

# App inventor:

**Bluetooth: передача данных**

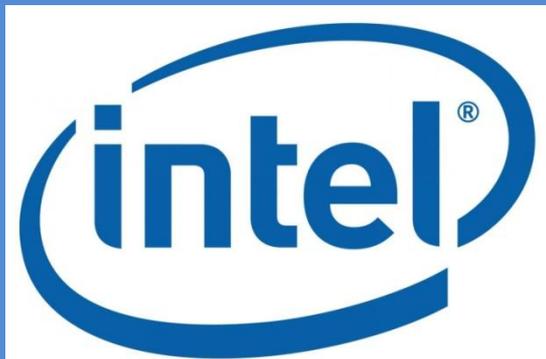


**Цель:**

Узнать: что такое «**Bluetooth**».

Научится передавать информацию с помощью «**Bluetooth**»

# Немного Истории



В 1998 году инженер компании Intel **Джим Кардак (Jim Kardach)**, в составе большой команды разработчиков трудился над радиотехнологией ближнего действия.



Системой связи, которая могла бы **объединить** компьютеры и мобильные телефоны **универсальным протоколом связи**. Для этой технологии было перебрано множество названий, но ни одно из них не устраивало руководство компании. Одни были какими-то громоздкими, другие не отражали сути проекта.

# Немного Истории

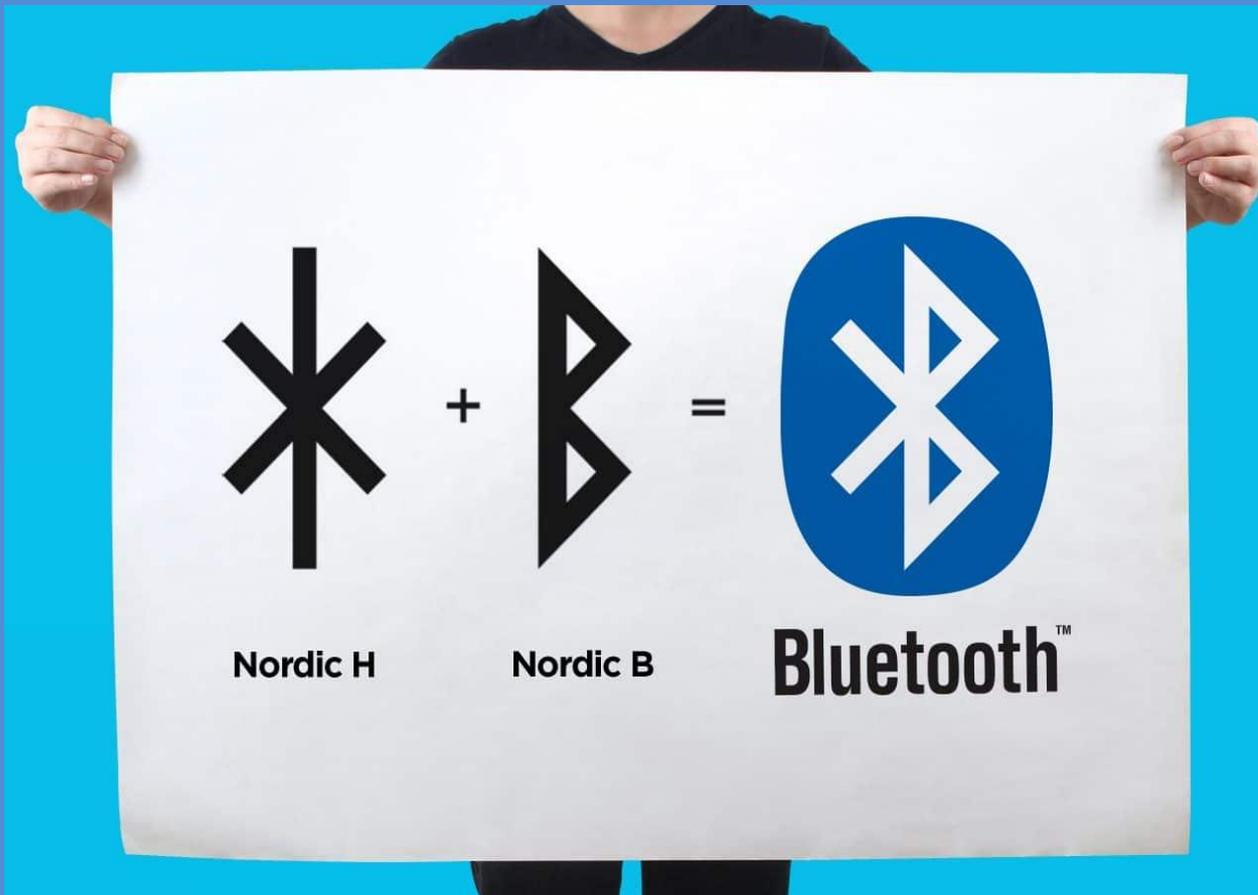
Тогда, **Джим Кардак** услышал от своего шведского коллеги историю об одном маленьком датском городке. Он известен лишь одной старинной достопримечательностью. Там установлены **рунные камни Еллингга**.

