# Оглавление

# Глоссарий и термины

# Общие сведения

Название: Проект «Bayqaw»

Тип работ: Научно-исследовательская и конструкторская работа

Срок выполнения: 31 августа 2018 г.

Ответственное лицо: Оспан Б.Б.

Уровень сложности: 7 из 10

Ссылка на Github репозиторий: https://github.com/clevtech/bayqaw

Ссылка на документацию: https://github.com/clevtech/bayqaw/wiki

Описание: Научно-исследовательская и конструкторская работа была заказана для проведения исследований в области применения BLE устройств ESP32 для создания совместного проекта «Bayqaw» и сопутствующих проектов «Bayqaw - СКУД» и «Bayqaw - Считыватель».

Цели:

* Работоспособность ESP32 как BLE устройства:
* Проверка гипотезы Mesh Network на основе BLE:
* Проверка гипотезы BNC:

# Теория протокола "BLE"

Беспроводная технология Bluetooth с низким энергопотреблением (далее BLE) – ядро протокола коммуникации по Bluetooth нового поколения, выпущенная в 2009 году. На данный момент практический все девайсы поддерживают данный протокол. Особенности данного протокола:

* **Сверхмалое пиковое энергопотребление** – оборудование, работающее на технологии BLE имеют очень низкое энергопотребление.
* **Малый объем информации** – BLE девайсы могут передавать только небольшой объем данных (от 125 кб/с до 2 мб/с).
* **Работа с несколькими топологиями сети** – протокол позволяет работу сразу в нескольких режимах топологии, то есть: возможность рекламировать определенную информацию всем устройствам или передавать информацию только одному устройству.

**Основные характеристики сети:**

* Частота: 2.4GHz ISM Band (2.402 – 2.480 GHz Utilized)
* Количество каналов: до 40, зависит от оборудования (2 Mhz на канал)
* Сила сигнала: от 1 mw до 100 mw
  + Class 1: 100 mW (+20 dBm)
  + Class 1.5: 10 mW (+10 dbm)
  + Class 2: 2.5 mW (+4 dBm)
  + Class 3: 1 mW (0 dBm)
* Скорость соединения: от 125 кб/с до 2 мб/с
  + LE 2M PHY: 2 Mb/s
  + LE 1M PHY: 1 Mb/s
  + LE Coded PHY (S=2): 500 Kb/s
  + LE Coded PHY (S=8): 125 Kb/s
* Тип использования каналов: Frequency-Hopping Spread Spectrum
* Тип модуляции каналов: GFSK
* Максимальный размер данных: 251 byte (при mesh 29 byte)
* Максимальное количество нодов в режиме mesh: 32,767
* Максимальное количество суб-mesh сетей: 4,096 ​

**Профили связи BLE:**

* GAP: профиль общего доступа
* GATT: профиль общих атрибутов

# Подготовка к разработке

Далее приведены инструкции по подготовке среды разработки.

Характеристики разработки:

* Язык программирования: C++
* Фреймворк: Arduino
* IDE: Atom PlatformIO
* Микроконтроллер: ESP32

Инструкция по установке зависимостей:

* Необходимо установить среду разработки Atom:
  + Ссылка на установку: <https://flight-manual.atom.io/getting-started/sections/installing-atom/>
* Необходимо установить надстройку PlatformIO, она автоматический установит все необходимые зависимости:
  + <http://docs.platformio.org/en/latest/ide/atom.html#ide-atom>
* Необходимо установить GitHub:
  + Ссылка на установку: <https://gist.github.com/derhuerst/1b15ff4652a867391f03>
  + Ссылка на инструкцию по пользованию: <https://habr.com/post/125799/>

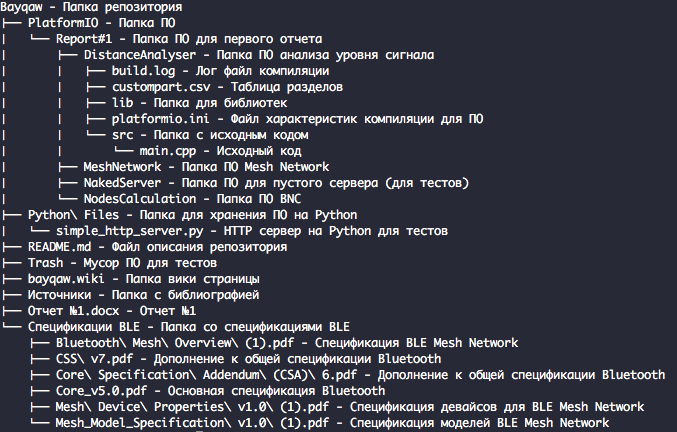
Начало работы:

* Необходимо скачать репозиторий разработки:
  + Ссылка на репозиторий: <https://github.com/clevtech/bayqaw>
* Далее необходимо открыть проект разработки в редакторе Atom:
  + На главной странице PlatformIO (открывается в вкладке Platformio в меню Atom) нужно выбрать открыть проект.
  + Далее необходимо выбрать репозиторий «Bayqaw».
* ВАЖНО! Необходимо каждую папку внутри папки «Platformio» в репозитории «Bayqaw» открыть отдельной вкладкой.
* Инструкция по работе со средой PlatformIO находится по ссылке: <https://www.losant.com/blog/getting-started-with-esp32-and-platformio>

# Репозиторий «Bayqaw»

Репозиторий «Bayqaw» находится в сервисе «Github» по ссылке: <https://github.com/clevtech/bayqaw>

Репозиторий имеет данную структуру:



# GAP и GATT профили

Стек протокола Bluetooth разделен на две категории: контроллер и хост. Каждая категория имеет подкатегории, которые выполняют определенные роли. Мы рассмотрим две подкатегории: общий профиль доступа (GAP) и профиль общих атрибутов (GATT).

GAP определяет общую топологию сетевого стека BLE.

GATT подробно описывает, как передаются атрибуты (данные), когда устройства имеют выделенное соединение.

GATT специально фокусируется на том, как данные форматируются, упаковываются и отправляются в соответствии с описанными правилами. В сетевом стеке BLE протокол атрибутов (ATT) тесно связан с GATT, где GATT находится непосредственно поверх ATT. GATT фактически использует ATT для описания того, как данные обмениваются с двух подключенных устройств.

**Общий профиль доступа (GAP)**

Существует два механизма, которые устройство BLE может использовать для связи с внешним миром: трансляция или соединение. Эти механизмы подчиняются основным принципам доступа (GAP). GAP определяет, как устройства с поддержкой BLE могут сделать себя доступными и как два устройства могут напрямую взаимодействовать друг с другом.

**Подключение**

Устройство может присоединиться к сети BLE, приняв эти роли, указанные в GAP:

* **Broadcasting**: для данного типа соединения, не обязательно соединения устройств с друг-другом.
  + **Broadcaster**: устройство, которое транслирует пакеты общедоступных рекламных данных, например, как долго нажата кнопка.
  + **Observer**: устройства, которые прослушивают данные в рекламных пакетах, отправленных вещателем. Между вещателем и наблюдателем не происходит никакого соединения.
* **Соединение.** Для данных типов ролей соединение должно быть между устроиствами.
  + **Периферийное устройство (сервер):** устройство, которое рекламирует свое присутствие, поэтому центральные устройства могут установить соединение. После подключения периферийные устройства больше не передают данные другим центральным устройствам и остаются подключенными к устройству, принявшему запрос на соединение.
    - Периферийные устройства являются маломощными, потому что они должны периодически отправлять маяки. Центральные устройства отвечают за установление связи с периферийными устройствами.
    - I-beacon является примером периферийного устройства BLE.
  + **Центральное (клиент):** устройство, которое инициирует соединение с периферийным устройством, сначала прослушивая рекламные пакеты. Центральное устройство может подключаться ко многим другим периферийным устройствам.
    - Когда центральное устройство хочет подключиться, оно отправляет пакет данных запроса на соединение к периферийному устройству. Если периферийное устройство принимает запрос от центрального устройства, устанавливается соединение.
    - Iphone является примером клиента, когда он подключается к I-Beacon.

Центральные устройства могут обновлять параметры подключения. Центральное устройство обычно устанавливает параметры соединения между периферийным устройством и им самим. Центральное устройство может изменять только параметры подключения. Однако периферийное устройство может попросить центральное устройство изменить параметры соединения.

Периферийные или центральные устройства могут завершать подключения. Соединения могут заканчиваться по разным причинам: аккумулятор устройства может сесть или сетевые помехи могут привести к сбою соединения. Устройства также могут намеренно отключиться от своих серверов.

**Профиль общего атрибута (GATT)**

**Роли**

* **Клиент:** отправляет запрос на сервер GATT. Клиент может читать и/или записывать атрибуты, найденные на сервере.
* **Сервер:** Одной из главных ролей сервера является сохранение атрибутов. Как только клиент делает запрос, сервер должен сделать доступными атрибуты.

Роли GAP и GATT по существу независимы друг от друга. Периферийные или центральные устройства могут действовать как сервер или клиент, в зависимости от того, как течет поток данных.

У GATT серверы передают два типа информации:

* Service UUID – 36-значный HEX код, определяющий сам сервис, то есть характеристики сервера.
* Characteristic UUID – 36-значный HEX код, определяющий характеристику (поток информации) внутри сервиса (до 37 характеристик).

Примером Characteristic UUID является характеристика «Alert».

|  |  |
| --- | --- |
| **Key** | **Value** |
| 0 | Simple Alert: General text alert or non-text alert |
| 1 | Email: Alert when Email messages arrives |
| 2 | News: News feeds such as RSS, Atom |
| 3 | Call: Incoming call |
| 4 | Missed call: Missed Call |
| 5 | SMS/MMS: SMS/MMS message arrives |
| 6 | Voice mail: Voice mail |
| 7 | Schedule: Alert occurred on calendar, planner |
| 8 | High Prioritized Alert: Alert that should be handled as high priority |
| 9 | Instant Message: Alert for incoming instant messages |
| 10 - 250 | Reserved for future use |

Таблица определений для сервиса «Alert».

# Теория "Mesh Network"

Mesh Network – вид топологии сети, при которой каждая ячейка (далее нода, node) может нести в себе роль коммутатора для следующего элемента сети. Основной характеристикой сети является ее отказоустойчивость.

