

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
1 Выбор языка и среды разработки.....	5
2 Технология Bluetooth.....	6
3 Разработка интерфейса пользователя.....	9
3.1 Построение диаграммы состояний интерфейса.....	9
3.2 Разработка форм интерфейса.....	11
4 Разработка диаграммы последовательности действий.....	14
5 Выбор архитектуры.....	16
6 Пример работы системы.....	18
ЗАКЛЮЧЕНИЕ.....	20
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	21

ВВЕДЕНИЕ

В настоящее время устройства, использующие технологию Bluetooth для связи с другими устройствами, используются повсеместно. Люди слушают музыку через беспроводные наушники. Можно избавиться от лишних проводов на столе, подключив клавиатуру по Bluetooth к компьютеру. Эта универсальная технология позволила обеспечить надежную беспроводную передачу данных, что и обусловило ее популярность.

Подключение Bluetooth модуля к микропроцессорной системе может обеспечить возможность управления ей с другого устройства. Наличие Bluetooth в любом современном смартфоне определяет актуальность управления подобными системами с помощью мобильного приложения.

Целью данной практики является разработка программной системы, позволяющей управлять микропроцессорной системой контроля попадания волана в заданную зону бадминтонной площадки, а также подключение модуля Bluetooth к существующей микропроцессорной системе. Программная система должна обеспечивать следующий функционал:

1. Подключение к микропроцессорной системе.
2. Задание точек — линий игрового поля и аута.

3. Инициализация начала работы микропроцессорной системы, после чего нажатия на чувствительный к давлению мат будут вызывать реакцию системы — включение красного или зеленого светодиода.

ОСНОВНАЯ ЧАСТЬ

1 Выбор языка и среды разработки

Программная система должна работать на ОС Android. Главной особенностью Android является то, что в один момент времени существует множество устройств, на которых установлены разные версии операционной системы. Так, еще до выхода Android 11 предыдущая версия Android 10 использовалась примерно 8,2% устройств. Исходя из данных, представленных на рисунке 1, а также используемых элементов пользовательского интерфейса, было принято решение провести разработку под Android версии 5.0

ANDROID PLATFORM VERSION	API LEVEL	CUMULATIVE DISTRIBUTION
4.1 Jelly Bean	16	
4.2 Jelly Bean	17	99,8%
4.3 Jelly Bean	18	99,5%
4.4 KitKat	19	99,4%
5.0 Lollipop	21	98,0%
5.1 Lollipop	22	97,3%
6.0 Marshmallow	23	94,1%
7.0 Nougat	24	89,0%
7.1 Nougat	25	85,6%
8.0 Oreo	26	82,7%
8.1 Oreo	27	78,7%
9.0 Pie	28	69,0%
10. Q	29	50,8%
11. R	30	24,3%

Рисунок 1 — Распределение версий ОС по охвату устройств

Одной из лучших IDE для работы с платформой Android является Android Studio. Она предлагается в виде пакета вместе с Android SDK. Преимущество IDE заключается в наличии визуального конструктора для верстки страниц и встроенных эмуляторов различных устройств с ОС Android.

Android Studio поддерживает два основных языка разработки — Java и Kotlin. Также он способен поддерживать и другие языки программирования (к примеру, C/C++, C#). В данной работе было принято решение использовать язык программирования Kotlin, как более лаконичный и безопасный. Основное отличие Kotlin от Java заключается в том, что Kotlin требует меньше «шаблонного» кода, т.е. на выходе получается более простая для чтения система. Он также устраняет такие ошибки, как исключение нулевого указателя.

2 Технология Bluetooth

Bluetooth — производственная спецификация беспроводных персональных сетей (Wireless personal area network, WPAN). В настоящее время Bluetooth существует в двух спецификациях — Bluetooth Classic и Bluetooth Low Energy.

Bluetooth Classic также называют Bluetooth Basic Rate/Enhanced Data Rate (BR/EDR). Большей частью используется для беспроводной передачи аудио данных — в беспроводных громкоговорителях, наушниках, встроенных автомобильных развлекательных системах.

