Министерство образования и науки РФ
Федеральное государственное бюджетное образовательное
учреждение высшего профессионального образования
«Омский государственный университет им. Ф.М. Достоевского»
Кафедра компьютерных технологий и сетей.

УТВЕРЖДАЮ
Заведующий кафедрой
_____Лавров Д.Н.
«___»_____ 2014 г.

РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ ПЕРЕДАЧИ ДАННЫХ С ПОМОЩЬЮ СВЕТА

Курсовая работа
по направлению 090102.65 – Компьютерная безопасность

Научный руководитель:
к.т.н. , доцент
_____Лавров Д.Н.
«___»_____ 2014 г.

Выполнил:
студент группы СБС-001-О
_____ Лейфер К.И.
«___»_____ 2014 г.

Омск
2014

Содержание

Введение

В сетях передачи данных в качестве носителя информации часто используется видимое излучение. Однако, скорости, на которых осуществляется обмен данными, не позволяют человеку непосредственно наблюдать за передачей бит информации. К тому же, зачастую узнать, с какими целями при реализации протокола передачи данных физического уровня была выбрана та или иная схема кодирования, возможно только опытным путём.

Для того, чтобы на практике выяснить, как схемы физического и логического кодирования влияют на скорость передачи данных, помехоустойчивость, синхронизацию передатчика и приёмника и другие свойства сигнала, необходимо разработать приложение, которое позволит передавать небольшие объёмы данных на небольшие расстояния, используя видимый свет в качестве носителя информации. Приложение позволит пользователю выбирать алгоритм логического и физического кодирования, видеть передаваемую информацию в закодированном состоянии.

Целью моей курсовой работы является:

1) Разработка мобильного приложения «FlashTransmitter». Для передачи сигнала приложение будет использовать светодиодную вспышку, встроенную в большинство смартфонов, для приёма — датчик света.
2) Работа приложения со внешней библиотекой «FlashTransmitter-lib», обеспечивающей алгоритмы кодирования и декодирования информации, а так же возможность быстрого написания и внедрения собственных алгоритмов.

Глава 1. Обзор существующих технологий передачи информации с помощью света

Рассмотрим несколько технологий, использующих для передачи информации свет.

Наиболее широкое распространение получила волоконно-оптическая связь. Эксперименты по передаче данных в среде, обладающей свойством полного внутреннего отражения, проводились с 30х годов XX века. Первые сети передачи данных, построенные на оптоволокне появились в конце 70х годов. В течение последних трёх десятилетий инженерам удалось добиться существенного улучшения физических параметров оптоволокна (таких, как скорость затухания сигнала), что, вкупе с другими преимуществами данной технологией перед медными кабелями (меньшая масса, большая устойчивость к электромагнитным помехам, меньшая стоимость изготовления) делает волоконно-оптическую связь конкурентоспособной и востребованной.

Группа стандартов IrDA, получившая распространение в конце 1990х — начале 2000х годов, использует для передачи данных свет инфракрасного диапазона. Эта технология предусматривает передачу сигнала в воздушной среде. Расстояние передачи ограничено прямой видимостью между приёмником и передатчиком. В последней версии протокола физического уровня IrPHY, который на данный момент находится в состоянии разработки, ожидается поддержка скорости передачи данных в 100 Мбит/с.

Li-Fi – экспериментальная технология беспроводной передачи данных по воздуху с использованием обычных светодиодов одного или нескольких цветов. Учёным из Мюнхенского университета удалось достичь скорости передачи данных в 800 Мбит/с на расстоянии 1.8 м с использованием светодиодов 4-х цветов. Главная особенность проекта заключается в том, что в передаче данных в теории могут участвовать обыкновенные бытовые лампы, светодиоды внутри которых, благодаря высокой скорости мерцания, будут передавать данные незаметно для человеческого глаза. Это позволяет создать распределённую сеть передачи данных внутри квартиры и нивелирует главный недостаток подобных

технологий — необходимость приёмнику и передатчику находиться в прямой видимости [2].

Обоснование выбора ОС Android в качестве платформы для разработки мобильного приложения.

Приложение «FlashTransmitter» изначально планировалось разрабатывать под мобильные устройства, так как большинство из них наделены датчиком света и светодиодной вспышкой, то есть простейшими приёмником и передатчиком. При выборе мобильной операционной системы, под которую будет разработано приложение, учитывались такие факторы, как распространённость целевой ОС на рынке, порог вхождения в инструменты разработки, доступные для данной ОС, а так же возможность свободно и бесплатно разработать и протестировать приложение на большом количестве целевых устройств. Таблица 1 демонстрирует сравнительный анализ трёх наиболее популярных мобильных операционных систем на рынке: Android, iOS и Windows Phone.