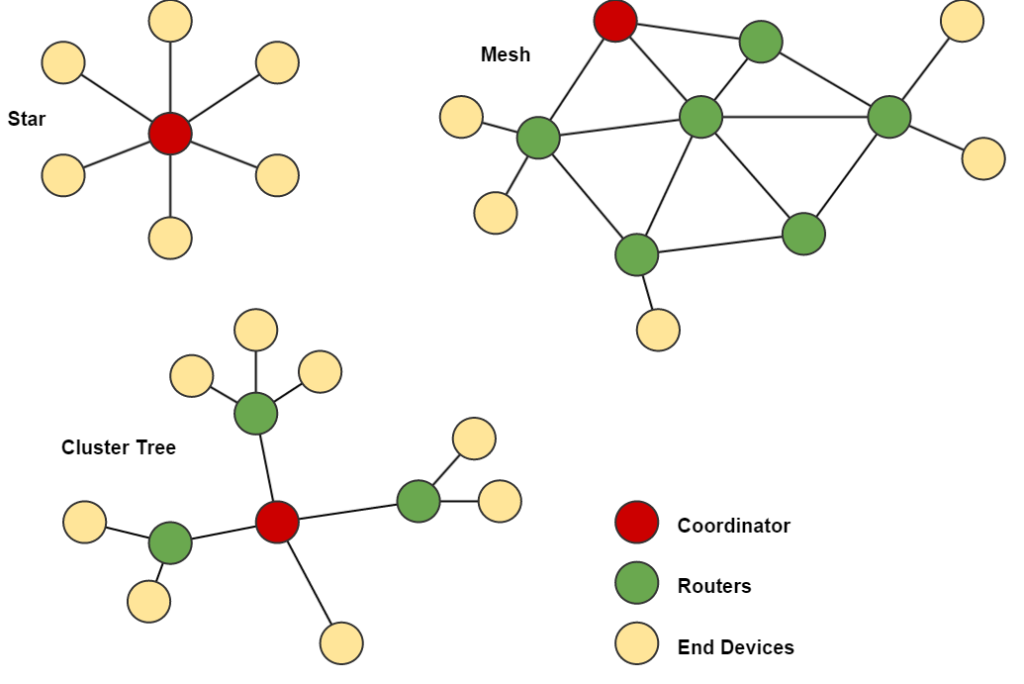


Рисунок 1 - Основные топологии сетей

Хотя большинство исследователей изучают топологию данной сети, но в большинстве своем существуют ряд различных ограничений теории данной топологии. Например, большинство из существующих сетей данной топологии, по сути являются разновидностью самоорганизующейся сети типа кластерного дерева или типа иерархического распределение сети. Из описаний топологии сети разными исследователями (Aravind Iyer, 2009) (John Bicket, 2005) (Yang Cao, 2006), можно определить основные характеристики сети для дальнейшей работы:

Характеристики топологии:

1. **Отказоустойчивость:** по причине того, что каждый элемент сети является и приемником и коммутатором, при выходе из строя количества ячеек n (при n << m, где m – общее количество ячеек сети), сеть продолжает работать, конечно без учета данных с нерабочих нодов.
2. **Масштабируемость:** разветвление и увеличение сети должно производиться без особого контроля.
3. **Самоорганизация:** сеть, построенная на данном типе топологии, должна поддерживать основы принципа управления робототехники «Рой», и само организовываться на иерархии (подробно в 2.1.).

# Проверка уровней сигнала BLE

Для проверки уровня сигнала был разработан эксперимент «Уровень сигнала BLE».

* + - * Цель: эксперимент был разработан и проведен для замера уровней сигнала между сервером и клиентом в разных ситуациях:
        + Сигнал в «чистом поле», без помех и преград
        + Сигнал через стандартные двери офисных помещений
        + Сигнал через бетонную стену
        + Сигнал через стеклянные двери
      * Методы: были разработаны две системы для измерения сигнала:
        + Голый сервер: рекламный GAP сервер с пустой информацией
        + Измеритель: GAP клиент с экраном на базе ESP32 для измерения уровня сигнала
      * Исходные коды: исходные коды по эксперименту хранятся в репозиторий проекта:
        + Голый сервер: bayqaw/platformio/Report#1/NakedServer
        + Измеритель: bayqaw/platformio/Report#1/DistanceAnalyser

# Сигнал в «чистом поле»

Сигнал в чистом поле был замерен в два этапа:

* Первый этап: замер малых расстояний от 0 см до 150 см
* Второй этап: замер длинных расстояний от 120 см до 2500 см

Первый этап дал следующие результаты.

Данные при расстоянии 0 см:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Номер** | **Расстояние (см)** | **Уровень сигнала (Db)** | **Среднее** | **Погрешность** |
| 1 | 0 | -20 | -19,24 | 4% |
| 2 | 0 | -19 | -19,24 | -1% |
| 3 | 0 | -20 | -19,24 | 4% |
| 4 | 0 | -18 | -19,24 | -7% |
| 5 | 0 | -18 | -19,24 | -7% |
| 6 | 0 | -19 | -19,24 | -1% |
| 7 | 0 | -17 | -19,24 | -13% |
| 8 | 0 | -20 | -19,24 | 4% |
| 9 | 0 | -20 | -19,24 | 4% |
| 10 | 0 | -20 | -19,24 | 4% |
| 11 | 0 | -19 | -19,24 | -1% |
| 12 | 0 | -20 | -19,24 | 4% |
| 13 | 0 | -20 | -19,24 | 4% |
| 14 | 0 | -19 | -19,24 | -1% |
| 15 | 0 | -18 | -19,24 | -7% |
| 16 | 0 | -19 | -19,24 | -1% |
| 17 | 0 | -20 | -19,24 | 4% |
| 18 | 0 | -18 | -19,24 | -7% |
| 19 | 0 | -20 | -19,24 | 4% |
| 20 | 0 | -20 | -19,24 | 4% |
| 21 | 0 | -20 | -19,24 | 4% |

Среднее значение сигнала при 0 см: -19 Db



Рисунок 2 - 0 см

Данные при расстоянии 10 см:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Номер** | **Расстояние (см)** | **Уровень сигнала (Db)** | **Среднее** | **Погрешность** |
| 1 | 10 | - 45,00 | - 47,05 | -5% |
| 2 | 10 | - 46,00 | - 47,05 | -2% |
| 3 | 10 | - 48,00 | - 47,05 | 2% |
| 4 | 10 | - 51,00 | - 47,05 | 8% |
| 5 | 10 | - 49,00 | - 47,05 | 4% |
| 6 | 10 | - 47,00 | - 47,05 | 0% |
| 7 | 10 | - 45,00 | - 47,05 | -5% |
| 8 | 10 | - 47,00 | - 47,05 | 0% |
| 9 | 10 | - 47,00 | - 47,05 | 0% |
| 10 | 10 | - 46,00 | - 47,05 | -2% |
| 11 | 10 | - 49,00 | - 47,05 | 4% |
| 12 | 10 | - 42,00 | - 47,05 | -12% |
| 13 | 10 | - 44,00 | - 47,05 | -7% |
| 14 | 10 | - 47,00 | - 47,05 | 0% |
| 15 | 10 | - 47,00 | - 47,05 | 0% |
| 16 | 10 | - 47,00 | - 47,05 | 0% |
| 17 | 10 | - 47,00 | - 47,05 | 0% |
| 18 | 10 | - 46,00 | - 47,05 | -2% |
| 19 | 10 | - 50,00 | - 47,05 | 6% |
| 20 | 10 | - 49,00 | - 47,05 | 4% |
| 21 | 10 | - 49,00 | - 47,05 | 4% |

Средний уровень сигнала при 20 см: -47 Db



Рисунок 3 - 10 см

Данные при расстоянии 20 см:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Номер** | **Расстояние (см)** | **Уровень сигнала (Db)** | **Среднее** | **Погрешность** |
| 1 | 20 | - 57,00 | - 57,38 | -1% |
| 2 | 20 | - 55,00 | - 57,38 | -4% |
| 3 | 20 | - 60,00 | - 57,38 | 4% |
| 4 | 20 | - 59,00 | - 57,38 | 3% |
| 5 | 20 | - 63,00 | - 57,38 | 9% |
| 6 | 20 | - 49,00 | - 57,38 | -17% |
| 7 | 20 | - 55,00 | - 57,38 | -4% |
| 8 | 20 | - 58,00 | - 57,38 | 1% |
| 9 | 20 | - 58,00 | - 57,38 | 1% |
| 10 | 20 | - 60,00 | - 57,38 | 4% |
| 11 | 20 | - 59,00 | - 57,38 | 3% |
| 12 | 20 | - 58,00 | - 57,38 | 1% |
| 13 | 20 | - 55,00 | - 57,38 | -4% |
| 14 | 20 | - 59,00 | - 57,38 | 3% |
| 15 | 20 | - 62,00 | - 57,38 | 7% |
| 16 | 20 | - 59,00 | - 57,38 | 3% |
| 17 | 20 | - 58,00 | - 57,38 | 1% |
| 18 | 20 | - 57,00 | - 57,38 | -1% |
| 19 | 20 | - 62,00 | - 57,38 | 7% |
| 20 | 20 | - 50,00 | - 57,38 | -15% |
| 21 | 20 | - 52,00 | - 57,38 | -10% |