# Рунные камни Еллингга

**Самый маленький и самый старый** из них поставлен королем Дании Гормом в честь жены, Тиры. **Камень побольше** — дело рук их сына, чье имя, в переводе на английский звучит как **Харольд Блютуз (Синезубый)**. Надпись на нем гласит «Харальд, покоривший всю Данию и Норвегию, кто крестил датчан».



# ЛОГОТИП



Что до логотипа,  
то он означает:  
**Харальд Блютуз.**

«ХБ», написанные  
рунным алфавитом  
слившись вместе  
образуют нечто  
древней подписи.

Тысячу лет назад датский король приказал вырезать и воздвигнуть камни в память о своих деяниях, а через век его имя увековечили в каждом смартфоне.

# Принцип функционирования протокола

Что же до **процесса передачи**, то выбрав файл в меню операционной системы, вы активируете радиопередатчик, работающий в ограниченном диапазоне частот в районе **2,4ГГц**, так называемом **ISM** — **Industry, Science and Medicine** — части спектра, выделенной специально под нужды различных высокотехнологичных устройств.

Bluetooth начинает отслеживать все сигналы, в этих пределах. Смартфон на другом конце делает то же самое — старается вычлнить из шума помех знакомые приветственные последовательности.

Стоит иметь в виду, что формально **Bluetooth** это разом и **приемник**, и **передатчик**, но одновременно работать в обоих режимах чип не может.

Исключение составляет звук в bluetooth-гарнитурах, но этот аспект технологии заслуживает отдельного рассказа.

# Принцип функционирования протокола

После того, как телефоны обнаружили друг друга, начинается настоящая магия.

**Первый Bluetooth контроллер** берет на себя роль **передатчика**, а **второй** становится **приемником**.

**Передатчик** формирует из радиоволн **особый шаблон**, который меняется 1600 раз за секунду «прыгая» с одной из доступных 79 частот (в пределах с 2,401ГГц до 2,480ГГц) на другую в почти что случайном порядке.

**Шаблон** выступает в роли **идентификатора**.

Обмениваться файлами смогут только те устройства, которые настроены на один и тот же шаблон.

В то же время для других приборов передаваемая информация ничем не отличается от обычных радиопомех.

Так удастся добиться не только приватности, но и беспрепятственной одновременной работы нескольких пар приемник-передатчик, например, в одной комнате.

# Принцип функционирования протокола

Непосредственно перед отправкой по bluetooth  
данные разбиваются на **пакеты**

*— определённым образом оформленные блоки данных,  
сопровождаемые инструкциями по сборке из них исходного файла.*

Процессор приемника обрабатывает эти пакеты,  
складывает их в один файл  
и помещает его в постоянную память устройства.

Если обнаруживается нехватка пакета, он запрашивается повторно.

Весь этот процесс требует куда меньше времени,  
чем понадобилось на то, чтобы его объяснить  
— обычно от нескольких секунд, до пары минут.

Время передачи файла зависит главным образом от его размера.

# BluetoothClient

Используйте **BluetoothClient** для соединения своего устройства с другими устройствами, посредством **Bluetooth**.