Таблица 1. Сравнительные характеристики мобильных ОС в качестве платформ для разработки приложений

Характеристика	**ОС**		
	Android	**iOS**	**Windows Phone**
Основной язык программирования	Java	Objective-C	C#
Наличие онлайн-документации	Есть (http://developer.android.com)	Есть (http://developer.apple.com)	Есть (http://msdn.microsoft.com)
Наличие бесплатных средств разработки	Да (ADT, Android Studio)	Да (инструменты разработки привязаны к экосистеме Apple)	Да (бесплатная ознакомительная версия средств разработки

			ограничена в функционале)
Бесплатная регистрация разработчика	Да	Нет	Нет (студентам предоставляется скидка)
Бесплатная публикация приложений	Да	Нет	Нет
Возможность официально установить приложение на устройство, минуя магазин приложений	Да	Нет	Нет

Как видно из таблицы, в качестве целевой ОС наиболее предпочтительна именно Android, поэтому выбор был сделан в её пользу.

Глава 2. принцип работы приложения «FlashTransmitter».

Приложение «FlashTransmitter» может работать в двух режимах: в качестве передатчика сигнала и в качестве приёмника. В обоих режимах пользователю доступен выбор логического и физического кодирования. Два пользователя, использующие свои устройства для передачи данных, должны выбрать одинаковые схемы кодирования для правильного распознавания сигнала приёмником. Все доступные способы кодирования предоставляются отдельно написанной библиотекой «FlashTransmitter-lib».

При использовании устройства в качестве передатчика пользователю доступен выбор алгоритмов кодирования и скорости, на которой ведётся передача.

Если устройство используется в роли приёмника, пользователю доступен только выбор алгоритмов кодирования. Приёмник автоматически определяет начало передачи данных и скорость, с которой передаётся сигнал.

Глава 3. Описание функционала «FlashTransmitter»

Пользователь может выбрать роль своего устройства на стартовом экране приложения (см. рис. 3.1).

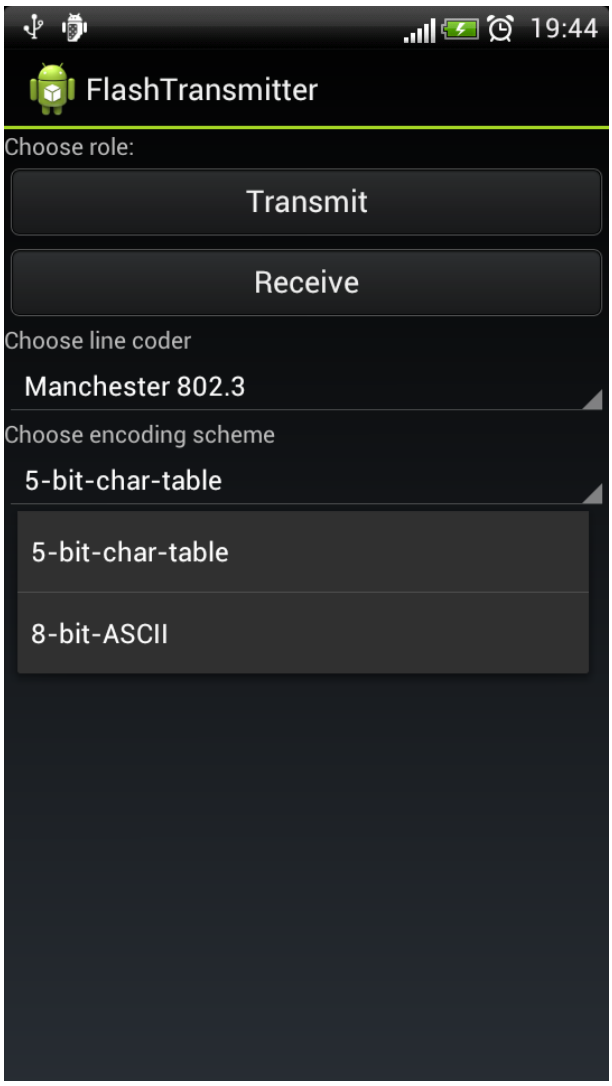

Рис. 3.1. Начальный экран приложения «FlashTransmitter»

На этом же экране доступен выбор схемы кодирования информации. Для того, чтобы все новые способы кодирования, реализованные в библиотеке «FlashTransmitter-lib» сразу же были доступны для использования, без изменения исходного кода приложения, выбор конкретной реализации интерфейсов кодирования информации происходит по принципу Dependency Injection.

В зависимости от роли, выбранной пользователем, открывается либо интерфейс передачи данных, либо интерфейс приёмника. Рассмотрим эти

экраны подробнее.