Среднее значение при 20 см: -57 Db

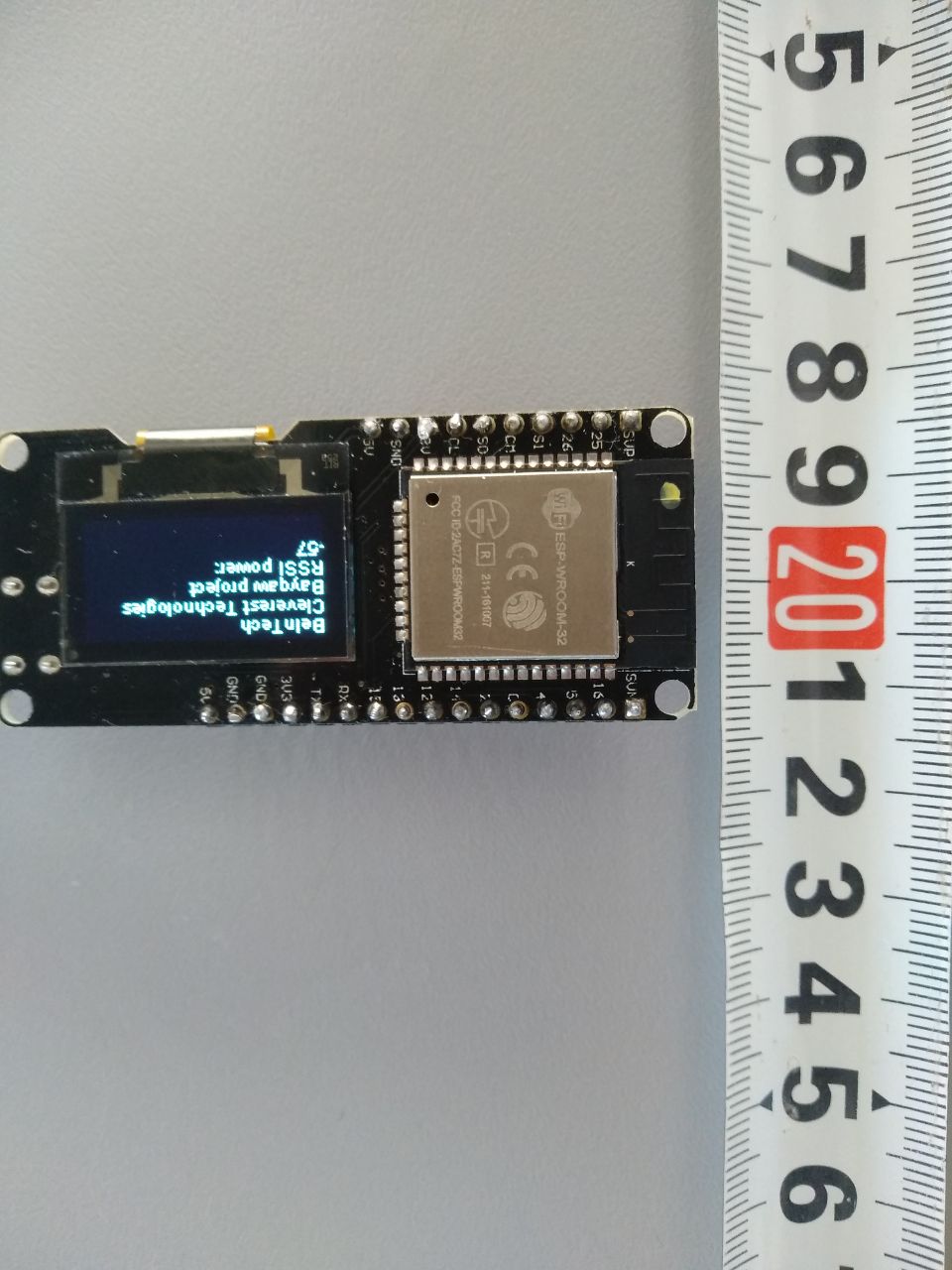


Рисунок 4 - 20 см

Данные при расстоянии 30 см:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Номер** | **Расстояние (см)** | **Уровень сигнала (Db)** | **Среднее** | **Погрешность** |
| 1 | 30 | - 54,00 | - 52,33 | 3% |
| 2 | 30 | - 55,00 | - 52,33 | 5% |
| 3 | 30 | - 50,00 | - 52,33 | -5% |
| 4 | 30 | - 52,00 | - 52,33 | -1% |
| 5 | 30 | - 50,00 | - 52,33 | -5% |
| 6 | 30 | - 55,00 | - 52,33 | 5% |
| 7 | 30 | - 60,00 | - 52,33 | 13% |
| 8 | 30 | - 49,00 | - 52,33 | -7% |
| 9 | 30 | - 55,00 | - 52,33 | 5% |
| 10 | 30 | - 52,00 | - 52,33 | -1% |
| 11 | 30 | - 51,00 | - 52,33 | -3% |
| 12 | 30 | - 50,00 | - 52,33 | -5% |
| 13 | 30 | - 51,00 | - 52,33 | -3% |
| 14 | 30 | - 55,00 | - 52,33 | 5% |
| 15 | 30 | - 48,00 | - 52,33 | -9% |
| 16 | 30 | - 52,00 | - 52,33 | -1% |
| 17 | 30 | - 55,00 | - 52,33 | 5% |
| 18 | 30 | - 52,00 | - 52,33 | -1% |
| 19 | 30 | - 50,00 | - 52,33 | -5% |
| 20 | 30 | - 51,00 | - 52,33 | -3% |
| 21 | 30 | - 52,00 | - 52,33 | -1% |

Среднее значение при 30 см: -52 Db



Рисунок 5 - 30 см

Данные при расстоянии 40 см:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Номер** | **Расстояние (см)** | **Уровень сигнала (Db)** | **Среднее** | **Погрешность** |
| 1 | 40 | - 64,00 | - 74,19 | -16% |
| 2 | 40 | - 70,00 | - 74,19 | -6% |
| 3 | 40 | - 69,00 | - 74,19 | -8% |
| 4 | 40 | - 78,00 | - 74,19 | 5% |
| 5 | 40 | - 77,00 | - 74,19 | 4% |
| 6 | 40 | - 75,00 | - 74,19 | 1% |
| 7 | 40 | - 75,00 | - 74,19 | 1% |
| 8 | 40 | - 73,00 | - 74,19 | -2% |
| 9 | 40 | - 77,00 | - 74,19 | 4% |
| 10 | 40 | - 72,00 | - 74,19 | -3% |
| 11 | 40 | - 70,00 | - 74,19 | -6% |
| 12 | 40 | - 79,00 | - 74,19 | 6% |
| 13 | 40 | - 78,00 | - 74,19 | 5% |
| 14 | 40 | - 75,00 | - 74,19 | 1% |
| 15 | 40 | - 76,00 | - 74,19 | 2% |
| 16 | 40 | - 75,00 | - 74,19 | 1% |
| 17 | 40 | - 75,00 | - 74,19 | 1% |
| 18 | 40 | - 79,00 | - 74,19 | 6% |
| 19 | 40 | - 74,00 | - 74,19 | 0% |
| 20 | 40 | - 70,00 | - 74,19 | -6% |
| 21 | 40 | - 77,00 | - 74,19 | 4% |

Среднее значение при 40 см: -74 Db



Рисунок 6 - при 40 см

Данные при расстоянии 50 см:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Номер** | **Расстояние (см)** | **Уровень сигнала (Db)** | **Среднее** | **Погрешность** |
| 1 | 50 | - 62,00 | - 70,62 | -14% |
| 2 | 50 | - 69,00 | - 70,62 | -2% |
| 3 | 50 | - 77,00 | - 70,62 | 8% |
| 4 | 50 | - 60,00 | - 70,62 | -18% |
| 5 | 50 | - 62,00 | - 70,62 | -14% |
| 6 | 50 | - 75,00 | - 70,62 | 6% |
| 7 | 50 | - 75,00 | - 70,62 | 6% |
| 8 | 50 | - 73,00 | - 70,62 | 3% |
| 9 | 50 | - 77,00 | - 70,62 | 8% |
| 10 | 50 | - 72,00 | - 70,62 | 2% |
| 11 | 50 | - 70,00 | - 70,62 | -1% |
| 12 | 50 | - 60,00 | - 70,62 | -18% |
| 13 | 50 | - 78,00 | - 70,62 | 9% |
| 14 | 50 | - 61,00 | - 70,62 | -16% |
| 15 | 50 | - 76,00 | - 70,62 | 7% |
| 16 | 50 | - 77,00 | - 70,62 | 8% |
| 17 | 50 | - 75,00 | - 70,62 | 6% |
| 18 | 50 | - 69,00 | - 70,62 | -2% |
| 19 | 50 | - 74,00 | - 70,62 | 5% |
| 20 | 50 | - 70,00 | - 70,62 | -1% |
| 21 | 50 | - 71,00 | - 70,62 | 1% |

Среднее значение: -70 Db

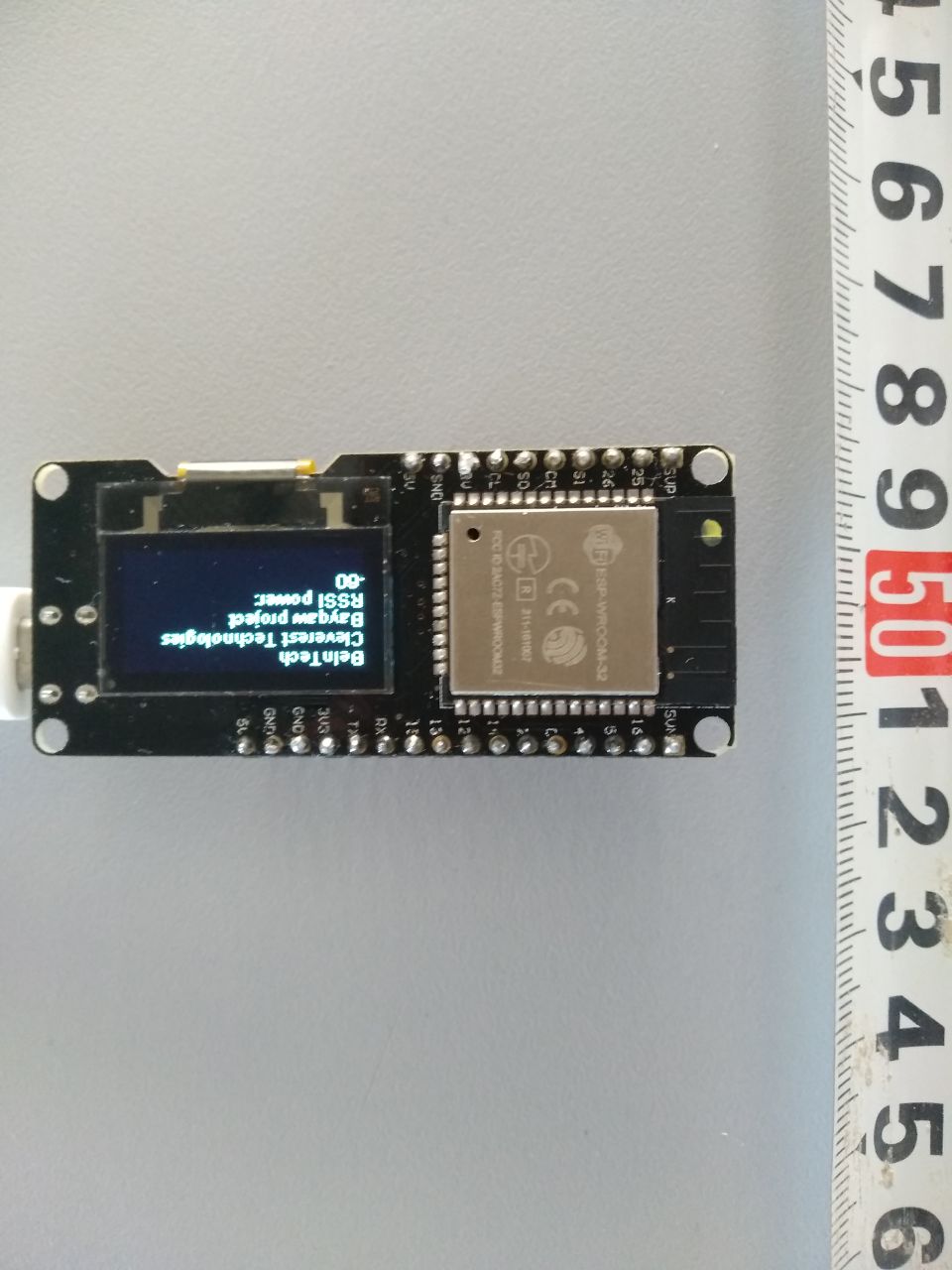


Рисунок 7 - при 50 см

Данные при расстоянии 100 см:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Номер** | **Расстояние (см)** | **Уровень сигнала (Db)** | **Среднее** | **Погрешность** |
| 1 | 100 | - 75,00 | - 72,43 | 3% |
| 2 | 100 | - 77,00 | - 72,43 | 6% |
| 3 | 100 | - 79,00 | - 72,43 | 8% |
| 4 | 100 | - 72,00 | - 72,43 | -1% |
| 5 | 100 | - 71,00 | - 72,43 | -2% |
| 6 | 100 | - 70,00 | - 72,43 | -3% |
| 7 | 100 | - 71,00 | - 72,43 | -2% |
| 8 | 100 | - 79,00 | - 72,43 | 8% |
| 9 | 100 | - 77,00 | - 72,43 | 6% |
| 10 | 100 | - 72,00 | - 72,43 | -1% |
| 11 | 100 | - 70,00 | - 72,43 | -3% |
| 12 | 100 | - 69,00 | - 72,43 | -5% |
| 13 | 100 | - 74,00 | - 72,43 | 2% |
| 14 | 100 | - 73,00 | - 72,43 | 1% |
| 15 | 100 | - 64,00 | - 72,43 | -13% |
| 16 | 100 | - 75,00 | - 72,43 | 3% |
| 17 | 100 | - 62,00 | - 72,43 | -17% |
| 18 | 100 | - 69,00 | - 72,43 | -5% |
| 19 | 100 | - 74,00 | - 72,43 | 2% |
| 20 | 100 | - 77,00 | - 72,43 | 6% |
| 21 | 100 | - 71,00 | - 72,43 | -2% |

