

---

## Проект: «Андроид-Мышь»

### Введение

Проект Андроид-Мышь позволяет использовать мобильный телефон, находящийся под управлением операционной системы Андроид, в качестве беспроводной блютуз-мыши. Этот документ описывает структуру проекта и объясняет принципы его работы.

Проект, в частности, примечателен тем, что не требует установки стороннего программного обеспечения на компьютере, с которым будет взаимодействовать телефон, то есть на компьютер не надо устанавливать каких-либо специальных программ. Тем самым, телефон с установленной программой Андроид-Мышь будет взаимодействовать с любой операционной системой, что делает его универсальным решением для дистанционного позиционирования мыши. Отметим также удобство управления курсором: его позиционирование на экране компьютера осуществляется одной рукой.

- Скачать исходный код можно по этой ссылке

<https://github.com/Arseny-N/Android2Mouse/archive/master.zip>.

- Узнать о процессе установки можно по этой ссылке

<http://arseny-n.github.io/Android2Mouse/doc/howTo.html>.

- Увидеть, как телефон выполняет функции мыши можно по этой ссылке

<http://arseny-n.github.io/Android2Mouse/doc/index.html>.

- Ознакомиться с кодом можно по этой ссылке

<https://github.com/Arseny-N/Android2Mouse>.

---

## **Оглавление**

1. Цель проекта
2. Реализация проекта
  - 2.1 Принципы работы технологии Bluetooth
    - 2.1.1 HIDP
    - 2.1.2 SDP
    - 2.1.3 Код класса
    - 2.1.4 L2CAP
  - 2.2 Метод реализации проекта
    - 2.2.1 Hidp\_clientd
    - 2.2.2 Графический интерфейс
    - 2.2.3 Attribute Changer
    - 2.2.4 Сеанс работы
  - 2.3 Процесс разработки, установки и компиляции
    - 2.3.1 Используемые программы
    - 2.3.2 Процесс разработки
    - 2.3.3 Процесс компиляции
    - 2.3.4 Процесс установки
  - 2.4 Описание файлов проекта
3. Результат проекта
4. Литература

---

**1. Цель проекта:** создать программное обеспечение, позволяющее использовать смартфон в качестве манипулятора, заменяющего классическую компьютерную мышь. Смартфон с установленным программным обеспечением, должен был соответствовать следующим параметрам:

- Обеспечивать позиционирование курсора и клики левой и правой кнопками мыши.
- Управляться одной рукой.
- Не требовать поверхности для передвижения.
- Взаимодействовать с любой операционной системой, установленной на управляемом компьютере.
- Не требовать установки дополнительного программного обеспечения на управляемом компьютере.

## **2.1 Принципы работы технологии Bluetooth**

Чтобы понять, как Андроид-мышь выполняет свою работу, читателю следует вспомнить основные принципы технологии Bluetooth. Поскольку данный документ ставит перед собой цель описать структуру проекта Андроид-Мышь, а не принципы работы Bluetooth, многое будет упрощено или упущено. Но на протяжении всего документа автор будет отсылать читателя именно к этой части документа, т. к. описанные здесь технологии и термины, являются ключевыми для понимания его работы.

### **2.1.1 HIDP**

Любое периферийное устройство, которое использует в виде канала связи Bluetooth, должно отвечать спецификации HIDP (Human Interface Device Profile). Данная спецификация описывает протоколы, которые должно использовать устройство, и вводит некоторые критерии. Спецификация была создана с опорой на схожий стандарт USB, также направленный на периферийные устройства. В нашем случае мы будем рассматривать требования, предъявляемые к Bluetooth-мышь.

Чтобы устройство распознавалось как мышь и могло выполнять ее роль, оно должно:

- Иметь соответствующий SDP-рекорд.
- Иметь ID класса, соответствующий нужному устройству.
- Должно уметь установить L2CAP связь с хостом\*.

\* Далее мы будем использовать термин хост в качестве обозначения компьютера, с которым взаимодействует Андроид-мышь, такое название встречается в спецификациях Bluetooth.