Bluetooth Low Energy (BLE) — технология с низким энергопотреблением. Благодаря этой особенности, активно применяется в IoT. Появилась в версии стандарта Bluetooth 4.0.

Эти два типа Bluetooth несовместимы друг с другом. Устройства, которые используют Bluetooth Classic, не могут напрямую связываться с устройствами, использующими BLE, и наоборот. Однако есть устройства, которые поддерживают оба типа соединения. Тогда связь между устройствами может происходить так, как на рисунке 2.

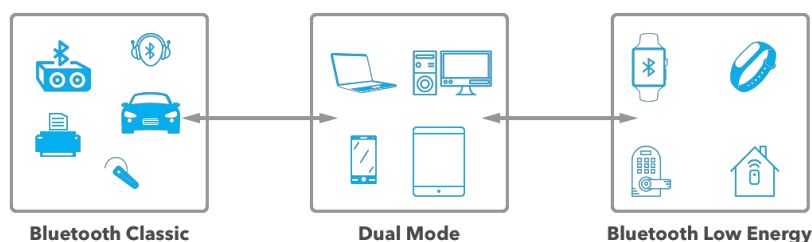


Рисунок 2 — Типы Bluetooth-устройств

Bluetooth Classic обычно используется для приложений, в которых требуется передавать большое количество информации. Несмотря на то, что классический Bluetooth не оптимизирован для низкого энергопотребления, он поддерживает большую скорость передачи (максимум 3 МБит/с против 2 МБит/с BLE). Bluetooth Classic поддерживает один тип топологии — Точка-Точка (P2P). Описанные выше характеристики являются преимуществами для данной работы, для использования была выбрана именно эта технология.

Для подключения к микропроцессорной системе был выбран модуль Bluetooth HC-05. Модуль поддерживает стек Bluetooth 2.0 + EDR. Подключение модуля осуществляется с помощью разъема ХН-2-6Р, выводы которого имеют следующее назначение:

STATE — к цифровому выводу контроллера, который предназначен для передачи информации о режимеработы модуля;

RX — к выводу TX последовательного порта контроллера для передачи данных;

TX — к выводу RX последовательного порта контроллера для передачи данных;

VCC — к “+” источника питания контроллера;

GND — «земля» (общий провод).

Модуль представляет из себя печатную плату, на которой смонтированы компоненты, отвечающие за установление связи с контроллером по последовательному интерфейсу UART и организацию передачи данных по Bluetooth каналу при сопряжении с каким либо другим электронным устройством.

Особенностью модуля является возможность его работы в нескольких режимах:

1. Режим поиска устройств и сопряжение с ними, после чего свободно выполняется передача данных между устройствами;

2. Режим передачи данных между модулем и сопряженным устройством. В этом случае возможны два варианта взаимодействия модуля с подключаемыми устройствами:

2.1. Slave — является подчиненным по отношению к другим устройствам. В этом режиме модуль способен только принимать и передавать данные управляющему устройству, с которым он сопряжен. Например, данный режим используется при подключении модуля к смартфону. Режим Slave установлен в настройках модуля по умолчанию.

2.2. Master — модуль является управляющим по отношению к другим устройствам. В этом режиме Bluetooth модуль способен осуществлять поиск Bluetooth-устройств, выполнять сопряжение с ними и, в дальнейшем, управлять их работой, передавая и принимая данные от подчиненного (Slave) устройства.

Данный режим может использоваться для обеспечения связи между двумя контроллерами. В режим Master можно перевести модуль через режим программирования.

4. Режим программирования. Данный режим позволяет изменять всевозможные настройки модуля: запрашивать и менять его имя, адрес, скорость передачи данных; менять режимы со Slave на Master и обратно, запрашивать статусы и классы, выполнять изменение кода доступа к модулю, а также менять другие параметры работы.

В данной работе модуль работает в режиме Slave.