Среднее значение: 72 Db

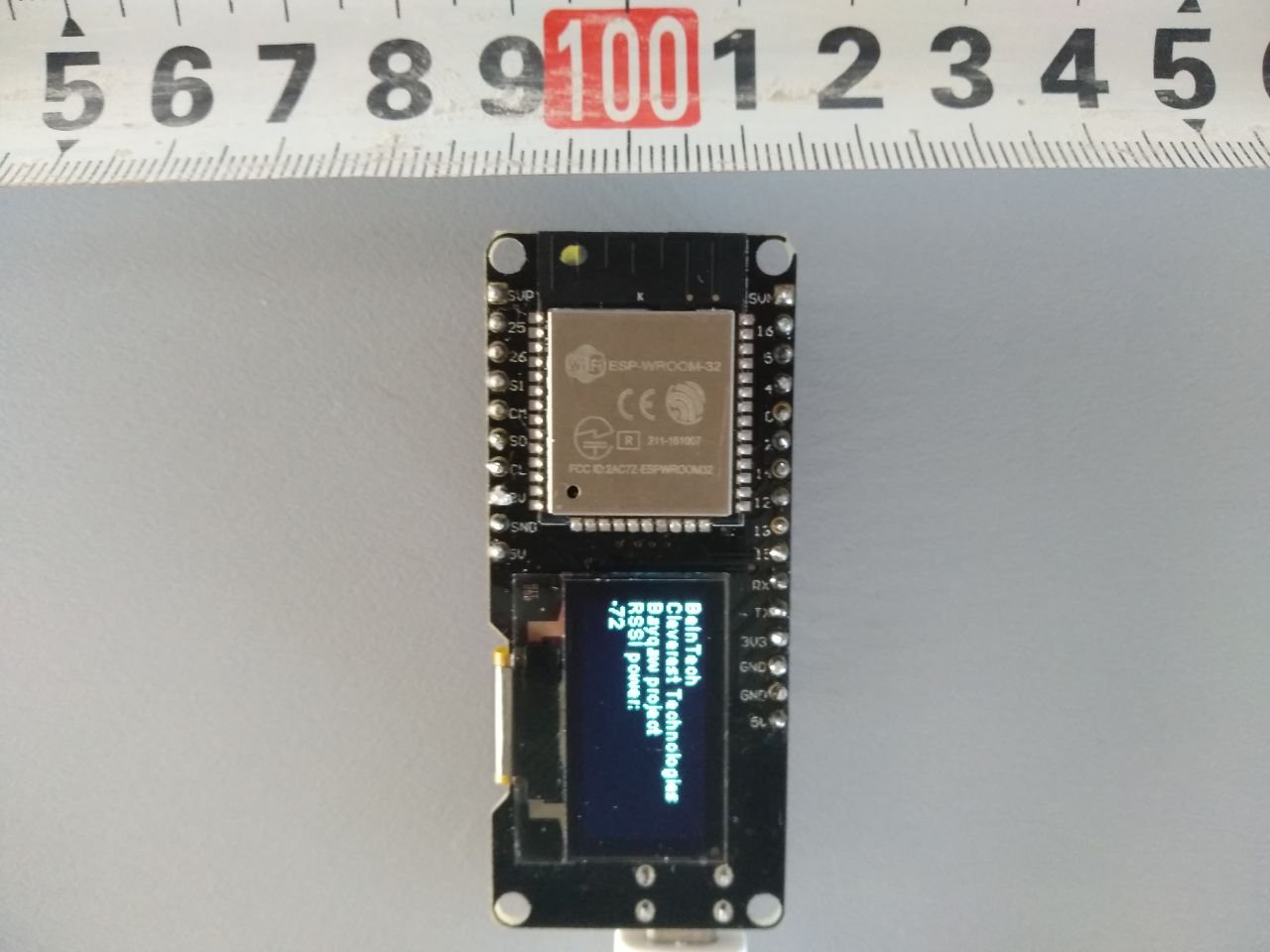


Рисунок 8 - при 100 см

Второй этап дал следующие результаты.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Номер** | **Расстояние (см)** | **Уровень сигнала (Db)** | **Среднее** | **Погрешность** |
| 1 | 120 | - 80,00 | - 80,67 | -1% |
| 2 | 120 | - 81,00 | - 80,67 | 0% |
| 3 | 120 | - 81,00 | - 80,67 | 0% |
| 4 | 240 | - 84,00 | - 82,67 | 2% |
| 5 | 240 | - 82,00 | - 82,67 | -1% |
| 6 | 240 | - 82,00 | - 82,67 | -1% |
| 7 | 360 | - 89,00 | - 87,33 | 2% |
| 8 | 360 | - 88,00 | - 87,33 | 1% |
| 9 | 360 | - 85,00 | - 87,33 | -3% |
| 10 | 480 | - 88,00 | - 88,33 | 0% |
| 11 | 480 | - 89,00 | - 88,33 | 1% |
| 12 | 480 | - 88,00 | - 88,33 | 0% |
| 13 | 600 | - 90,00 | - 90,33 | 0% |
| 14 | 600 | - 89,00 | - 90,33 | -1% |
| 15 | 600 | - 92,00 | - 90,33 | 2% |

Максимальная длина сигнала: 23 метра

**Заключение**

Рисунок 9 - Уровень сигнала

Выводы:

* У уровня сигнала есть паттерны увеличения на определенных отрезках
* Сильное зашумление не дает возможности отслеживать сигнал со 100% вероятностью

# Сигнал через двери

Был проведен эксперимент для определения уровня сигнала через стандартные офисные двери из ДСП или ЛДСП.

Замеры для одной двери:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Номер** | **Количество дверей** | **Сигнал** | **Средний уровень** | **Погрешность** |
| 1 | 1 | - 89,00 | - 88,33 | 1% |
| 2 | 1 | - 90,00 | - 88,33 | 2% |
| 3 | 1 | - 88,00 | - 88,33 | 0% |
| 4 | 1 | - 88,00 | - 88,33 | 0% |
| 5 | 1 | - 89,00 | - 88,33 | 1% |
| 6 | 1 | - 89,00 | - 88,33 | 1% |
| 7 | 1 | - 85,00 | - 88,33 | -4% |
| 8 | 1 | - 89,00 | - 88,33 | 1% |
| 9 | 1 | - 88,00 | - 88,33 | 0% |

Через одну дверь сигнал теряет около 40% своего уровня.

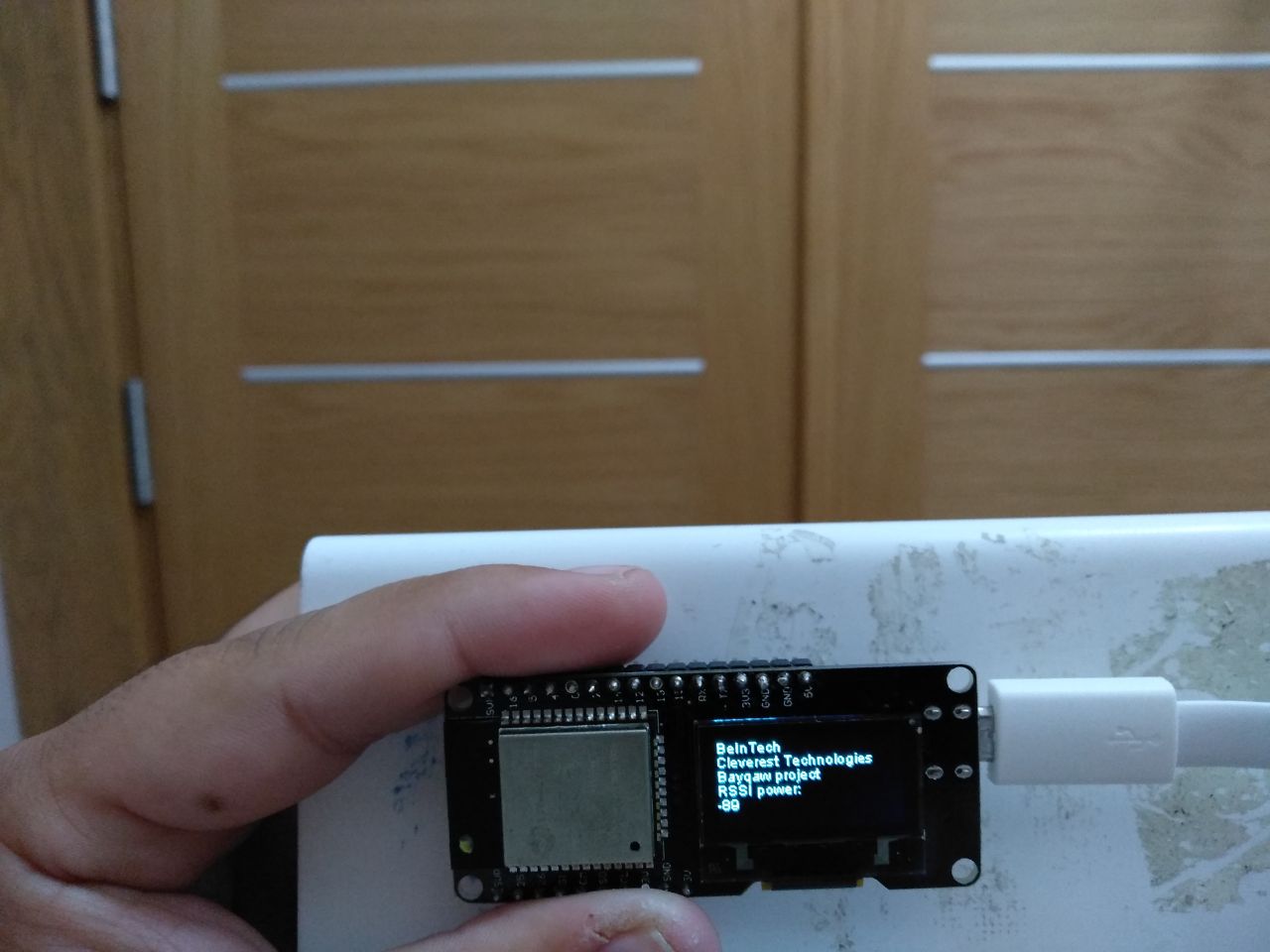


Рисунок 10 - Сигнал через одну дверь

Замеры через две двери:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Номер** | **Количество дверей** | **Сигнал** | **Средний уровень** | **Погрешность** |
| 1 | 2 | - 97,00 | - 96,67 | 0% |
| 2 | 2 | - 96,00 | - 96,63 | -1% |
| 3 | 2 | - 96,00 | - 96,71 | -1% |
| 4 | 2 | - 100,00\* | - 96,83 | 3% |
| 5 | 2 | - 95,00 | - 96,20 | -1% |
| 6 | 2 | - 96,00 | - 96,50 | -1% |
| 7 | 2 | - 100,00 | - 96,67 | 3% |
| 8 | 2 | - 96,00 | - 95,00 | 1% |
| 9 | 2 | - 94,00 | - 94,00 | 0% |