Работа приложения при передаче данных

Для передачи данных пользователь вводит текст, который хочет передать, выбирает частоту (Гц), с которой будет вестись передача и нажимает кнопку «отправить» (см. рис. 3.2)

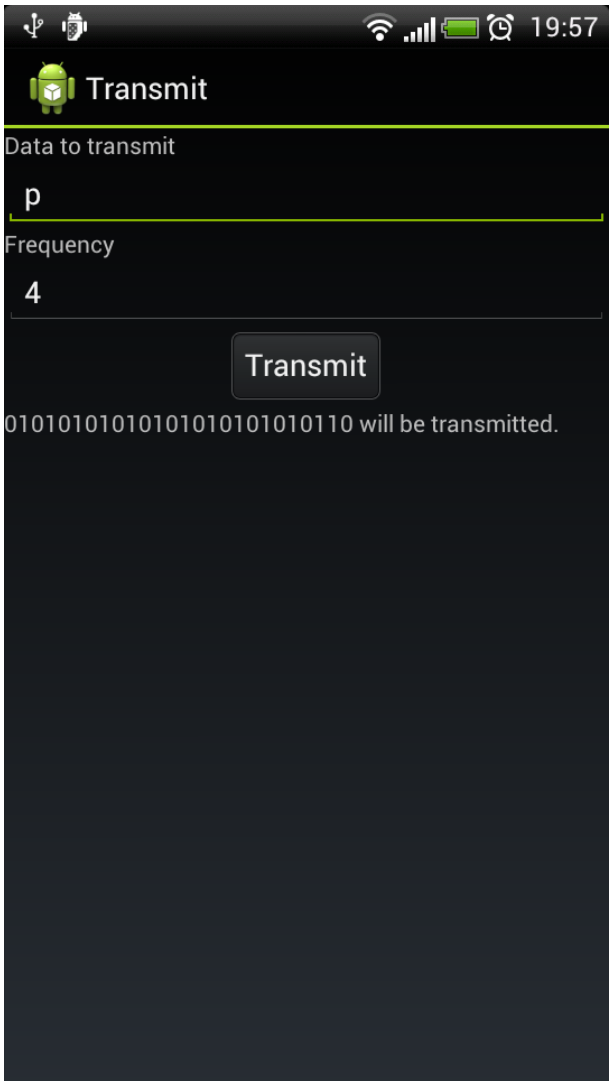

Рис. 3.2. Интерфейс передатчика

Работа передатчика состоит из следующих этапов:

1) Данные, введённые пользователем, преобразуются в массив состояний вспышки в библиотеке «FlashTransmitter-lib» в соответствии с выбранной схемой линейного и физического кодирования.
2) В отдельном потоке запускается процедура, которая с помощью вызова

API камеры меняет состояние вспышки в соответствии с полученным массивом состояний. Время, в течение которого вспышка находится в том или ином состоянии, вычисляется, исходя из частоты, выбранной пользователем. Работа с API камеры реализована в отдельном потоке, так как в ином случае обработка элементов пользовательского интерфейса была бы заблокирована в процессе передачи данных, и отменить его было бы невозможно.

3) Для того, чтобы отменить передачу данных, пользователь должен нажать аппаратную кнопку «назад», присутствующую во всех Android-устройствах.

Работа приложения при приёме данных

Для работы приёмника не требуется вмешательство пользователя, пользовательский интерфейс состоит, главным образом, из текстовой строки, отображающей полученные данные и некоторых других полей, необходимых для отображения дополнительной информации (текущая яркость, состояние передачи данных, параметры датчика света) (см. рис. 3.3)

