Выбор технологии связи, корпус, энергопотребление, печатная плата, безопасность, особенности бизнес-модели

Эта лекция для тех студентов, кто уже сделали выпускной проект, и теперь задача - превратить прототип в MVP. Рассмотрены те проблемы, которые есть у студентов в 90% случаев. Лекция носит технический характер и посвящена доработке проекта перед финалом конкурса.

Темы:

1. Выбор технологии связи
2. Корпус
3. Энергопотребление
4. Печатная плата
5. Безопасность
6. Особенности бизнес-модели в IoT

<https://youtu.be/DkSoXYPp0-8>

*Краткий конспект лекции ниже:*

Всем привет! Меня зовут Татьяна Волкова, я куратор трека по Интернету вещей в проекте Samsung Innovation Campus. Каждый год я просматриваю примерно 150 учебных проектов, которые студенты предоставляют в качестве выпускных аттестационных проектов. Многие из них очень хороши, однако пока еще находятся на стадии прототипа, и пока не достигли стадии MVP.

Давайте рассмотрим основные шаги, которые вам нужно сделать, прежде чем проект можно будет демонстрировать заказчику, подавать на конкурс или оформлять в заявке на грант. Сейчас я поговорю только о технической стороне вопроса, не затрагивая продуктовую. Будем считать, что у вас уже есть свежая и практически применимая идея, и вам нужно проработать ее технически.

**Выбор технологии связи**

Одна из самых часто встречающихся ошибок - выбор неоптимальной технологии связи. Как вы все знаете, в Интернете вещей в большинстве решений используется беспроводная связь, и существует множество вариантов, как ее можно реализовать. И вот здесь часто бывает следующая ошибка: студенты берут не ту технологию, которая лучше подходит, а ту, которую они уже знают.

Приведу реальный пример. Студенты делали систему управления для достаточно дорогого прибора, его стоимость в сборе по моим оценкам была 50-100 тысяч рублей. Комплектующие для прибора были предоставлены кафедрой. От них же требовалось собрать прибор и сделать для него систему дистанционного управления. К моему удивлению, они выбрали известный многим радиомодуль NRF24L01, а у него, помимо плюсов, таких как дешевизна и низкое энергопотребление, есть и минусы - модуль капризный, теряет пакеты, часто попадается заводской брак. В данном случае не очень понятно, зачем было брать дешевый модуль там, где можно было бы взять привычный всем WiFi или Bluetooth, тем более для такого дорогого прибора.

Другой пример. Студентка делала систему “Умный шлагбаум” для многоквартирного дома. Вроде бы правильная идея системы: есть Telegram-чатбот, через которого жильцы дома могут выдавать разовые пропуска на проезд для, например, курьеров. Очень удобно. Но была одна проблема: девушка выбрала связь по WiFi. И на вопрос, а собственно откуда на придомовой территории возьмется WiFi сеть, у нее не нашлось ответа. Действительно, точки доступа у жильцов обычно в квартирах, а вот во дворе вряд ли есть. А ведь достаточно было сменить протокол связи на GSM, например, и такого вопроса не возникло бы.

Чтобы убедиться, что вы сделали правильный выбор, задайте себе вопросы:

* Где будет находиться устройство? Дома? В лесу? На предприятии? На улице?
* Есть ли в планируемом месте установки ограничения по используемым сетям? Пример: в некоторых учреждениях сети WiFi запрещены либо в них закрыты некоторые порты.
* Если устройство соединяется с точкой доступа, то где находится точка доступа и кто поддерживает ее функционирование? Как в примере со шлагбаумом.
* Если возможности подключиться к Интернету нет, то как мы сможем работать с данными, которые собрало устройство? Возможен ли вариант по типу “черного ящика”, где мы спустя промежуток времени собираем данные?
* Если канал связи узкий, то возможно ли обрабатывать данные (или делать их предобработку) прямо на устройстве?