Значение 100 обозначает потерю сигнала



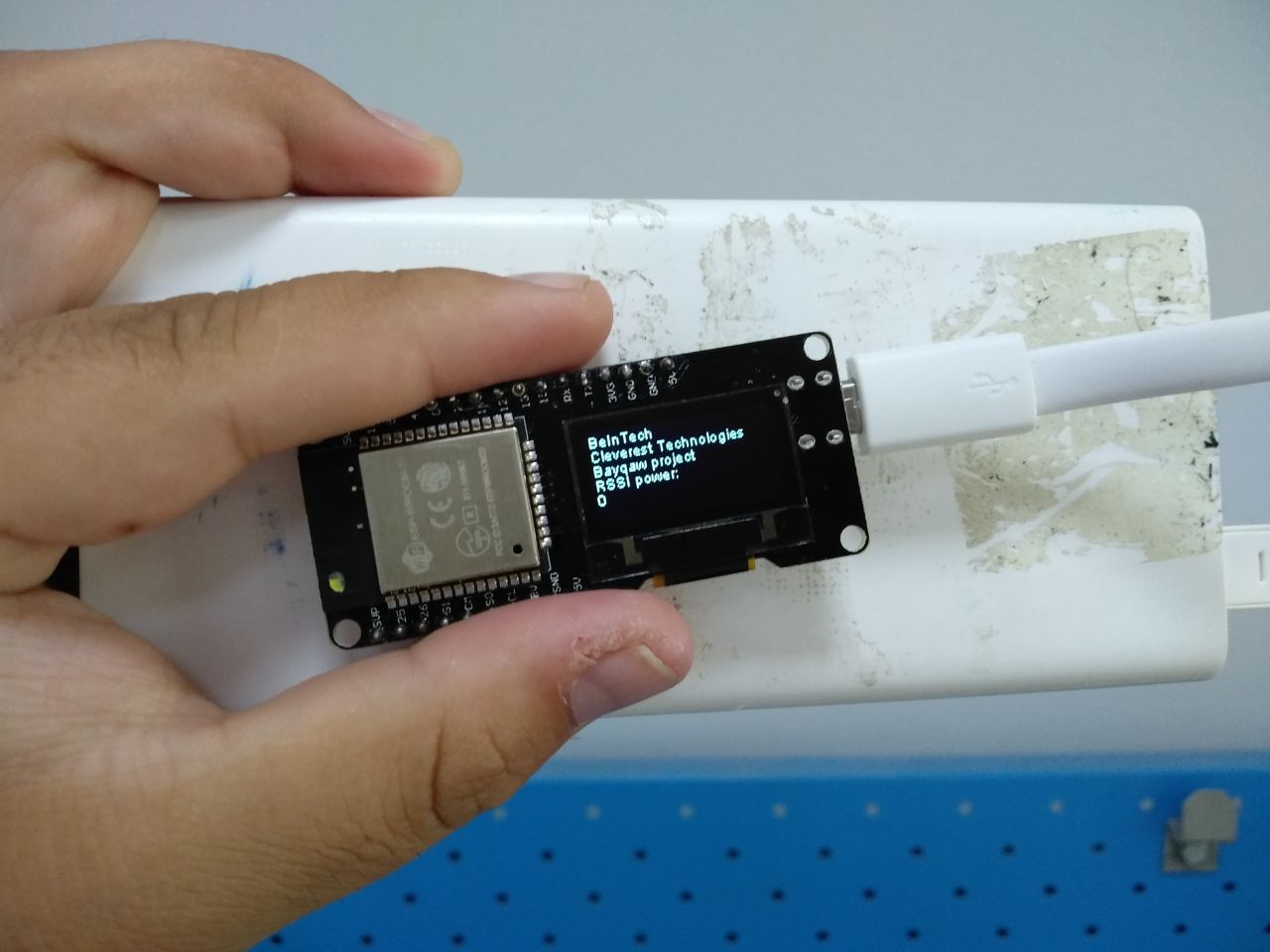
Рисунок 11 - Значение сигнала через две двери

**Заключение**

Сигнал BLE может пройти не более чем через две офисные двери. Рекомендуется не использовать более одной двери, так как временами сигнал теряется.

# Сигнал через стену

BLE сигнал не может пройти через бетонную стену. Не рекомендуется использовать BLE сигналы для коммуникации через бетонные стены.



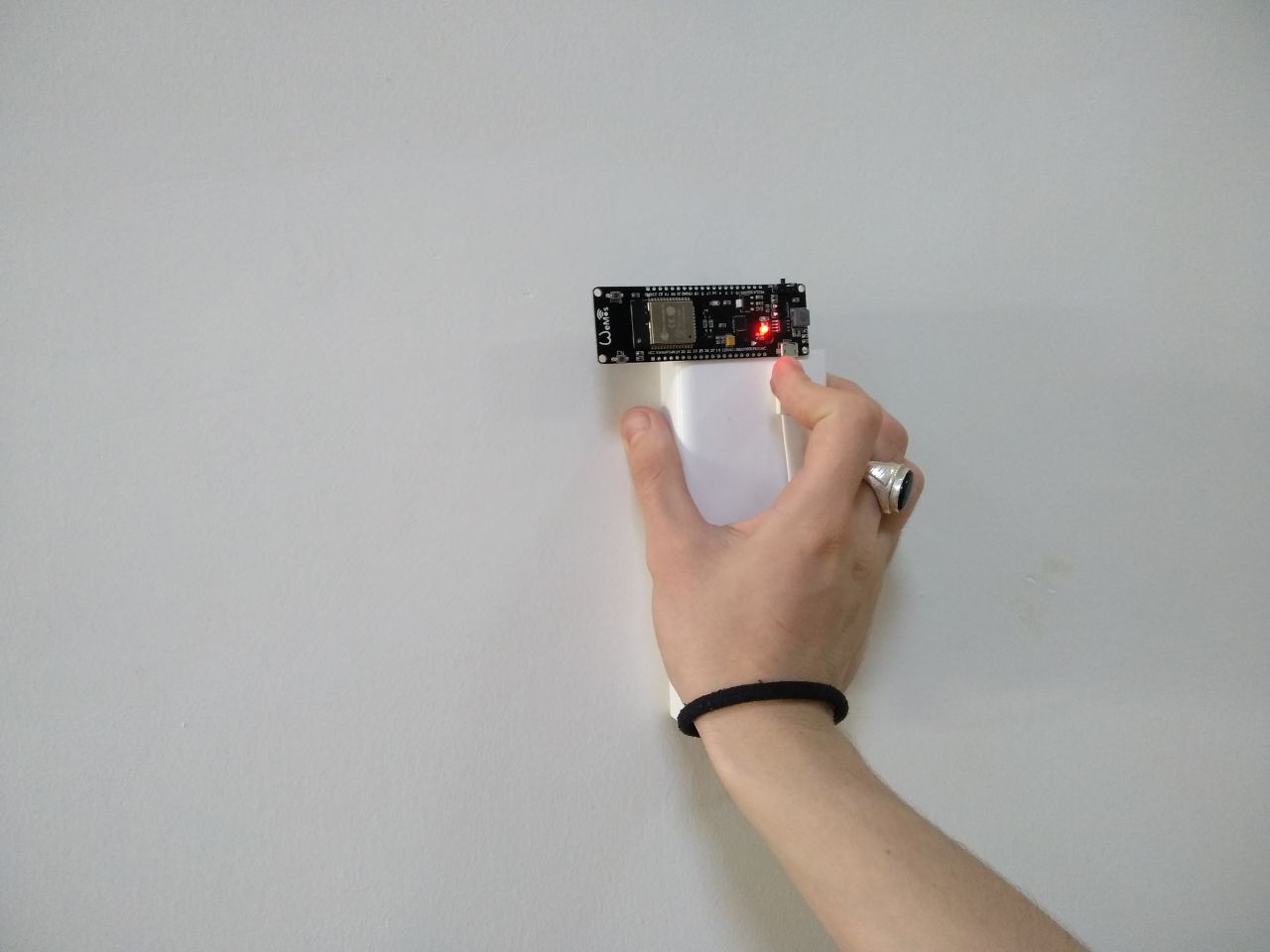


Рисунок 12 - Сигнал через бетонные стены

# Сигнал через стекло

Сигнал через стекло ведет себя аналогично сигналу через офисные двери.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Номер** | **Количество окон** | **Сигнал** | **Средний уровень** | **Погрешность** |
| 1 | 1 | - 72,00 | - 70,33 | 2% |
| 2 | 1 | - 71,00 | - 70,13 | 1% |
| 3 | 1 | - 72,00 | - 70,00 | 3% |
| 4 | 1 | - 70,00 | - 69,67 | 0% |
| 5 | 1 | - 69,00 | - 69,60 | -1% |
| 6 | 1 | - 70,00 | - 69,75 | 0% |
| 7 | 1 | - 70,00 | - 69,67 | 0% |
| 8 | 1 | - 69,00 | - 69,50 | -1% |
| 9 | 1 | - 70,00 | - 70,00 | 0% |

В среднем теряется 50% сигнала.

