BB%D1%8C%D1%8F%D0%BC%D1%81_(%D0%B8%D0%B7%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE)&action=edit&redlink=1), 2015. — 1376 с. — [ISBN 978-5-8459-1918-2](https://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D1%83%D0%B6%D0%B5%D0%B1%D0%BD%D0%B0%D1%8F:%D0%98%D1%81%D1%82%D0%BE%D1%87%D0%BD%D0%B8%D0%BA%D0%B8_%D0%BA%D0%BD%D0%B8%D0%B3/9785845919182).

[6] *Брюс Эккель.* Философия Java = Thinking in Java. — 3-е изд. — СПб.: [Питер](https://ru.wikipedia.org/wiki/%D0%9F%D0%B8%D1%82%D0%B5%D1%80_(%D0%B8%D0%B7%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE)), 2003. — 976 с. — [ISBN 5-88782-105-1](https://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D1%83%D0%B6%D0%B5%D0%B1%D0%BD%D0%B0%D1%8F:%D0%98%D1%81%D1%82%D0%BE%D1%87%D0%BD%D0%B8%D0%BA%D0%B8_%D0%BA%D0%BD%D0%B8%D0%B3/5887821051).

**ПРИЛОЖЕНИЯ**

### Программа на Arduino:

#include <Servo.h>

char incomingByte; // входящие данные

Servo RotatoryServo;

Servo FirstStepLeftServo;

Servo FirstStepRightServo;

Servo SecondStepServo;

Servo ThirdStepServo;

Servo ClampServo;

int mySignals[8]={0,0,0,0,0,0,0,0};

int CurrentAngle[6]={90,0,180,0,0,0};//4-down; 4-down;

void setup() {

Serial.begin(9600); // инициализация порта

pinMode(LED, OUTPUT);

RotatoryServo.attach(8);

FirstStepLeftServo.attach(3);

FirstStepRightServo.attach(4);

SecondStepServo.attach(5);

ThirdStepServo.attach(6);

ClampServo.attach(7);

}

void loop() {

if (Serial.available() > 0) { //если пришли данные

incomingByte = Serial.read(); // считываем байт

Serial.println(incomingByte);

if(isLowerCase(incomingByte))

{

int i=incomingByte-97;

mySignals[i]=0;

}else{

int i=incomingByte-65;

if(i % 2!=1 && mySignals[i+1]!=1)

mySignals[i]=1;

else if(i % 2==1 && mySignals[i-1]!=1)

mySignals[i]=1;

}

}

ProcessingSignals(mySignals);

delay(100);

}

void ProcessingSignals(int\* mySignals){

GoBreak(mySignals[0]);

GoTake(mySignals[1]);

GoDown(mySignals[2]);

GoUp(mySignals[3]);

GoRight(mySignals[4]);

GoLeft(mySignals[5]);

GoBack(mySignals[6]);

GoForward(mySignals[7]);

}

void GoBreak(int signal)

{

if(signal)

{

ClampServo.write(20);

delay(400);

}else{

ClampServo.write(180);

delay(400);

}

}

void GoTake(int signal)

{

if(signal)

{

GoBreak(1);

GoBreak(0);

mySignals[1]=0;

}

}

void GoDown(int signal)

{

if(signal){

if(CurrentAngle[1]>0)

{

if(CurrentAngle[3]>0)

{

if(CurrentAngle[4]>0)

{

CurrentAngle[4]-=10;

ThirdStepServo.write(CurrentAngle[4]);

}

}else{

CurrentAngle[3]-=10;

SecondStepServo.write(CurrentAngle[3]);

}

}

else

{

CurrentAngle[1]-=10;

FirstStepLeftServo.write(CurrentAngle[1]);

CurrentAngle[2]+=10;

FirstStepRightServo.write(CurrentAngle[2]);

}

}

void GoUp(int signal)

{

if(signal){

if(CurrentAngle[1]>90)

{

if(CurrentAngle[3]<90)

{

if(CurrentAngle[4]<90)

{

CurrentAngle[4]+=10;