Для определения того, в каком режиме работает модуль, существует система индикации с помощью индикаторного светодиода, находящегося на поверхности платы:

1. Частое мигание светодиода — модуль не подключен к устройствам, поиск устройств для подключения;

2. Светодиод мигает с частотой 1 Гц — модуль в режиме программирования;

3. Светодиод кратковременно мигает один раз в две секунды — модуль сопряжен с другим Bluetooth-устройством.

Модуль обладает техническими характеристиками, представленными в таблице 1.

Таблица 1 — Характеристики модуля НС-05

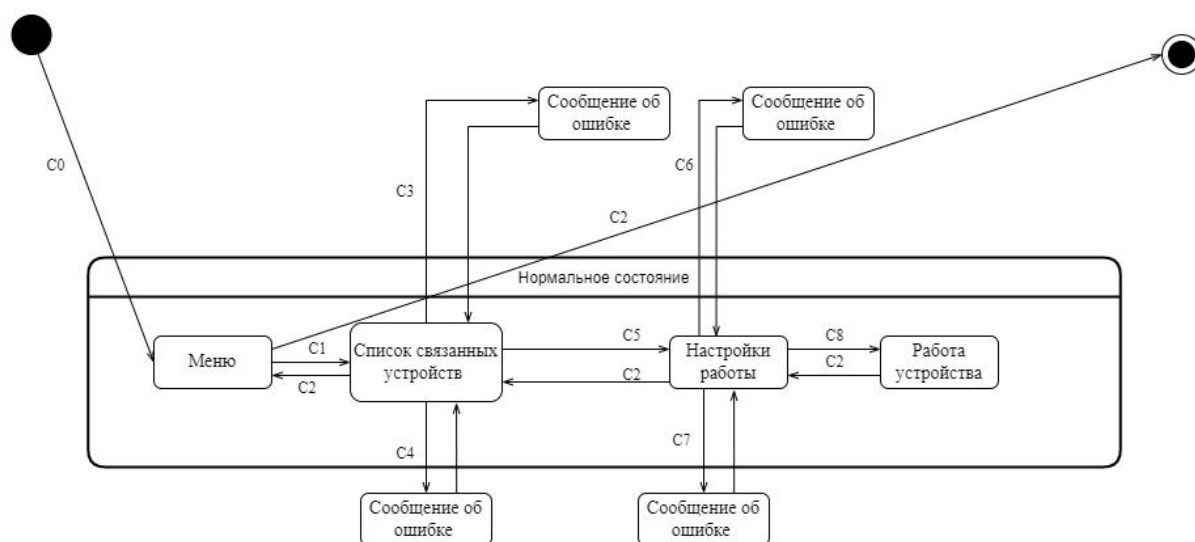
Характеристика	Значение
Диапазон частот радиосвязи	2,4–2,48 ГГц
Мощность передачи	0,25–2,5 мВт
Чувствительность	–80 dBm
Потребляемый ток	50 мА
Радиус действия	до 10 метров
Температура хранения	–40...85 °С
Рабочий диапазон температур	–25...75 °С

3 Разработка интерфейса пользователя

В разработанном приложении применен объектно-ориентированный подход к разработке интерфейсов — пользователь взаимодействует с объектами напрямую, инициируя выполнение операций.

3.1 Построение диаграммы состояний интерфейса

В приложении было разработано четыре формы для взаимодействия с пользователем. Диаграмма состояний интерфейса представлена на рисунке 3.



C0 — Запуск приложения/
Инициализация исходного
состояния
C1 — Нажатие кнопки "Start training"/
Инициализация экрана со списком
связанных устройств
C2 — Нажатие кнопки "Назад"
C3 — Отсутствие связанных устройств/
Загрузка пустого списка устройств
C4 — Выбор связанного устройства
[Ошибка подключения]/
Обработка ошибки

C5 — Выбор связанного устройства
[Корректное подключение],
нажатие кнопки/
Подключение к устройству,
инициализация экрана настроек
МП-системы
C6 — Ввод координат линий поля
[Данные некорректны]/
Обработка данных
C7 — Нажатие кнопки "Next"
[Ошибка со стороны мата]/
Обработка ошибки

C8 — Нажатие кнопки "Next"
[Данные корректны]/
Отправление данных, старт работы мата

Рисунок 3 — Диаграмма состояний работы интерфейса

При запуске приложения пользователь попадает на стартовый экран — меню. При нажатии на кнопку «Start training» происходит переход на экран со списком связанных устройств. Если Bluetooth на смартфоне выключен или связанных устройств нет, приложение инициализирует запуск всплывающего сообщения, содержащего информацию об ошибке.