Этот компонент использует **the Serial Port Profile (SPP)** для коммуникации.

Если заинтересованы в использовании **Bluetooth low energy**, то нам нужно будет обратить внимание на расширение (extension) **BluetoothLE**.

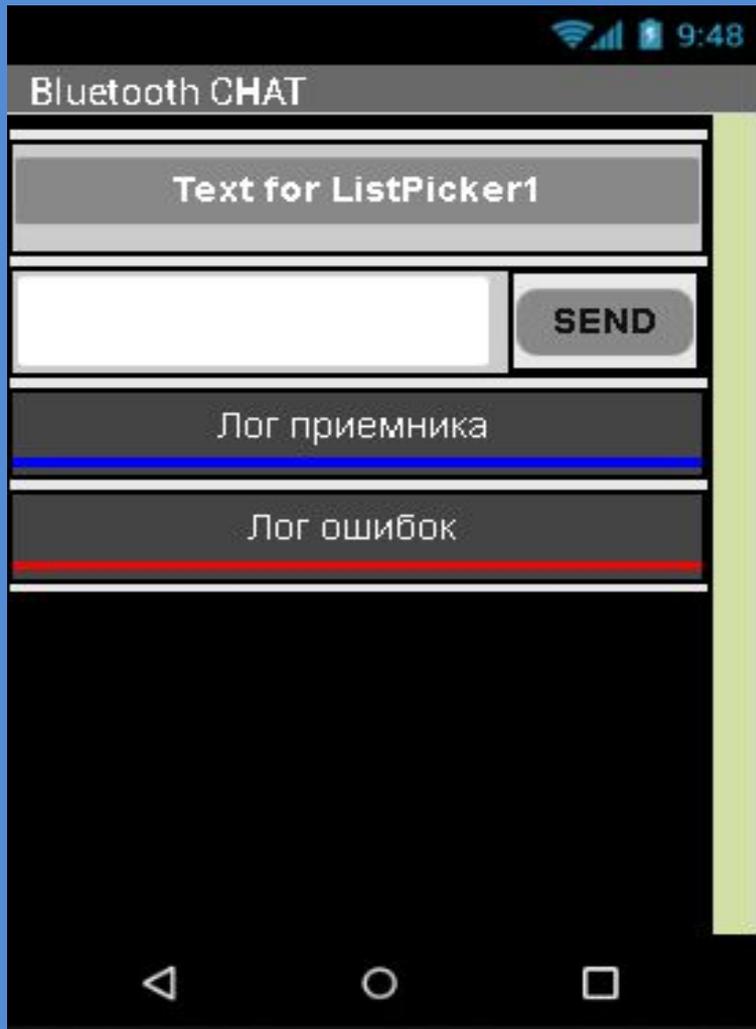
Дальше рассмотрим **Свойства, События, Методы BluetoothClient**.

# BluetoothServer

Используйте компонент **BluetoothServer**, чтобы переключить устройство в режим «сервера», который принимает соединения(connections) от других приложений, использующих компонент **BluetoothClient**.

Дальше рассмотрим **Свойства, События, Методы**  
**BluetoothServer**.

# Дизайн



<<< **ListPicker\_SELECT\_bluetooth**

(Для выбора устройства, к которому будем подключаться)

<<< **TextBox\_SEND\_Bluetooth** и **Button\_SEND**

(Поле ввода сообщения и кнопка «отправить»)

<<< **Label\_Bluetooth\_RECEIVER**

(Надпись. Хранит все полученные сообщения)

<<< **Label\_ERROR\_MESSAGE**

(Надпись. Хранит описание ошибок)

## Невидимые компоненты

    
BluetoothClient1 BluetoothServer1 ActivityStarter1

   
Clock\_Bluetooth\_RECEIVER TinyDB1

   
Clock\_Bluetooth\_ENABLE\_SERVER Sound1

# Принцип работы

инициализировать глобальную Bluetooth\_RECEIVER в " " "