ThirdStepServo.write(CurrentAngle[4]);

}

}else{

CurrentAngle[3]+=10;

SecondStepServo.write(CurrentAngle[3]);

}

}

else

{

CurrentAngle[1]+=10;

FirstStepLeftServo.write(CurrentAngle[1]);

CurrentAngle[2]-=10;

FirstStepRightServo.write(CurrentAngle[2]);

}

}}

void GoRight(int signal)

{

if(signal && CurrentAngle[0]>0)

{

CurrentAngle[0]-=20;

RotatoryServo.write(CurrentAngle[0]);

}else if(signal)

{

mySignals[4]=0;

}

}

void GoLeft(int signal)

{

if(signal && CurrentAngle[0]<160)

{

CurrentAngle[0]+=20;

RotatoryServo.write(CurrentAngle[0]);

}else if(signal)

{

mySignals[5]=0;

}

}

void GoBack(int signal)

{

if(signal){

if(CurrentAngle[1]>60)

{

if(CurrentAngle[3]<10)

{

if(CurrentAngle[4]<140)

CurrentAngle[4]+=10;

else

CurrentAngle[4]-=10;

ThirdStepServo.write(CurrentAngle[4]);

}else{

CurrentAngle[3]-=10;

SecondStepServo.write(CurrentAngle[3]);

}

}

else

{

CurrentAngle[1]+=10;

FirstStepLeftServo.write(CurrentAngle[1]);

CurrentAngle[2]-=10;

FirstStepRightServo.write(CurrentAngle[2]);

}

}

}

void GoForward(int signal)

{

if(signal){

if(CurrentAngle[1]>60)

{

if(CurrentAngle[3]<10)

{

if(CurrentAngle[4]<140)

CurrentAngle[4]+=10;

else

CurrentAngle[4]-=10;

ThirdStepServo.write(CurrentAngle[4]);

}else{

CurrentAngle[3]-=10;

SecondStepServo.write(CurrentAngle[3]);

}

}

}

}

### Класс MainActivity:

package com.nox.myapplication;

import android.app.Activity;

import android.bluetooth.BluetoothAdapter;

import android.bluetooth.BluetoothDevice;

import android.bluetooth.BluetoothSocket;

import android.content.Intent;

import android.os.Bundle;

import android.util.Log;

import android.widget.Button;

import android.widget.Toast;

import java.io.IOException;

import java.io.OutputStream;

import java.util.UUID;

public class MainActivity extends Activity {

private static final String TAG = "bluetooth1";

Button btnTake, btnBreak, btnRight, btnLeft, btnUp, btnDown, btnForward, btnBack;

HandleButtons HB;

ProcessingData PD;

int answer = 0;

private static final int REQUEST\_ENABLE\_BT = 1;

private BluetoothAdapter btAdapter = null;

private BluetoothSocket btSocket = null;

private OutputStream outStream = null;

// SPP UUID сервиса

private static final UUID MY\_UUID = UUID.fromString("00001101-0000-1000-8000-00805F9B34FB");

// MAC-адрес Bluetooth модуля

private static String address = "98:D3:31:40:3A:0A";

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

btnTake = (Button) findViewById(R.id.buttonTake);

btnBreak = (Button) findViewById(R.id.buttonBreak);

btnBack = (Button) findViewById(R.id.buttonBack);

btnForward = (Button) findViewById(R.id.buttonForward);

btnDown = (Button) findViewById(R.id.buttonDown);

btnUp = (Button) findViewById(R.id.buttonUp);

btnLeft = (Button) findViewById(R.id.btnLeft);

btnRight = (Button) findViewById(R.id.btnRight);

btAdapter = BluetoothAdapter.getDefaultAdapter();

checkBTState();

PD = new ProcessingData(outStream, MY\_UUID);

HB = new HandleButtons(PD, btnBack, btnForward, btnBreak, btnTake, btnDown, btnUp, btnLeft, btnRight);

HB.createButtonsHandlers();

}

@Override

public void onResume() {

super.onResume();

Log.d(TAG, "...onResume - попытка соединения...");

// Set up a pointer to the remote node using it's address.