При выборе устройства с названием модуля Bluetooth происходит установка соединения. В случае, если оно прошло успешно, происходит переход на экран задания настроек. Иначе — инициализируется очередное всплывающее окно с ошибкой.

В окне задания настроек пользователь имеет поле, в которое может ввести числа. Если значение в этом поле окажется вне установленного интервала, система выведет сообщение с соответствующей ошибкой.

Если в момент нажатия кнопки «Next» в полях будет задано корректное значение, данные будут отправлены микропроцессорной системе. Если в процессе инициализации внутренних структур и подготовке к работе в системе

возникнет ошибка, соответствующее сообщение будет отправлено устройству. В таком случае, будет инициализировано всплывающее окно с описанием ошибки. Иначе происходит переход в новое окно с сообщением, что микропроцессорная система приступает к работе.

3.2 Разработка форм интерфейса

Исходя из вышеперечисленных состояний, были разработаны формы интерфейса, представленные на рисунках 4-7.

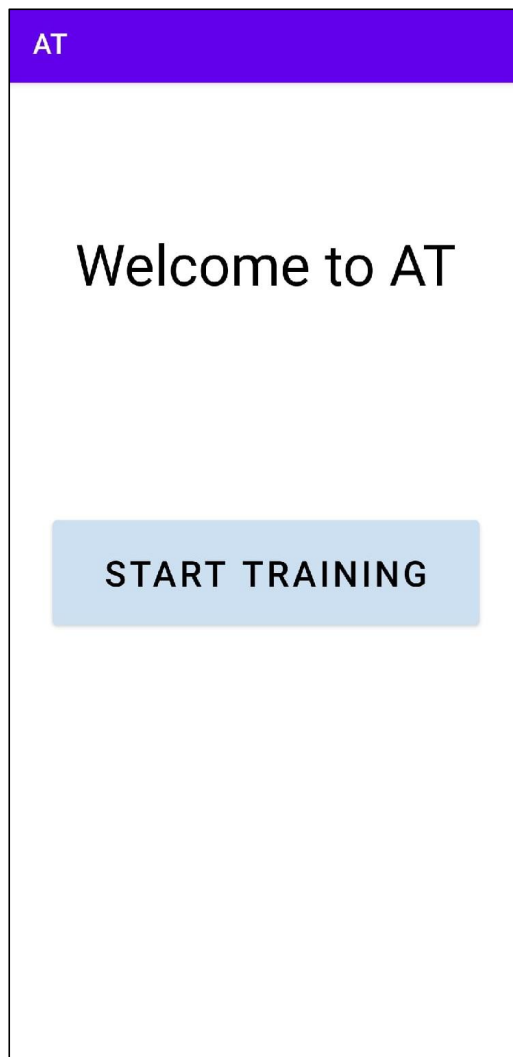


Рисунок 4 — Форма стартового экрана

АТ

AirPods

7B:58:95:33:88:67

HC-05

00:21:06:08:2A:CF

i12

41:42:01:42:95:EB

Наушники AirPods (Ольга)

F0:5C:D5:DE:30:6E

X5

00:58:00:00:18:54

Yandex.Station-2a0290

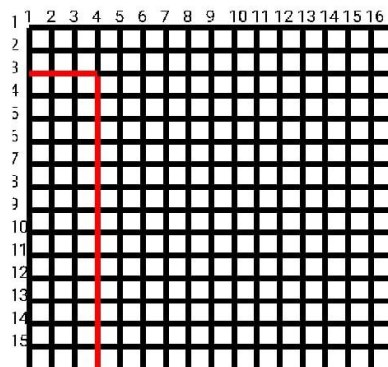
C0:84:7D:78:F8:43

NEXT

Рисунок 5 — Форма списка связанных устройств

AT

Select the lines
(left inner zone will be 'in' field)