когда Screen1 ▾ .Инициализировать

делать

- если `call MAC_address_and_name ▾ = ▾ " 00000000000000000000000000000000 " ▾`
- то
  - присвоить ListPicker\_SELECT\_bluetooth ▾ Текст ▾ в " SELECT DEVICE Bluetooth " ▾
  - присвоить ListPicker\_SELECT\_bluetooth ▾ ЦветФона ▾ в
- иначе
  - присвоить ListPicker\_SELECT\_bluetooth ▾ Текст ▾ в call MAC\_address\_and\_name ▾
  - присвоить ListPicker\_SELECT\_bluetooth ▾ ЦветФона ▾ в
- присвоить Clock\_Bluetooth\_RECEIVER ▾ ИнтервалТаймера ▾ в 100 ▾
- присвоить ActivityStarter1 ▾ Действие ▾ в " android.bluetooth.adapter.action.REQUEST\_ENABLE " ▾
- вызов ActivityStarter1 ▾ .НачатьДействие

в MAC\_address\_and\_name

результат

- вызов TinyDB1 ▾ .ПолучитьЗначение
  - тег " MAC address and name " ▾
  - оценитьЕслиТэгОтсутствует " 00000000000000000000000000000000 " ▾

# Принцип работы

когда ListPicker\_SELECT\_bluetooth ▾ .ПередВыбором

делать присвоить ListPicker\_SELECT\_bluetooth ▾ Элементы ▾ в BluetoothClient1 ▾ АдресаИИмена ▾

когда ListPicker\_SELECT\_bluetooth ▾ .ПослеВыбора

делать  если вызов BluetoothClient1 ▾ .Подключить адрес ListPicker\_SELECT\_bluetooth ▾ Выбор ▾

то присвоить ListPicker\_SELECT\_bluetooth ▾ ЦветФона ▾ в 

присвоить ListPicker\_SELECT\_bluetooth ▾ Текст ▾ в ListPicker\_SELECT\_bluetooth ▾ Выбор ▾

присвоить Screen1 ▾ Заголовок ▾ в  соединить " SAVED: " ListPicker\_SELECT\_bluetooth ▾ Выбор ▾

вызов TinyDB1 ▾ .СохранитьЗначение  
тег " MAC address and name "

сохранитьЗначение ListPicker\_SELECT\_bluetooth ▾ Выбор ▾

иначе присвоить Screen1 ▾ Заголовок ▾ в " Failed to connect to Bluetooth DEVICE "

когда ActivityStarter1 ▾ .После действия

результат

делать присвоить Screen1 ▾ Заголовок ▾ в " Bluetooth CHAT "