**Корпус**

Если мы говорим о каком-то минимальном MVP, то уже необходим корпус. Для тестов, для эскизного проекта, годится и картонная коробка. Но если показывать проект заказчику, то нужен корпус хотя бы в минимальном виде.

Из возможностей, доступных студенту, самая простая технология изготовления корпуса - это 3D-печать.Сейчас во многих вузах стоят 3D-принтеры с общим доступом.

Даже если вы не владеете навыками работы в специализированных пакетах, таких как Autodesk Fusion или Компас 3D, вы всегда можете набросать дизайн простого корпуса в бесплатной онлайн-среде TinkerCAD от той же компании Autodesk, причем даже сделать это не с нуля, а взяв один из готовых бесплатных шаблонов, библиотеки которых есть в Интернете. Полученный на выходе файл годится для 3D печати, даже если вы никогда этим не занимались.

Печатный корпус можно сделать не только функциональным, но и дизайнерски красивым, чтобы его было приятно взять в руки. Например, классный корпус сделали студенты МЭИ Лиза Голубева и Андрей Дергачев, и мне он очень понравился с первого взгляда. Причем ребята не сразу смогли добиться красивой формы, кое-что пришлось переделывать с учетом ошибок. Подробнее об их проекте вы можете прочитать в [статье](https://habr.com/ru/companies/samsung/articles/749472/) на Хабре.

Если доступа к 3D-принтеру нет, то какие остаются варианты? Во-первых, всегда можно купить корпус, в качестве примера приведу корпус за 300 рублей, который я купила для своего устройства в одном из интернет-магазинов. Это корпус на DIN-рейку. Такие корпуса бывают разных размеров, и вы точно сможете найти подходящий. Покупала я корпус с большим запасом, чтобы было куда разместить провода и разъемы. Уменьшение - это уже следующий шаг.

И наконец, если даже 300 рублей вам жалко тратить, то всегда есть вариант найти что-то из подручных материалов. К примеру, делая проект “Умный кубик”, я обратила внимание на очень красивую и удобную упаковку из плотного картона. Упаковка была от товара - настоящего кубика Рубика. Не выброшенная коробка оказалась идеально подходящей для моего устройства, и если не знать заранее, то можно подумать, что это MVP. Конечно, такое подойдет только для самой первой версии устройства, но это лучше, чем ничего.

Подведу итог. Можно условно разделить все эти стадии корпуса на уровни, как в компьютерной игре.

* Уровень 0 - корпус отсутствует, устройство представляет собой разрозненные соединенные проводами модули или отдельную печатную плату
* Уровень 1 - корпус из картонной коробки или иных подручных материалов
* Уровень 2 - стандартный покупной корпус из пластика
* Уровень 3 - кастомизированный под конкретно ваше устройство корпус, напечатанный на 3D-принтере.

**Печатная плата**

Если у вас предполагается несколько модулей в вашем устройстве, даже всего лишь 2 или 3, то уже как правило хочется сделать печатную плату. Провода запутываются, беспаечная макетка плохо держит элементы, транспортировку прототип не выдерживает. Если вы решили все же остановиться на этой стадии и печатную плату не делать (например, вам не хватает навыков), то как минимум из документации подготовьте монтажную схему, чтобы было видно, как соединяются модули в вашей системе.