3

4

Select the number of errors in row

4

NEXT

Рисунок 6 — Форма задания настроек

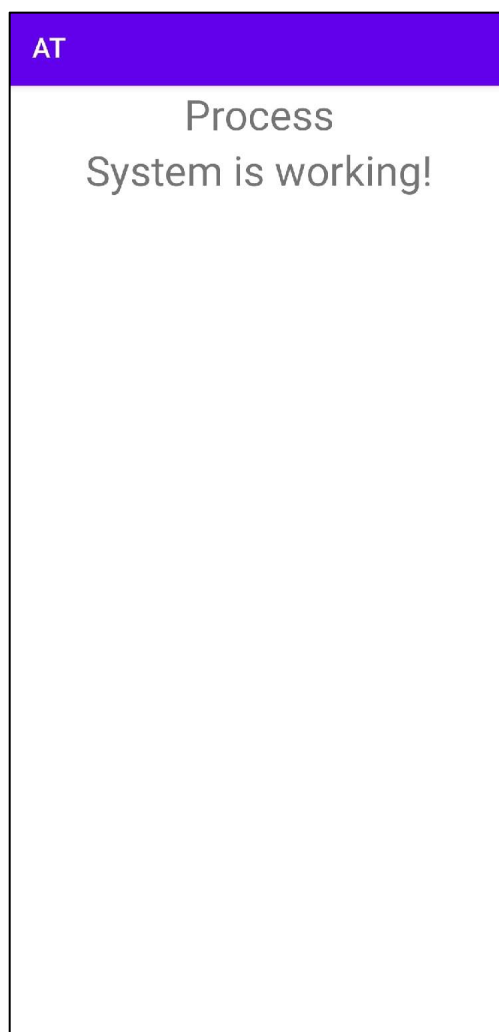


Рисунок 7 — Форма готовности МП-системы к работе

4 Разработка диаграммы последовательности действий

Взаимодействие компонентов системы рассмотрим на примере диаграммы последовательности действий при инициализации начала тренировки на рисунке 8.