# Принцип работы

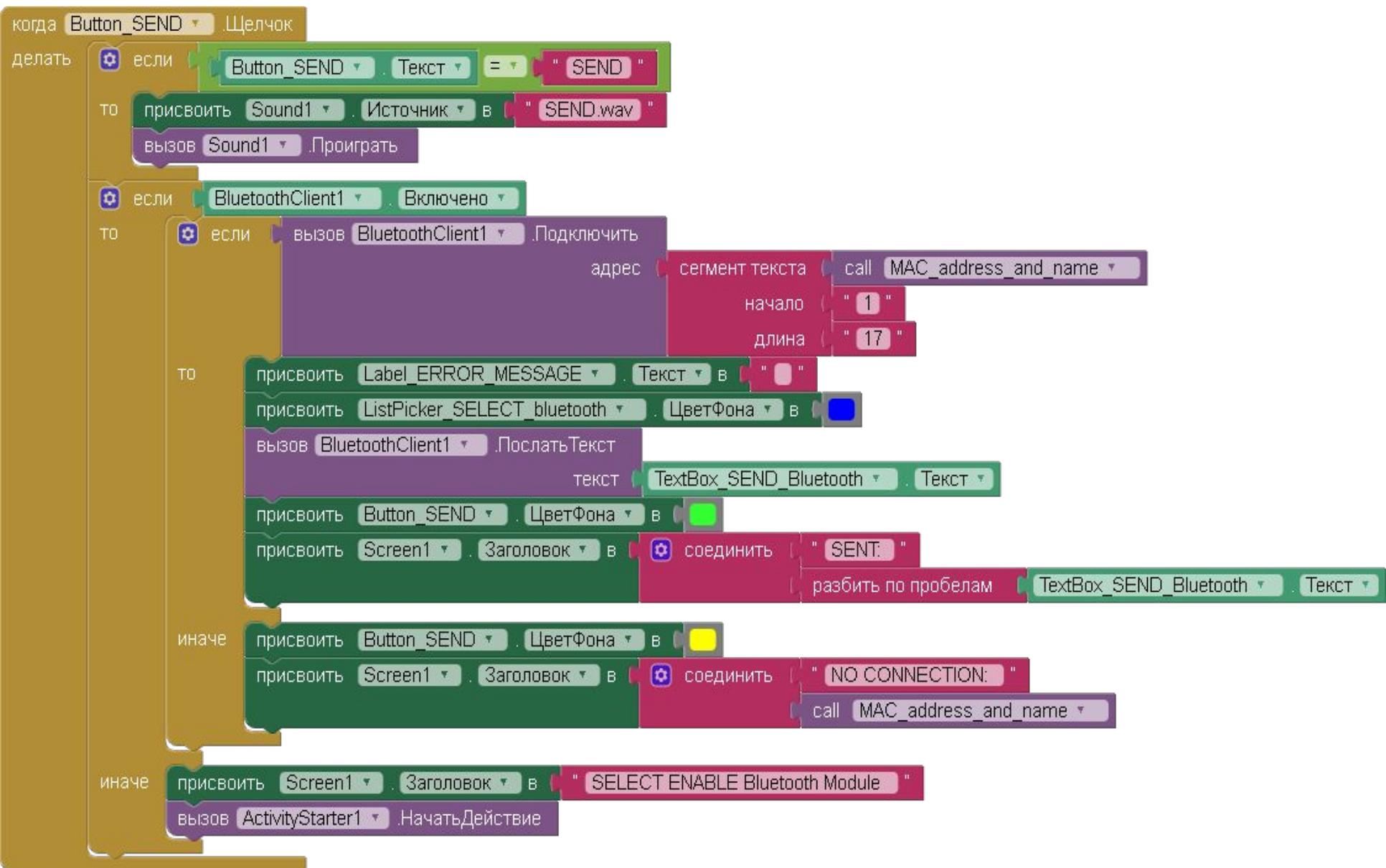
когда Screen1 ▾ .ПроизошлаОшибка

компонент    имяФункции    номерОшибки    сообщение

делать

- если Screen1 ▾ . Заголовок ▾ ≠ ▾ " v.1 Bluetooth CHAT ↗ REMOTE CONTROL "
- то
  - присвоить Label\_ERROR\_MESSAGE ▾ . Текст ▾ в
  - соединить
    - получить номерОшибки ▾
    - " \n "
    - получить сообщение ▾
    - " \n "
    - получить имяФункции ▾
    - " \n "
    - получить компонент ▾