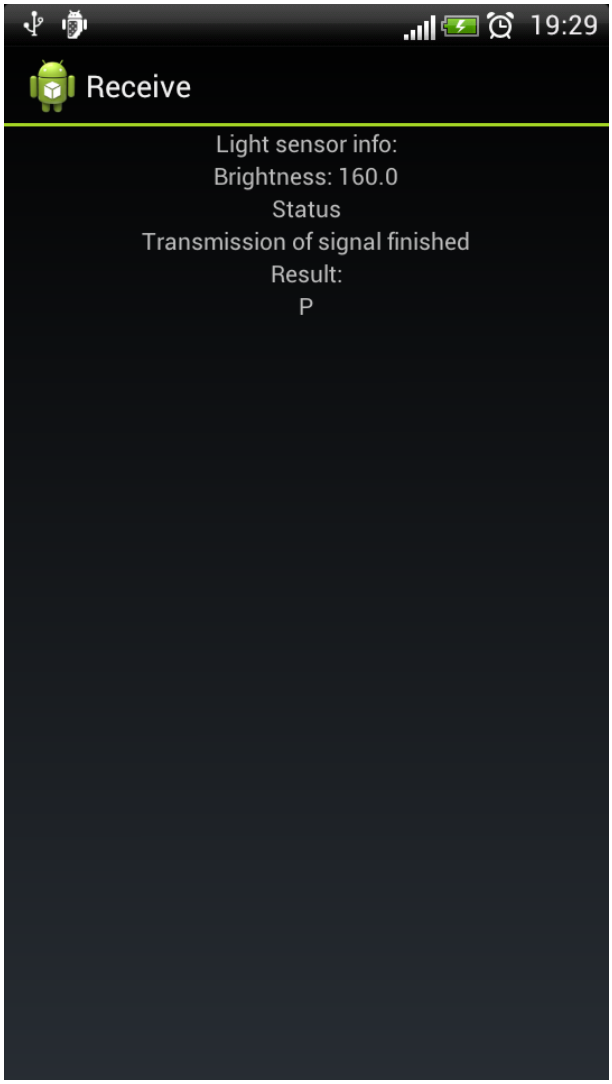

Рис. 3.3. Пользовательский интерфейс приёмника.

Приёмник сигнала определяет уровень яркости с помощью вызова API сенсоров (в нашем случае — датчика света). При выделении сигнала из показаний датчика света, приложение учитывает несколько особенностей работы датчика света:

1. Этот сенсор был разработан для подстройки яркости дисплея, поэтому имеет довольно существенную задержку. Это не критично при

использовании сенсора по назначению, но становится трудностью при передаче данных (приложение было протестировано на большом количестве различных Android-устройств, результаты показали задержку, в среднем, около 50 мс).

2. В ОС Android при написании драйвера к датчику света, разработчик имеет возможность объявить характеристики датчика, такие, как максимальная яркость или минимальная задержка сенсора, непосредственно в драйвере. На практике, при разработке приложения «FlashTransmitter», мне пришлось столкнуться с тем, что ни в одном из тестируемых устройств эта информация не соответствовала истине. Поэтому приёмник самостоятельно определяет эти параметры во время приёма сигнала.
3. При обычных условиях освещённости количество света, которое принимает фотофильтр смартфона, постоянно меняется. Поэтому приёмник отфильтровывает небольшие колебания и фиксирует только значительные изменения яркости, которые вероятнее всего соответствуют полезному сигналу.
4. Android Sensors API устроен таким образом, что у разработчика нет возможности узнать яркость в текущий момент времени. Вместо этого API предоставляет возможность «подписаться» на получение новых данных от датчика света. В этом случае при любом изменении яркости будет вызвана функция onSensorChanged().
5. Приёмник не имеет информации о том, какая частота выставлена на передатчике. К тому же, приёмнику неизвестно, когда начнётся передача сигнала.

Из перечисленных выше особенностей становится ясно, что фиксировать сигнал по уровню яркости — очень сложная задача, так как уровень яркости зависит от условий среды, в которой осуществляется передача данных, к тому же, уровень фоновой яркости может поменяться непосредственно в процессе

передачи. Таким образом, было принято решение фиксировать не яркость, а её изменение (резкий рост или, наоборот, резкий спад соответствуют переднему или заднему фронту волны).

До того момента, когда приёмник фиксирует резкий рост яркости, он собирает статистику о состоянии среды передачи — фоновая яркость, минимальная задержка при получении новых значений яркости.

Когда приёмник фиксирует первый фронт волны, он начинает запоминать временные отметки всех фронтов волн. В то же время в отдельном потоке запускается процедура, которая сравнивает время, в течение которого не поступал новый фронт волны с максимальным временным отрезком между двумя соседними зафиксированными фронтами. Если время, в течение которого не был зафиксирован резкий рост или спад яркости, превышает максимальное расстояние между соседними фронтами волн на определённую величину, передача данных завершается. Такая мера с отдельным потоком необходима, исходя из п. 4 особенностей датчика света. Если производить проверку внутри функции onSensorChanged(), то при длительном отсутствии изменения яркости передача данных не завершится никогда.

После того, как приёмник отметил завершение сеанса связи, полученный массив временных отметок передаётся на декодирование библиотеке «FlashTransmitter-lib». После декодирования полученная строка отображается на экране приложения.

Заключение

Написанное приложение позволяет передавать небольшие объёмы данных на небольшие расстояния (40-80 см, в зависимости от условий освещения) с помощью видимого излучения, а так же использовать различные способы кодирования информации, расширять функционал приложения собственными алгоритмами кодирования. Таким образом, опытным путём можно изучить свойства различных алгоритмов кодирования данных.

Список использованной литературы:

1. Рето Майер «Android 4. Программирование приложений для планшетных компьютеров и смартфонов», Эксмо, 2013г.
2. Daten unterwegs mit Licht

http://www.fraunhofer.de/de/presse/presseinformationen/2011/august/daten-unterwegs.html

3. Android Camera API reference

http://developer.android.com/reference/android/hardware/Camera.html

4. Android Sensor API reference

http://developer.android.com/reference/android/hardware/Sensor.html