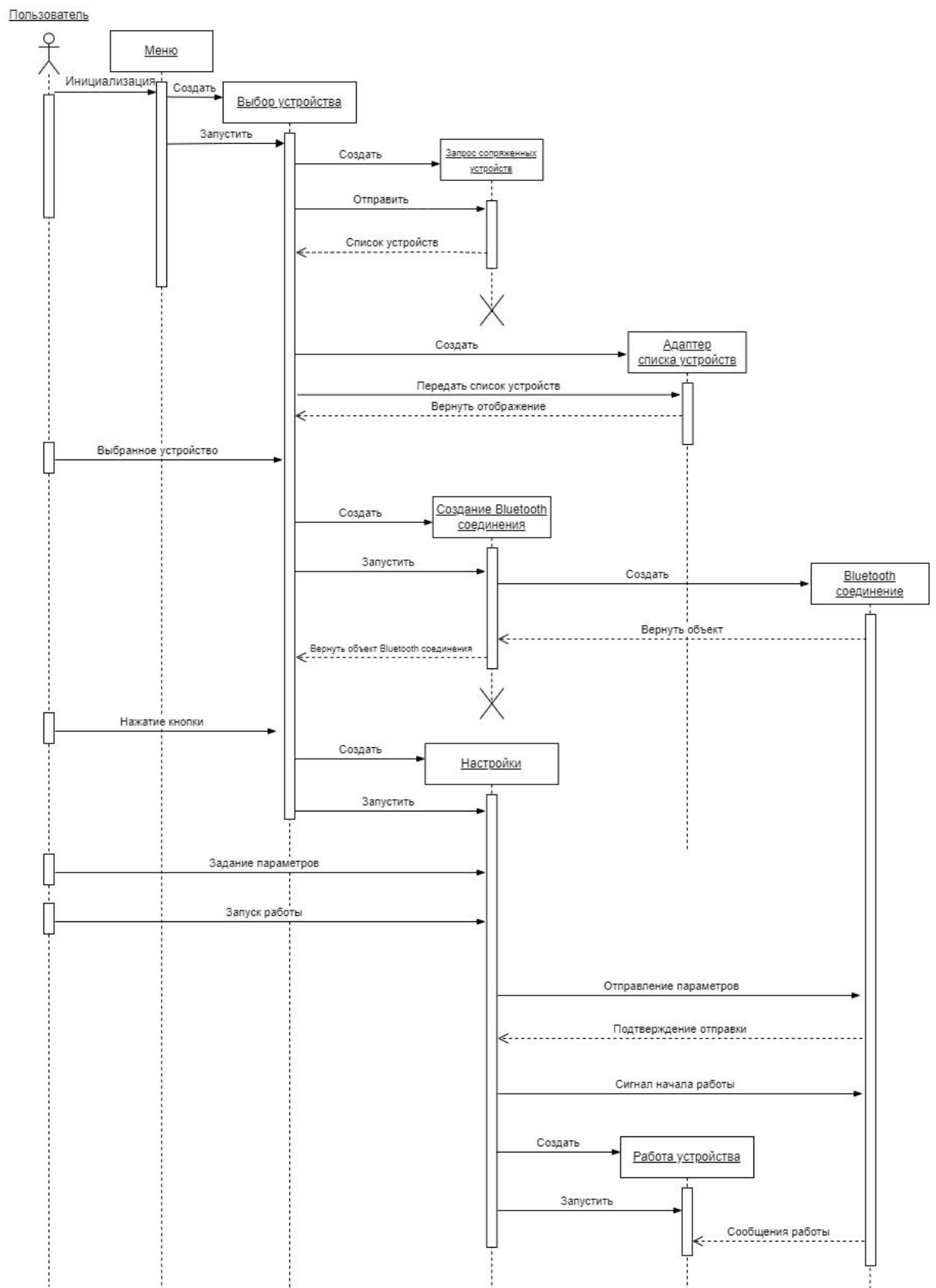


Рисунок 8 — Диаграмма последовательности действий при инициализации начала тренировки

Отображение списков производится с помощью адаптеров, которые создаются при инициализации экрана. Создание Bluetooth соединения

происходит следующим образом. Для начала создается временный объект, который, в случае успешной установки соединения, возвращает Bluetooth сокет, Он в дальнейшем используется в итоговом классе Bluetooth потока.

5 Выбор архитектуры

Одними из основных компонентов приложения в Android являются Activity и Fragment.

Activity отвечает за визуальную часть приложения, за взаимодействие с пользователем. Обычно один Activity отвечает за один экран.

Fragment — модульные, переиспользуемые части пользовательского интерфейса. Они не самостоятельны, а зависят от Activity. Жизненные циклы Activity и Fragment представлены на рисунке 9.