* Уровень 0 - это отсутствие печатной платы, монтаж на проводах или беспаечной макетке. Навыки кроме знания основ схемотехники не требуются.
* Уровень 1 - Макетная плата с впаянными компонентами. Потребуется навык пайки. По деньгам это недорого, требуется только сама макетка.
* Уровень 2 - Печатная плата, изготовленная подручными средствами (ЛУТ, фоторезист). Потребуется навык разводки печатной платы в редакторе. По деньгам это по-прежнему очень бюджетно. Печатная плата будет поводом навести порядок в вашем проекте, ведь без электрической принципиальной схемы ее уже не сделаешь. Здесь тоже есть уровни, как и с корпусом. На мой взгляд, сделать печатную плату не требует особых финансовых вложений - если говорить о ПО, то если у вас нет лицензии на Altium, то всегда есть бесплатные и кроссплатформенные редакторы, такие как KiCAD и Eagle. Текстолит и макетки стоят недорого. Но при этом, плата - сразу большой плюс к проекту, поскольку с ней проект действительно выглядит законченным и аккуратным.
* Уровень 3 - Печатная плата, изготовленная промышленным методом. Потребуется доступ к оборудованию или финансовые возможности - тут уже, к сожалению, вложения побольше, плюс возможно с первого раза сделать хорошую плату не выйдет.

**Безопасность**

С безопасностью хорошо то, что ничего нового здесь придумывать не нужно. Всего лишь нужно знать типовые виды атак и смотреть, защиту от каких атак предлагает производитель ПО/железа, которым вы пользуетесь. И в большинстве случаев вам будет этого достаточно. Изучайте стек от производителя.

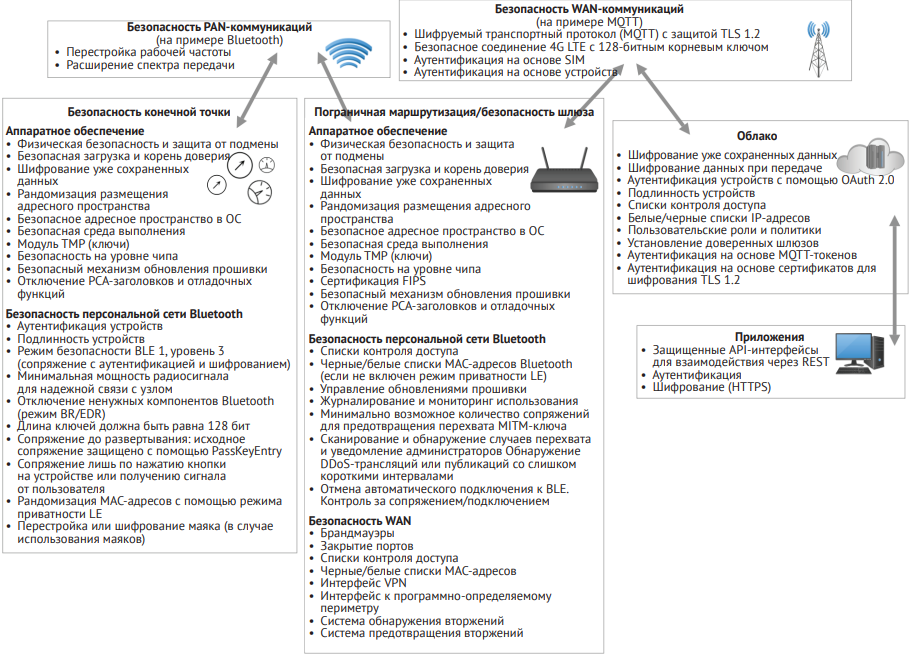
Приведу пример. Допустим, вы решили использовать в проекте технологию ESP-NOW - это упрощенный протокол связи WiFi, предлагаемый компанией Espressif, которая делает микроконтроллеры серии ESP. Этот протокол немного более энергоэффективный, и проще в работе, чем WiFi, то есть больше подходит для IoT. Если все делать как написано в тьюториалах (например на randomnerd), то вы научитесь отправлять и получать сообщения через этот протокол, там все очевидно, создается callback функция и так далее. Но если вы этим ограничитесь, то вы никогда не используете шифрование и все сообщения будут идти незашифрованными. И только если вы будете искать тьюториал по ключевым словам ESP-NOW encrypt, то появится ссылка на другой тьюториал, где уже объясняется, как назначить ключи. Вот таким простым способом уже можно повысить безопасность своей системы. Почитать о ESP-NOW можно в [статье](https://habr.com/ru/articles/502260/) на Хабре.