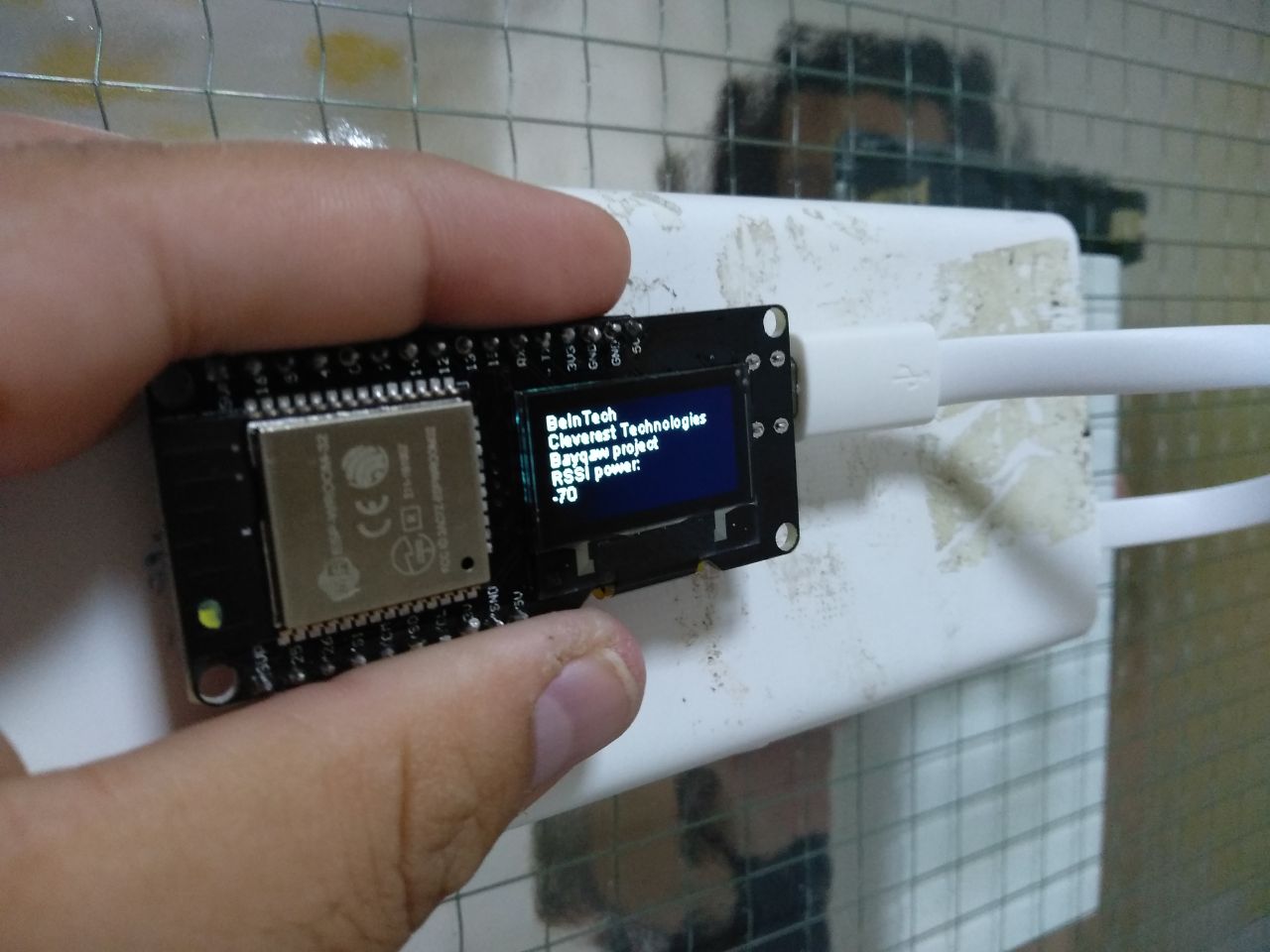


Рисунок 13 - Через стекло

**Заключение**

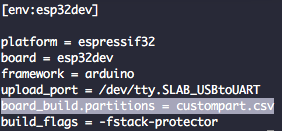
Не рекомендуется использовать.

# Изменение таблицы разделов

При компиляции кода стек ПО отвечающий за работу BLE может занимать до 90% flash памяти, стандартно отведенной для исходного кода в микроконтроллерах ESP32. По этой причине необходимо увеличение раздела для загрузки путем изменения таблицы разделов в ESP32.

Данная процедура в среде PlatformIO делается путем создания кастомной таблицы разделов в папке проекта и изменением файла загрузки в папке проекта.

Данные манипуляции проводятся этим алгоритмом:

* Необходимо создать новую таблицу разделов
  + /project/custompart.csv
  + 
  + Как видно, мы отдали 3 мб под загрузку, вместо стандартных 1-1,5 мб
* В файле настроек загрузки необходимо указать файл с таблицей разделов
  + /project/platformio.ini
  + Необходимо добавить строчку: board\_build.partitions = custompart.csv
  + 

После данных манипуляций, написанное ПО будет вмещаться в память устройства.

# Mesh Network

На данный момент ведутся работы по созданию BLE Mesh Network и их можно охарактеризовать следующим образом:

* **Рекламные сети:** сети, которые используют только GAP серверы для передачи своей информации, работу ведет компания SIG.
* **Научные работы:** на данный момент существует ряд работ в научных кругах, где ведется поиск решений для Mesh Network

Для создания Mesh Network нашей компанией был разработан собственный алгоритм передачи иерархии и информации.

# Алгоритм "BPDN" - BLE of Parent and Daughter Nodes

Основные свойства алгоритма:

* **Робототехника «Роя»:** в основу системы были взяты принципы «Swarm» Robotics, то есть:
  + Каждый элемент сети должен быть одинаковым
  + Каждый элемент сети должен быть самоопределяющимся внутри сети
  + Сеть должна само организовываться без участия оператора
* **Принцип дерева с ветками:** каждый элемент сети должен подключаться к самому старшему по иерархии элементу сети, чтобы распределять нагрузку на сеть
* **Принцип глубины:** сеть должна иметь возможность масштабироваться в глубину и в ширину

**Идентификация ячеек:**

Стандартный UUID ячейки выглядит так:

F0000000-B5C3-F393-E0A9-E50E24DCCA9E

0хF – определение приоритета, где F – ячейка родитель (ячейка, которая подключается к интернету)

1х0 – определение очереди, если на уровне приоритета ячейки есть другие ячейки, ячейка выбирает следующий свободный лот, то есть число 8 на втором байте обозначает, что в данном приоритете уже есть 7 других ячеек

8-11хB5С3 – определяет, что ячейка является частью определенного Mesh Network-а

35хE – определяет назначение характеристики, 1 – значит характеристика для родителей ячейки, 2 – значит характеристика для дочерей

**Алгоритм:**

Основа алгоритма в использовании GAP сервера для передачи данных приоритета в коде UUID.

Рисунок 14 - Топология

Как видно с топологии, основным HEX байтом для сети является нулевой байт, определяющий приоритет, за ним идет первый байт, который определяет количество ячеек на данном уровне.

Ячейки клиенты подключаются к серверам и передают в характеристики ячеек серверов данные в виде String строки. Данная строка позволяет вводить любую по форме и типу информацию.

**Положительные моменты:**

* Каждый нод сети является одним устройством
* Каждый нод поддерживает общение по BLE и по Wi-Fi
* Возможность масштабировать сеть другими девайсами, например фитнесс браслеты

**Отрицательные моменты:**

* Скорость передачи файлов может занимать много времени из-за синхронной природы явления
* При работе в двух режимах потребляется много энергии

# Практика Mesh Network

Для проведения тестов и экспериментов была разработана система Mesh Network на основе протокола «BPDN». Исходный код находится в папке «MeshNetwork» по пути: /bayqaw/PlaformIO/Report#1/MeshNetwork в репозиторий.

Рисунок 15 - Алгоритм работы ноды

# Определение приоритета

Функция «void define\_priority()» определяет приоритет существующего нода. Ячейка ищет ячейки рядом с собой, отбирает те, что являются нодами ее сети и присваивает себе приоритет.

Рисунок 16 - Алгоритм "Define Priority"

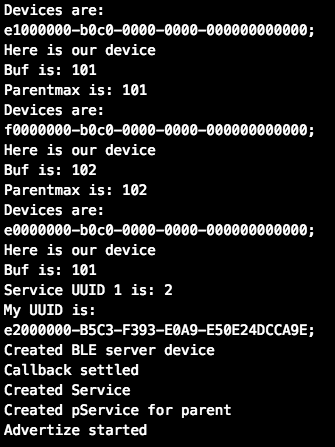


Рисунок 17 - Пример работы алгоритма

* Нод нашел три девайса с характеристикой нашей сети
* Он подключается к самому старшему (UUID: F000…)
* Он становится UUID: E200…
* Он получает вторым номером 2, потому что уже существуют 2 нода в приоритете Е
* После этого нод запускает 5-минутное вещание, чтобы остальные новые ноды могли получить приоритет

# Прием информации

Прием информации нод ведет в роли сервера, нод запускает сервер и клиенты дочки и родители подключаются к нему и вносят свою информацию. Сбор информации ведется внутри функции «server» и запускается на 20 секунд.

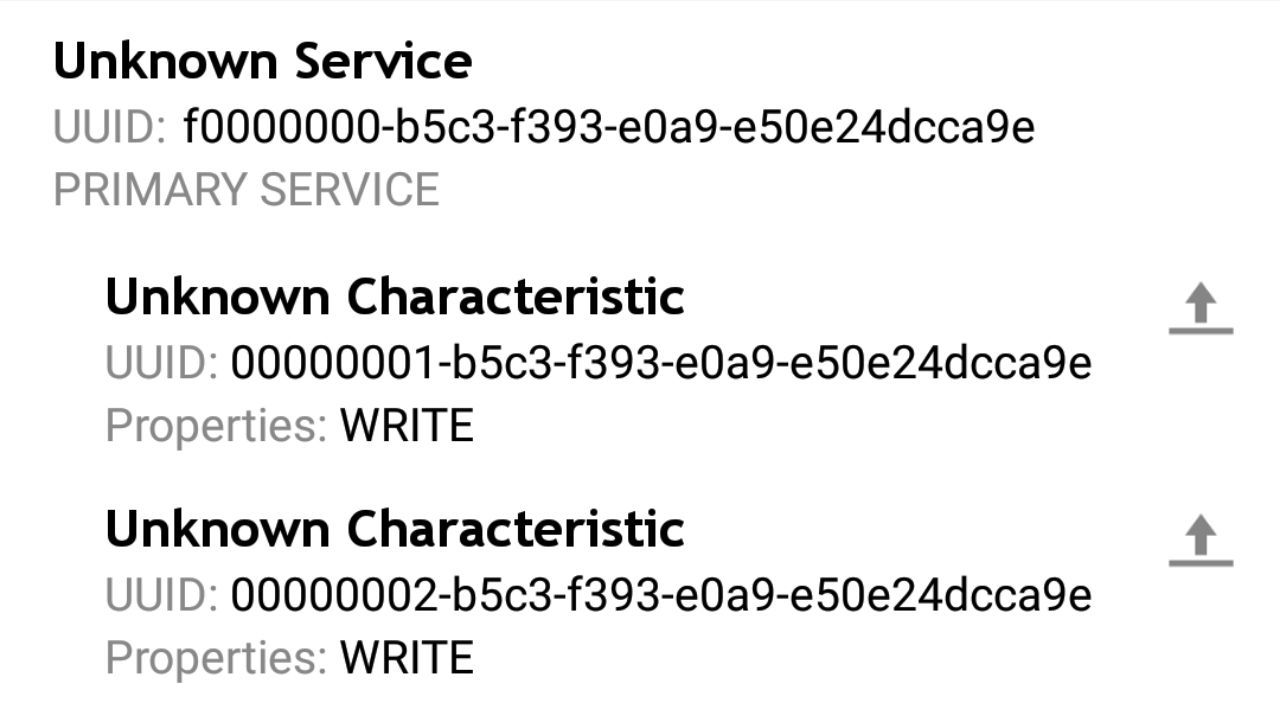


Рисунок 18 - Характеристики сервера для приема информации

Нод запускает две характеристики:

* charUUID\_P("00000001-B5C3-F393-E0A9-E50E24DCCA9E");
  + Характеристика для родительного нода
* charUUID\_D("00000002-B5C3-F393-E0A9-E50E24DCCA9E");
  + Характеристика для дочернего нода

Примечание: родители ноды также получают информацию от HTTP сервера по Wi-Fi с помощью POST запроса.