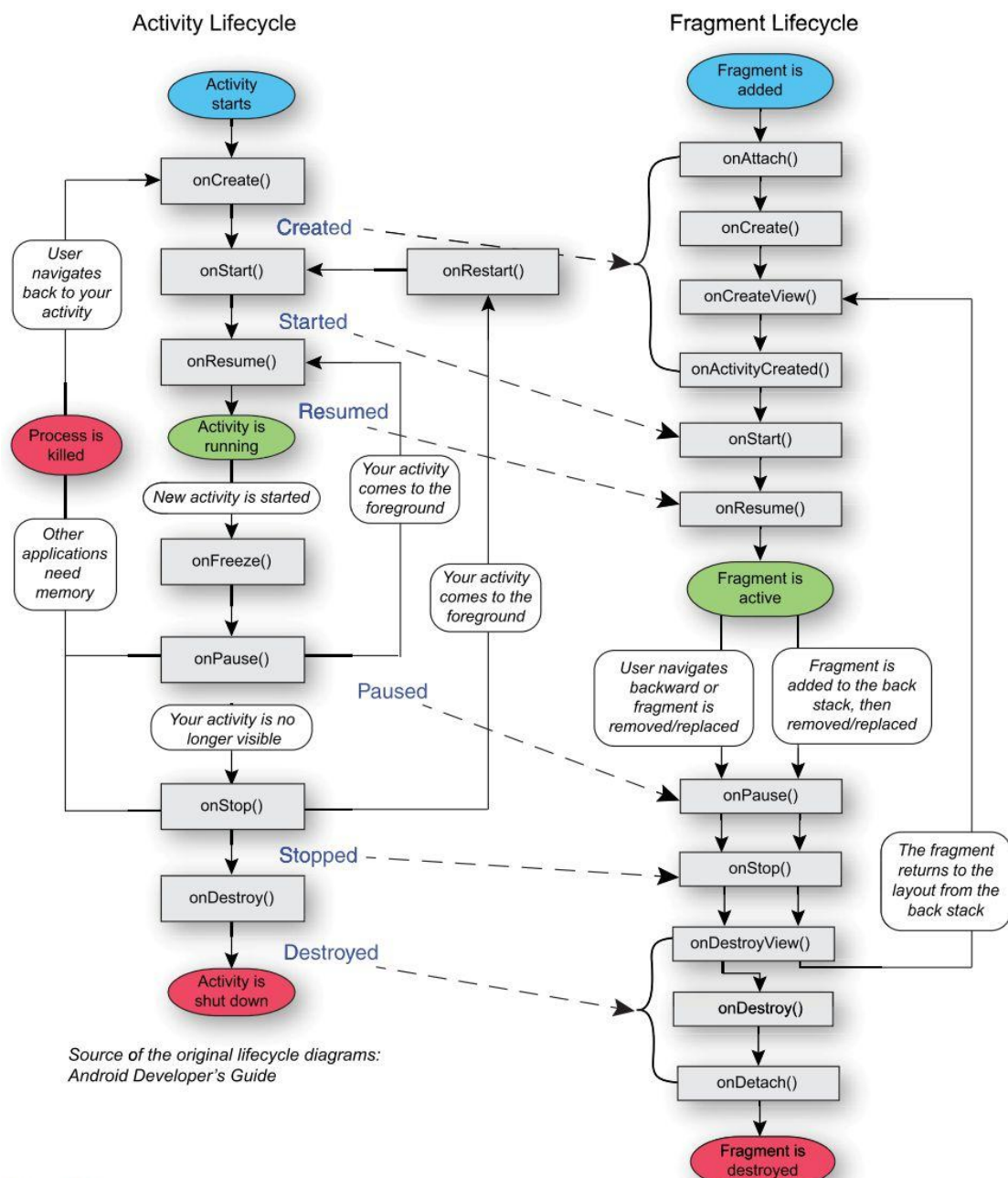


Рисунок 9 — Жизненные циклы Activity и Fragment

У фрагмента жизненный цикл немного длиннее, но его методы все равно соотносятся с методами Activity.

В данном приложении было принято решение использовать архитектуру Single Activity. В ней присутствует только одна Activity и несколько Fragment. Основным преимуществом в данной архитектуре является то, что Fragment легче, чем Activity — присутствует выигрыш в производительности. Также в данном приложении необходимо реализовать объект сокета, соединяющего смартфон и плату Arduino по Bluetooth. Для того, чтобы иметь возможность работать с установленным соединением из любой формы и каждый раз не

пересоздавать его, сокет можно инициализировать в MainActivity. Тогда соединение будет существовать во всех экранах вплоть до выхода из приложения.

6 Пример работы системы

Код программной системы представлен в репозитории git по ссылке — <https://github.com/moroz-matros/AT>.

Отладка производилась на физическом устройстве. Выполнялся стандартный пользовательский сценарий:

1. Производился выбор нужного устройства.
2. Задавались параметры работы микропроцессорной системы.
3. Инициализировался старт работы микропроцессорной системы.

Примеры работы макета микропроцессорной системы под управлением приложения Android представлены на рисунках 10 и 11.