BluetoothDevice device = btAdapter.getRemoteDevice(address);

// Two things are needed to make a connection:

// A MAC address, which we got above.

// A Service ID or UUID. In this case we are using the

// UUID for SPP.

try {

btSocket = device.createRfcommSocketToServiceRecord(MY\_UUID);

} catch (IOException e) {

errorExit("Fatal Error", "In onResume() and socket create failed: " + e.getMessage() + ".");

}

// Discovery is resource intensive. Make sure it isn't going on

// when you attempt to connect and pass your message.

btAdapter.cancelDiscovery();

// Establish the connection. This will block until it connects.

Log.d(TAG, "...Соединяемся...");

try {

btSocket.connect();

Log.d(TAG, "...Соединение установлено и готово к передачи данных...");

Toast.makeText(getApplicationContext(), "Connected", Toast.LENGTH\_LONG).show();

} catch (IOException e) {

try {

btSocket.close();

} catch (IOException e2) {

errorExit("Fatal Error", "In onResume() and unable to close socket during connection failure" + e2.getMessage() + ".");

}

}

// Create a data stream so we can talk to server.

Log.d(TAG, "...Создание Socket...");

try {

outStream = btSocket.getOutputStream();

} catch (IOException e) {

errorExit("Fatal Error", "In onResume() and output stream creation failed:" + e.getMessage() + ".");

}

}

@Override

public void onPause() {

super.onPause();

Log.d(TAG, "...In onPause()...");

if (outStream != null) {

try {

outStream.flush();

} catch (IOException e) {

errorExit("Fatal Error", "In onPause() and failed to flush output stream: " + e.getMessage() + ".");

}

}

try {

btSocket.close();

} catch (IOException e2) {

errorExit("Fatal Error", "In onPause() and failed to close socket." + e2.getMessage() + ".");

}

}

private void checkBTState() {

// Check for Bluetooth support and then check to make sure it is turned on

// Emulator doesn't support Bluetooth and will return null

if (btAdapter == null) {

errorExit("Fatal Error", "Bluetooth не поддерживается");

} else {

if (btAdapter.isEnabled()) {

Log.d(TAG, "...Bluetooth включен...");

} else {

//Prompt user to turn on Bluetooth

Intent enableBtIntent = new Intent(BluetoothAdapter.ACTION\_REQUEST\_ENABLE);

startActivityForResult(enableBtIntent, REQUEST\_ENABLE\_BT);

}

}

}

private void errorExit(String title, String message) {

Toast.makeText(getBaseContext(), title + " - " + message, Toast.LENGTH\_LONG).show();

finish();

}

}

### Класс HandleButtons:

package com.nox.myapplication;

import android.view.MotionEvent;

import android.view.View;

import android.widget.Button;

public class HandleButtons implements View.OnTouchListener {

Button btnTake, btnBreak, btnRight, btnLeft,

btnUp, btnDown, btnForward, btnBack;

ProcessingData PD;

public HandleButtons(ProcessingData PD, Button btnBack,

Button btnForward, Button btnBreak, Button btnTake,

Button btnDown, Button btnUp, Button btnLeft,

Button btnRight) {

this.btnBack = btnBack;

this.btnForward = btnForward;

this.btnBreak = btnBreak;

this.btnTake = btnTake;

this.btnDown = btnDown;

this.btnUp = btnUp;

this.btnLeft = btnLeft;

this.btnRight = btnRight;

this.PD = PD;

}

void createButtonsHandlers() {

btnDown.setOnTouchListener(this);

btnUp.setOnTouchListener(this);

btnLeft.setOnTouchListener(this);

btnRight.setOnTouchListener(this);

btnTake.setOnTouchListener(this);

btnBreak.setOnTouchListener(this);

btnForward.setOnTouchListener(this);

btnBack.setOnTouchListener(this);