---

Первый и последний пункты являются ключевыми, без их выполнения устройство не сможет выполнять поставленные перед ним задачи. Выполнение второго пункта зависит от хоста, но в общем случае им пренебрегать не следует. Далее мы подробно рассмотрим каждый пункт.

### **2.1.2 SDP**

SDP протокол даёт возможность приложениям выяснять, какие сервисы устройства доступны и их характеристики. Мы не будем подробно описывать SDP протокол, ограничимся только кратким изложением его сути.

Любое устройство, предоставляющее какие-либо сервисы Bluetooth, должно иметь SDP-сервер, который по запросу другого устройства будет отправлять списки сервисов и их характеристики запрашивающему клиенту. По сути, все основные функции, выполняемые Bluetooth устройством, описываются такими записями.

SDP запись устройства, соответствующего спецификации HIDP содержит, в частности, специальное поле, именуемое в спецификации как HID запись.

Схожесть в названии не случайна, как было сказано ранее, HIDP берёт свои корни из USB, там запись играет ту же роль, что и в Bluetooth, а именно описывает устройство и структуру пакета, посылаемого устройством при появлении информации. Упрощая, HID рекорд говорит следующее: «Это устройство — мышь, у неё три кнопки, в первых двух байтах будет информация о её нахождении, а в последнем — информация о состоянии её кнопок».

### **2.1.3 Код класса**

Любое Bluetooth устройство имеет некоторые атрибуты, которые считываются при его обнаружении. Одним из таких является код (тут речь идёт о более низком уровне, нежели SDP; следует отличать эти атрибуты от атрибутов в SDP рекорда). Его предназначение — дать возможность ОС быстро выяснять тип обнаруженного устройства.

### **2.1.4 L2CAP**

L2CAP протокол предоставляет услуги по работе с данными, в Bluetooth он является основой для более высокоуровневых протоколов типа HIDP.

Если привести аналогию с TCP, то: «IP адресом» устройства является MAC, а «портами» являются каналы. У каждого канала есть свой номер или PSM. Периферийное устройство, желающее установить связь с неким хостом и начать трансляцию пакетов, должно, согласно спецификации HIDP, открыть два канала с PSM равными 0x13 и 0x12.

---

## 2.2 Метод реализации проекта

В этой части документа читатель будет ознакомлен основными частями проекта и с причинами такого деления. Также в конце для большей ясности мы опишем сеанс работы с данным проектом.

Андроид-мышь можно разделить на три основные части:

- Hidp\_clientd или демон (расшифровку термина см. ниже).
- Графический интерфейс.
- Attribute Changer.

Далее будут использоваться следующие термины:

- *Исполняемый файл* или *бинарник* — файл, содержащий программу в том виде, в котором она может быть исполнена компьютером. В этой работе мы не будем относить этот термин к java-бинарникам.
- *Графическое приложение* — приложение, отображающее информацию на экране телефона в понятном для пользователя виде.
- *Демон* — программа, работающая без контролирующего терминала, т. е. программа, выполняемая в фоновом режиме.

### 2.2.1 Hidp\_clientd

Hidp\_clientd — это демон, устанавливающий L2CAP соединение и отсылающий поступающую к нему информацию (подробнее о L2CAP см. 1.2). Он хранится в виде бинарника в памяти телефона и запускается графическим интерфейсом. Установление соединения осуществляется при помощи открытия двух сокетов L2CAP с соответствующими PSM.

Открытие сокета — операция, требующая прав супер-пользователя; следовательно, телефон, на котором запущен hidp\_clientd, должен позволять запускать приложения с правами супер-пользователя, или, по иному, быть «рутнутым».

Hidp\_clientd — отдельный бинарник, поскольку стандартная программная среда Андроид не позволяет ни запускать приложения с правами супер-пользователя, ни создавать сокеты.

### 2.2.2 Графический интерфейс

Графический интерфейс — это приложение, которое считывает прикосновения пользователя и интерпретирует их, предварительно запустив демон. Это делается при помощи специальной java-библиотеки.

---

Далее данные записываются в специальный файл `fifo`, откуда их позже считывает демон. При завершении сеанса демону отсылается специальный код, и он тоже прекращает работу.