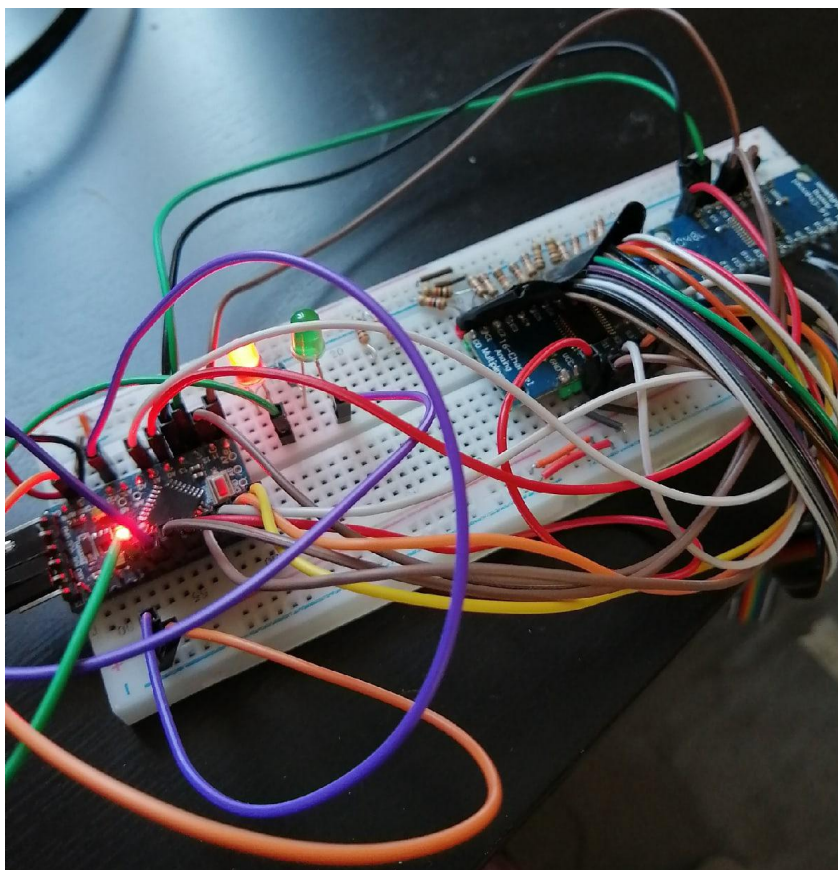


Рисунок 10 — Включение красного светодиода при попадании в аут


```
out 12 14 891  
E/Arduino Message: out 13 14 727  
E/Arduino Message: out 14 13 645  
E/Arduino Message: out 12 14 743  
E/Arduino Message: out 13 14 817  
E/Arduino Message: out 14 14 809  
out 12 13 662
```

Рисунок 11 — Вывод в отладочную консоль значений при попадании в
аут

ЗАКЛЮЧЕНИЕ

В ходе прохождения практики были достигнуты поставленные цели и выполнены следующие задачи:

1. Произведено ознакомление с технологией Bluetooth;
2. Было разработано и реализовано мобильное приложение под ОС Android, отвечающее требованиям задания.
3. Продемонстрирована работоспособность микропроцессорной системы под управлением мобильного приложения.

Отчет был оформлен в соответствии с ГОСТ 7.32-2017.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ 7.32-2017. Отчет о научно-исследовательской работе. Структура и правила оформления.
2. Иванова Г.С. Технология программирования: Учебник для вузов. — М.: Изд-во МГТУ им. Н.Э. Баумана, 2002. — 320с.: ил. (Сер. Информатика в техническом университете.)
3. Android Developers [Электронный ресурс]. — URL: <https://developer.android.com/> (дата обращения 2022-02-08)
4. Как выбрать язык программирования для создания Андроид — приложения [Электронный ресурс]. — URL: <https://habr.com/ru/post/477578/> (дата обращения 2022-02-08)
5. Официальный сайт языка программирования Kotlin [Электронный ресурс] — URL: <https://kotlinlang.org/> (дата обращения 2022-02-10)