Теперь, допустим, вы изучили, что предлагается по безопасности на уровне конкретных компонентов вашего устройства. Как перейти на уровень безопасности вашей системы в целом? Ответ такой. Нужно составить модель угроз, то есть перечислить все возможные нежелательные ситуации, и выработать реакции на каждую из этих ситуаций. Здесь сразу разберемся с терминологией: в чем разница между угрозой и уязвимостью? Все очень просто, угроза - это некоторая опасность, а уязвимость - то узкое место в системе, через которую эта угроза реализуется.

Есть аббревиатура ФСТЭК (Федеральная служба по техническому и экспортному контролю), в большинстве случаев в России вас в различных учреждениях будут спрашивать на соответствие именно ФСТЭКовской методике. На сайте ФСТЭК существует раздел [БДУ](https://bdu.fstec.ru/threat) - база данных угроз, все возможные по российским стандартам угрозы можно прочитать там. В базе достаточно много угроз, которые вам изначально не подходят, например там есть угрозы, связанные с машинным обучением - их сразу пропускаете, если не используете этих технологий. Выбираете только те угрозы, которые релевантны. Для облегчения работы берете готовые списки, например на сайте БДУ есть раздел “Типовые уязвимости веб-приложений”. Затем составляете себе табличку, где в левом столбце угрозы, а в правом - принятые меры безопасности.

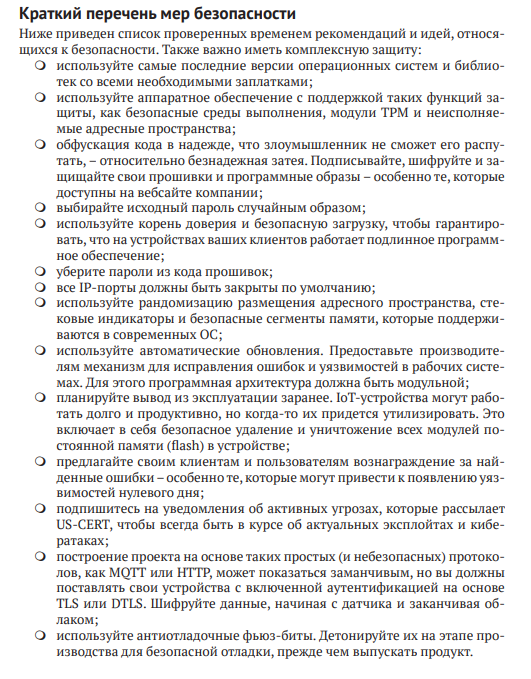
Изучая список уязвимостей, вы поймете, что многие перечисленные здесь ситуации вам хорошо известны и по сути являются прописными истинами: что многофакторная аутентификация лучше, чем однофакторная. Что пароли нельзя хранить в чистом виде, а нужно хранить хэши. Что нужно ограничивать попытки логина, чтобы вас не взломали перебором. В целом я не буду заострять внимание здесь, а просто отошлю вас к неплохой [статье](https://habr.com/ru/companies/ic-dv/articles/457516/) на Хабре, где рассказывается о процессе составления модели угроз по методике ФСТЭК.

Теперь, что делать, если вас интересуют конкретно рекомендации именно по типовым системам Интернета вещей, и вам не очень интересно просматривать весь список возможных угроз? В этом случае вы можете ознакомиться с готовыми схемами. Я взяли пример из книги Перри Ли - “Архитектура интернета вещей”.



Как вы видите по картинке, безопасность Интернета вещей сложно обеспечить именно из-за разнородного вертикального стека используемых технологий, и здесь по-хорошему необходима слаженная работа нескольких специалистов.