### 2.2.3 Attribute Changer

Этот раздел программы состоит из двух частей. Первая часть — бинарник, который при запуске либо сменяет класс устройства и добавляет соответствующую SDP запись, либо удаляет запись и возвращает нужный класс. Вторая часть — приложение, которое показывает пользователю нынешний класс устройства и предлагает его сменить (подробнее о SDP см. 1.1).

### 2.2.4 Сеанс работы

Сеанс работы с Андроид-мышью осуществляется следующим образом.

1. Пользователь запускает Attribute Changer и меняет их так, чтобы телефон стал мышью.
2. В стандартном меню Bluetooth создаёт пару с управляемым компьютером.
3. Запускает основное приложение и в списке спаренных устройств выбирает нужное.
4. Использует программу Андроид-мышью.
5. Завершает сеанс.

## 2.3 Описание процесса разработки, установки и компиляции

- *Процесс сборки или компиляции* — это процесс преобразования исходного кода в бинарный код, понятый компьютеру.
- *Процесс установки* — это процесс перенесения уже скомпилированных файлов в каталог, откуда они будут считаны и интерпретированы.

### 2.3.1 Используемые программы

Кратко опишем программы, использованные для разработки, установки и сборки проекта.

- `Arm-linux-gnueabi-gcc` — кросс-компилятор, пригодный для компиляции кода для arm процессоров (именно на arm работает большинство телефонов). Он использовался для компиляции `hidp_clientd` и `attributeChanger`.
- Пакет программ Android SDK от Google использовался для разработки

---

графических приложений на Андроид.

- EclipseIDE\* — графическая программа, позволяющая писать, компилировать и устанавливать код. Использовалась специальная версия Eclipse IDE со встроенными программами разработки под Андроид, далее она будет называться просто Eclipse IDE. Процесс сборки приложения под Андроид — это процесс компиляции некоторых файлов и перенесения их в файл с расширением .apk .

- Adb (Android Debug Bridge) — утилита, позволяющая переносить файлы на устройство с Андроидом и устанавливать и собирать файл с расширением .apk. Это утилита командной строки, что позволяет использовать её в скриптах.

### **2.3.2 Процесс разработки**

Графические приложения были написаны и скомпилированы в Eclipse IDE. Программы, хранящиеся в бинарном виде, были написаны в текстовом редакторе.

### **2.3.3 Процесс компиляции**

Adb компилирует графические приложения, а arm-linux-gnueabi-gcc компилирует бинарники.

### **2.3.4 Процесс установки**

Adb устанавливает графические приложения и переносит программы, хранящиеся в бинарном виде.

## **2.4 Описание файлов проекта**

В этой разделе документа описана «файловая структура» проекта. Для большей ясности ниже после названия папки будет стоять наклонная черта.

- build/ — в этой директории хранятся скомпилированные файлы, готовые к установке на устройство.

- src/ — в этой директории хранится исходный код, из которого создаются файлы в директории build/.

- BecomeAMouse/, BluetoothMouse/ — каталоги с кодом графических приложений.

- bin/ — каталог с кодом бинарных приложений.

---

- tools/ — в этой директории хранятся файлы, нужные для сборки файлов в каталоге build/.

- tmp/ -- эта директория нужна для временных файлов, создаваемых во время процесса установки.

- doc/ -- в этой директории хранится документация к проекту.

- install.bat, install.sh, Makefile — файлы, необходимые для компиляции и установки проекта.

### **3. Результат**

Результатом данного проекта является набор программ, отвечающий всем поставленным требованиям.

С демонстрационным видеороликом о работе Андроид-Мыши можно ознакомиться по ссылке <http://arseny-n.github.io/Android2Mouse/doc/index.html>.

### **4. Литература**

При разработке данного проекта использовались следующие интернет-ресурсы:

- Спецификации Bluetooth

<https://www.bluetooth.org/>

<http://www.palowireless.com/infotooth>

- Спецификации HID

<http://www.usb.org/developers/hidpage/>

- Документация с официального портала Андроид

<http://developer.android.com/>

- Энциклопедия «Википедия»

<https://en.wikipedia.org>