Рисунок 19 – Алгоритм приема информации



Рисунок 20 - Пример приема информации

На примере выше видно, что наш сервер имеет UUID: e0000… так как в сети уже есть нода с приоритетом F0000…

Далее он принимает сообщение от родителя и от дочки.

# Передача информации

Нода включает режим клиента и ищет другие ноды, которые являются его родителем или дочерью. Далее нод подключается к родителям или дочерям в виде клиента и записывает свои данные в характеристики. За данный функционал отвечает функция «client».

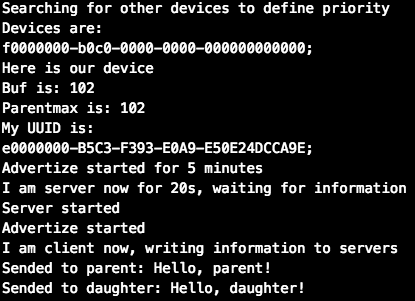


Рисунок 21 - Пример отправки информации

В примере выше, нод отправляет два сообщения родителю и дочери.

# Прием информации от HTTP сервера

Если нода является родительской, то она подключается к Wi-Fi и получает/отправляет информацию на HTTP сервер расположенный по IP адресу в памяти. За данный функционал отвечает функция «void TakeFromInternet()».



Рисунок 22 - Пример работы с HTTP сервером

На примере выше, нода становится родительской и после этого подключается к HTTP серверу и получает информацию.

# Система "BNC" - BLE Nodes Calculator

BNC – система по подсчету BLE beacon-ов определенной характеристики по данным:

* UUID
* Имя ноды
* Производитель

Данный программное обеспечение находится по пути: bayqaw/PlatformIO/Report#1/NodesCalculation