Также в книге Перри Ли имеется краткий список - своего рода чеклист - именно по самым частым уязвимостям в системах Интернета вещей. Пробегитесь по нему и подумайте, что из этого относится к вашей системе:



Подведем итог:

Уровни  
0. Безопасность отсутствует в принципе, вы надеетесь на добрые намерения всех, кто будет иметь дело с вашим устройством  
1. Вы посмотрели стек, с которым работаете, и воспользовались заложенной производителем функцией.  
2. Вы комплексно изучили вашу систему и сделали аудит безопасности, построив модель угроз.

**Энергопотребление**

Я всегда хочу знать ответ на простой вопрос. Если ваше устройство работает от аккумуляторов, то сколько оно проработает? Годится даже примерная оценка. В идеале - измерение.

Вопросы для самопроверки:

* Уходит ли ваше устройство в режим сна между сеансами коммуникации?
* Возможно ли повысить энергоэффективность устройства, сменив технологию связи?
* Предусмотрена ли индикация о разряде аккумулятора? Звуковая/светодиодная? В приложении пользователя?
* Кто будет заряжать аккумуляторы, когда они сядут? Как устроен этот процесс логистически?

**Особенности бизнес-модели в IoT: долгий жизненный цикл продукта; емкость рынка**

*За эту подглаву благодарим автора — преподавателя РТУ МИРЭА Антона Миронова*

И все же нельзя не затронуть продуктовую сторону, как бы этого ни хотелось.

Стоит подумать о вопросах поддержки "сложного и длинного" жизненного цикла системы интернета вещей, не превращая устройство в кирпич. А он и правда сложный и длинный. Пользователь покупает вещь, и не всегда понимает, что это такой же гаджет, как и многое вокруг него.

И задача производителя - обеспечить поддержку этого гаджета на протяжении всего периода жизни. Обновления ПО, обеспечение совместимости со старыми устройствами, возможность обслуживания без участия пользователя. Иначе рано или поздно пользователь останется с кирпичом. Особенно к этому чувствителен массовый b2b сегмент - умные витрины, системы опросы приборов учета и т.д.

Компании, работающие на индивидуальных заказчиков, имеют свои инжиниринговые мощности, и у них все, как правило, делается под уникальный техпроцесс и за очень большие бюджеты.  А вот массовый сегмент — шеринги, управляющие компании, аграрии и прочие — живут на готовых платформах и полностью от них зависят. И подводить клиентов в такой ситуации для платформы — верный путь к провалу, потому что доверие зарабатывается долго, а теряется очень быстро.

Возрастает роль стандартизации и Open Source, потому что ничто не вечно, и у компании будут и слияния, и поглощения, и кто-то с рынка уйдет вообще. В этих условиях,  будет надежнее пользоваться стандартными/открытыми решениями для лучшей взаимозаменяемости.

Еще одна проблема - емкость рынка и очень долгая петля возврата. Устройства служат подолгу, а продавать что-то надо. Так называемый "Парадокс игры Эволюция". В финале появляется суперхищник, который съел всех и умер от голода. Тут так же: с насыщением рынка большая компания должна или похудеть, или найти новые рынки, или она не сможет прокормиться. Поэтому идея, что "вот станем большими - и все будет хорошо" на самом деле иллюзия.

Проблема решается как раз моделью подписки и сервиса, но может вести к потере ключевых кадров. Почему? Модель подписки и сервиса при не очень большой емкости рынка приводит к тому, что команда делает продукт, потом основные усилия уходят на поддержку и  сервис. При этом наиболее квалифицированным инженерам интереснее заниматься разработкой, а не поддержкой, и они уходят.

Как итог, следующее поколение устройств делать будет новое поколение инженеров, потому что процессы идут волнами, следом за первым бумом технологии. Когда приходит время массового обновления парка решений (который часто строится за год-два) от первоначальной команды людей, из тех, кто стоял у истоков, остается 1-2 человека. Это опять же, беды средних копаний, у крупных есть поток заказчиков и  непрерывный RnD-процесс.

В заключение, хотелось бы посоветовать вам книгу:  
Алексей Корнилов - Основы проектирования приложений Интернета вещей