}

@Override

public boolean onTouch(View v, MotionEvent event) {

int answer = 0;

switch (v.getId()) {

case R.id.btnLeft:

answer = 102;

break;

case R.id.btnRight:

answer = 101;

break;

case R.id.buttonBack:

answer = 103;

break;

case R.id.buttonForward:

answer = 104;

break;

case R.id.buttonBreak:

answer = 97;

break;

case R.id.buttonTake:

answer = 98;

break;

case R.id.buttonUp:

answer = 100;

break;

case R.id.buttonDown:

answer = 99;

break;

}

switch (event.getAction()) {

case MotionEvent.ACTION\_DOWN: // нажатие

answer -= 32;

PD.sendData(Character.toString((char) answer));

break;

case MotionEvent.ACTION\_UP: // отпускание

case MotionEvent.ACTION\_CANCEL:

PD.sendData(Character.toString((char) answer));

break;

}

return false;

}

}

### Класс передачи данных:

package com.nox.myapplication;

import java.io.OutputStream;

import android.util.Log;

public class ProcessingData{

private OutputStream outStream;

private static final UUID MY\_UUID

public ProcessingData(OutputStream outStream,UUID MY\_UUID){ {

this.outStream=outStream;

this.MY\_UUID=MY\_UUID;

}

public void sendData(String message) {

byte[] msgBuffer = message.getBytes();

Log.d(TAG, "...Посылаем данные: " + message + "...");

try {

outStream.write(msgBuffer);

} catch (IOException e) {

String msg = "In onResume() and an exception occurred during write: " + e.getMessage();

if (address.equals("00:00:00:00:00:00"))

msg = msg + ".\n\nВ переменной address у вас прописан 00:00:00:00:00:00, вам необходимо прописать реальный MAC-адрес Bluetooth модуля";

msg = msg + ".\n\nПроверьте поддержку SPP UUID: " + MY\_UUID.toString() + " на Bluetooth модуле, к которому вы подключаетесь.\n\n";

errorExit("Fatal Error", msg);

}

}

}

### AndroidManifest:

<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="com.nox.myapplication"

android:installLocation="preferExternal" >

<uses-permission android:name="android.permission.BLUETOOTH" />

<uses-permission android:name="android.permission.BLUETOOTH\_ADMIN" />

<application

android:allowBackup="true"

android:icon="@mipmap/ic\_launcher"

android:label="@string/app\_name"

android:supportsRtl="true"

android:theme="@style/AppTheme">

<activity android:name=".MainActivity">

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

</application>

</manifest>

### MainActivity:

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:paddingBottom="@dimen/activity\_vertical\_margin"

android:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

android:paddingTop="@dimen/activity\_vertical\_margin"

tools:context="com.nox.myapplication.MainActivity">

<LinearLayout

android:orientation="vertical"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:gravity="center\_vertical">

<LinearLayout

android:orientation="horizontal"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:gravity="center\_horizontal">

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Up"

android:id="@+id/buttonUp" />

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Down"

android:id="@+id/buttonDown" />

</LinearLayout>

<LinearLayout

android:orientation="horizontal"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:gravity="center\_horizontal">

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="left"

android:id="@+id/btnLeft"

android:layout\_below="@+id/textView"

android:layout\_toRightOf="@+id/textView"

android:layout\_toEndOf="@+id/textView" />

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="take"

android:id="@+id/buttonTake"

android:layout\_below="@+id/btnLeft"

android:layout\_alignRight="@+id/txtArduino"

android:layout\_alignEnd="@+id/txtArduino" />

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Break"

android:id="@+id/buttonBreak" />

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Right"

android:id="@+id/btnRight"

android:layout\_alignTop="@+id/btnLeft"

android:layout\_toRightOf="@+id/btnLeft"

android:layout\_toEndOf="@+id/btnLeft" />

</LinearLayout>

<LinearLayout

android:orientation="horizontal"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:gravity="center\_horizontal">

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Forward"

android:id="@+id/buttonForward" />

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Back"

android:id="@+id/buttonBack" />

</LinearLayout>

</LinearLayout>

</RelativeLayout>