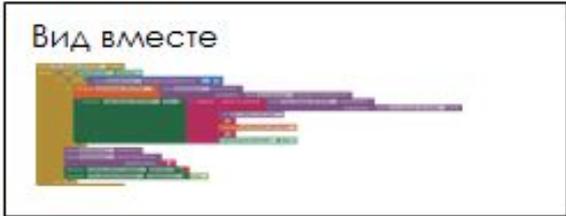
# Принцип работы



# Принцип работы

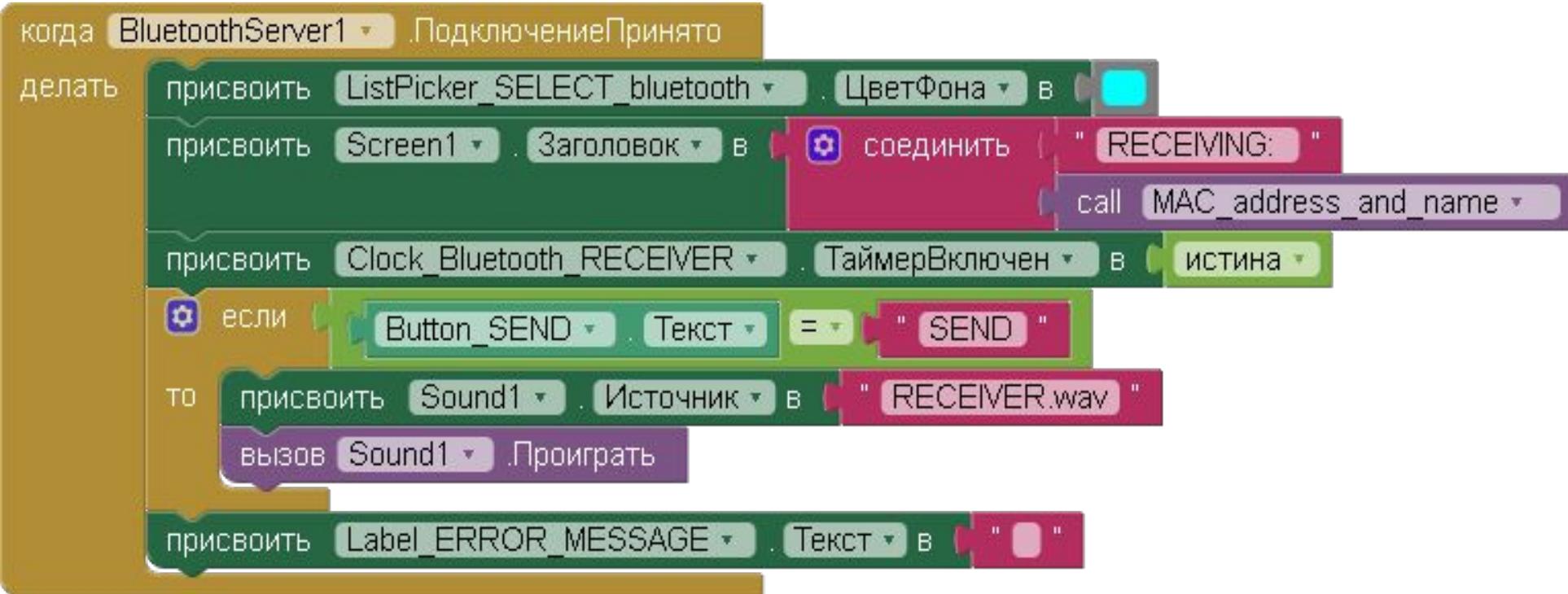
```
когда Clock_Bluetooth_ENABLE_SERVER Таймер
  делать
    если BluetoothClient1 Включено
      то
        если не BluetoothServer1 Приглашено
          то
            если не Clock_Bluetooth_RECEIVER ТаймерВключен
              то
                присвоить Clock_Bluetooth_RECEIVER ТаймерВключен в истина
```

```
когда Clock_Bluetooth_RECEIVER Таймер
  делать
    если BluetoothClient1 Включено
      то
        если вызов BluetoothServer1 БайтовДоступноДляПринятия > 0
          то
            присвоить global Bluetooth_RECEIVER в вызов BluetoothServer1 ПолучитьТекст
              числоБайтов вызов BluetoothServer1 БайтовДоступноДляПринятия
            присвоить Label_Bluetooth_RECEIVER Текст в ←
          вызов BluetoothServer1 Отключиться
          вызов BluetoothServer1 ПринятьПодключение
            названиеСервиса
          присвоить ListPicker_SELECT_bluetooth ЦветФона в
          присвоить Clock_Bluetooth_RECEIVER ТаймерВключен в ложь
```



```
соединить
  разбить по пробелам вызов Clock_Bluetooth_RECEIVER ФорматВремени
    немедленный вызов Clock_Bluetooth_RECEIVER Сейчас
  call MAC_address_and_name
  "\n"
  получить global Bluetooth_RECEIVER
  "\n"
  Label_Bluetooth_RECEIVER Текст
```

# Принцип работы



# Задания:

Требуется :

1. Нарисовать схематически логику типового приложения передачи данных.
2. Реализовать очистку данных в приложении.

# Вопрос:

1. В честь чего или кого **Bluetooth** получил свое название?
2. Как осуществляется в передача данных через **Bluetooth**?
3. Где **Bluetooth** может нам пригодиться?