**Алгоритм работы:**

Рисунок 23 - Алгоритм BNC

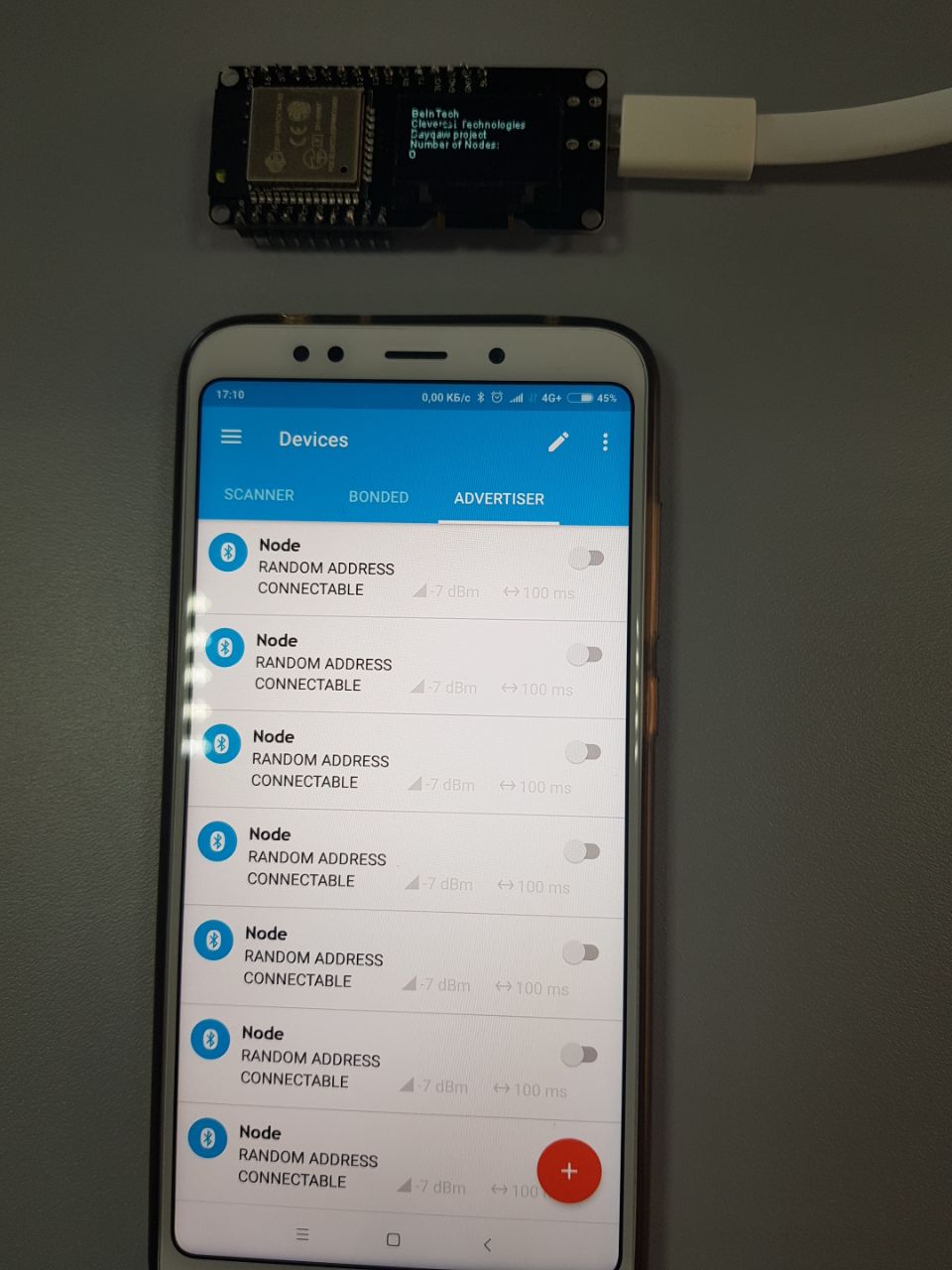


Рисунок 24 - BNC показывает 0

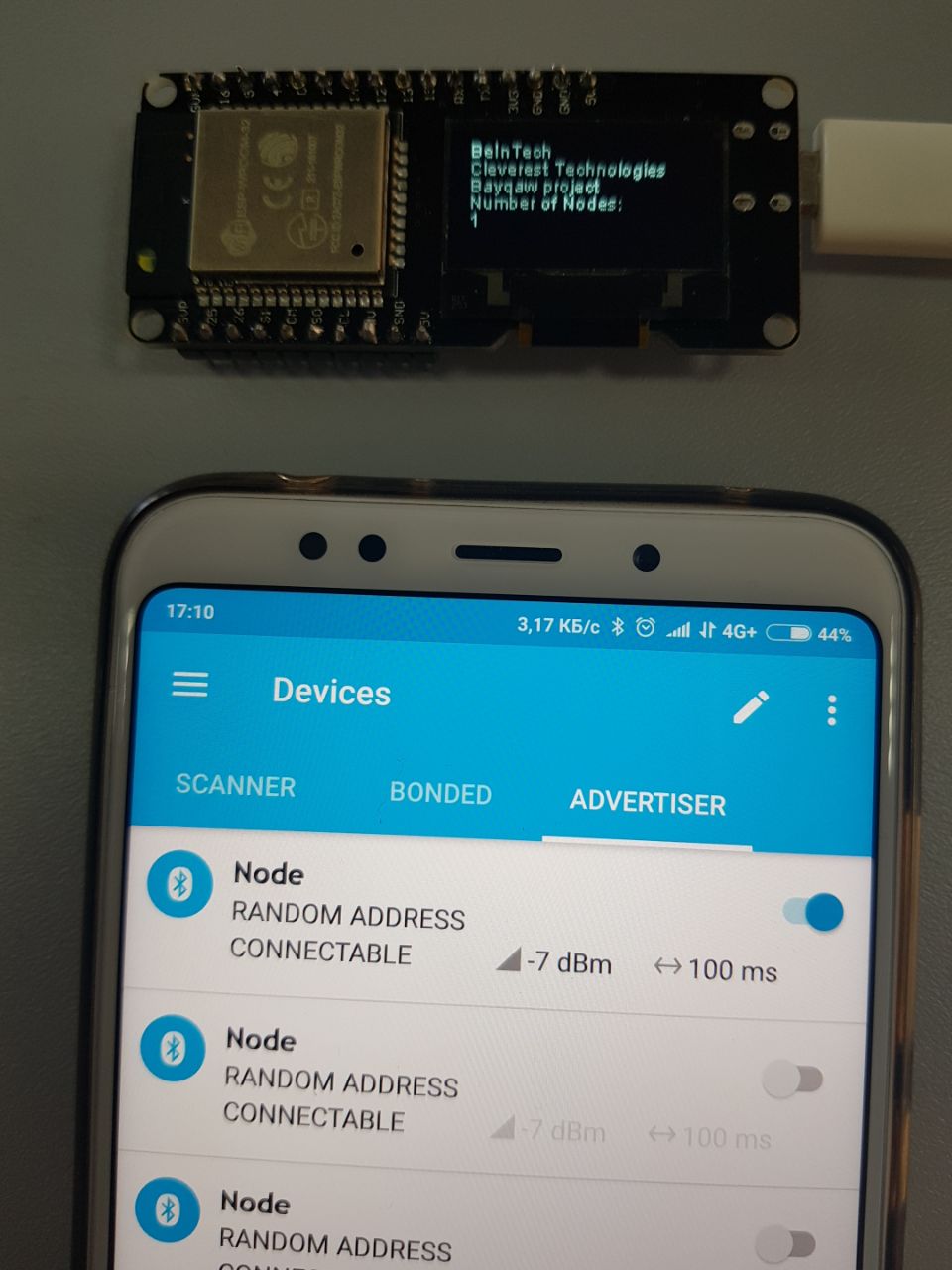


Рисунок 25 - BNC показывает 1

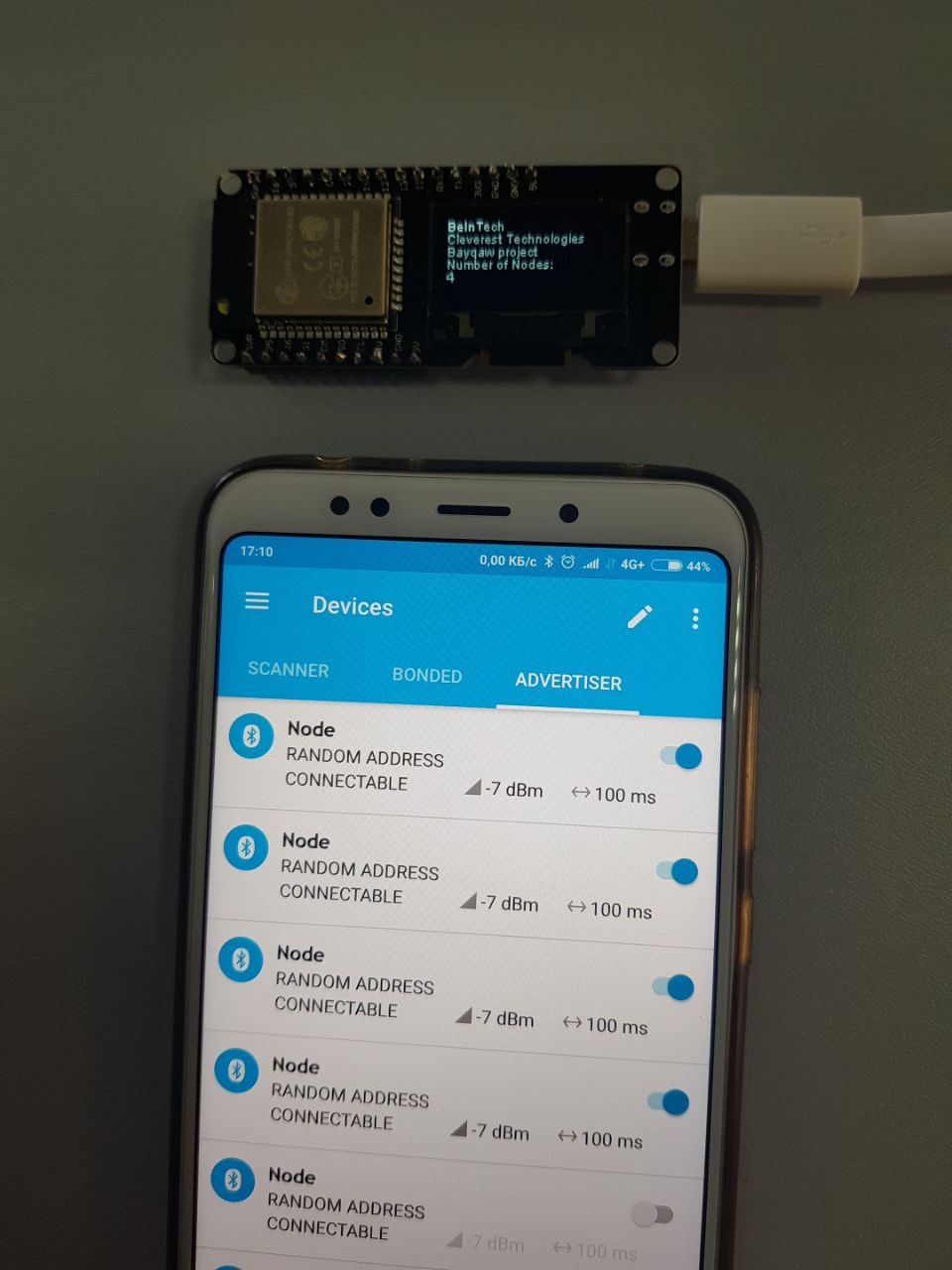


Рисунок 26 - BNC показывает 4

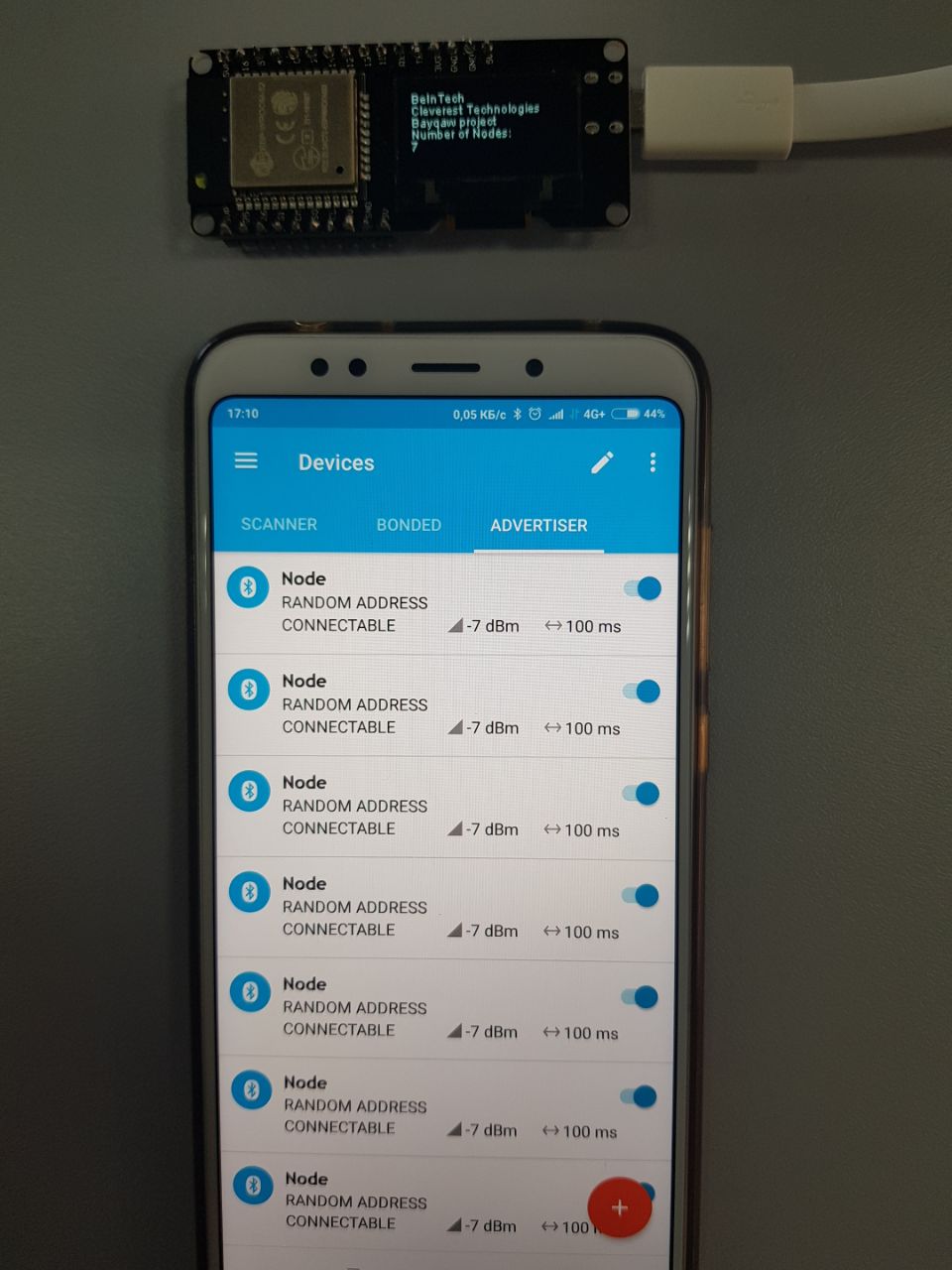


Рисунок 27 - BNC показывает 7

# Заключение и рекомендации

Основной целью данной научно-исследовательской и конструкторской работы было проверка нескольких гипотез:

* Работоспособность ESP32 как BLE устройства:
  + Проверка уровней сигнала
  + Проверка работоспособности кода на C++
* Проверка гипотезы Mesh Network на основе BLE:
  + Проверка и разработка системы присвоения иерархии
  + Создание самоорганизующейся сети
  + Создание прототипа Mesh Network
  + Прием и передача данных
  + Слияние Wi-Fi и BLE внутри одного устройства
* Проверка гипотезы BNC:
  + Проверка гипотезы
  + Разработка устройство

Дополнительные исследования были проведены для дальнейшей работы:

* Увеличение памяти для скетчей:
  + Была проведена работа по изменению таблицы разделов ESP32

Статус НИОКР:

* Неподтверждённые гипотезы: отсутствуют
* Статус НИОКР: положительный
* Готовность к переходу на следующий этап: ожидание доставки компонентов из Китая
* Сложность этапа: 7 из 10

Ряд рекомендаций для дальнейшей работы:

* Увеличение скорости передачи:
  + Для увеличение скорости передачи данных рекомендуется использовать двойное устройство ESP32, где один элемент будет работать как приемник, другой как передатчик.
* Улучшение энергопотребления:
  + Рекомендуется использовать одноплатные компьютеры как ноды родители, так как сами ESP32 очень сильно греются при работе одновременно с BLE и Wi-Fi.

# Список изображений

[Рисунок 1 - Основные топологии сетей 9](#_Toc523163581)

[Рисунок 2 - 0 см 11](#_Toc523163582)

[Рисунок 3 - 10 см 12](#_Toc523163583)

[Рисунок 4 - 20 см 13](#_Toc523163584)

[Рисунок 5 - 30 см 14](#_Toc523163585)

[Рисунок 6 - при 40 см 15](#_Toc523163586)

[Рисунок 7 - при 50 см 16](#_Toc523163587)

[Рисунок 8 - при 100 см 17](#_Toc523163588)

[Рисунок 9 - Уровень сигнала 18](#_Toc523163589)

[Рисунок 10 - Сигнал через одну дверь 19](#_Toc523163590)

[Рисунок 11 - Значение сигнала через две двери 20](#_Toc523163591)

[Рисунок 12 - Сигнал через бетонные стены 21](#_Toc523163592)

[Рисунок 13 - Через стекло 22](#_Toc523163593)

[Рисунок 14 - Топология 25](#_Toc523163594)

[Рисунок 15 - Алгоритм работы ноды 27](#_Toc523163595)

[Рисунок 16 - Алгоритм "Define Priority" 28](#_Toc523163596)

[Рисунок 17 - Пример работы алгоритма 29](#_Toc523163597)

[Рисунок 18 - Характеристики сервера для приема информации 30](#_Toc523163598)

[Рисунок 19 – Алгоритм приема информации 31](#_Toc523163599)

[Рисунок 20 - Пример приема информации 31](#_Toc523163600)

[Рисунок 21 - Пример отправки информации 32](#_Toc523163601)

[Рисунок 22 - Пример работы с HTTP сервером 33](#_Toc523163602)

[Рисунок 23 - Алгоритм BNC 34](#_Toc523163603)

[Рисунок 24 - BNC показывает 0 35](#_Toc523163604)

[Рисунок 25 - BNC показывает 1 36](#_Toc523163605)

[Рисунок 26 - BNC показывает 4 37](#_Toc523163606)

[Рисунок 27 - BNC показывает 7 38](#_Toc523163607)

# Библиография

Aravind Iyer, C. R. (2009). What is the Right Model for Wireless Channel Interference? *IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS*.

John Bicket, D. A. (2005). Architecture and Evaluation of an Unplanned 802.11b Mesh Network. *Proceedings of the 11th annual international conference on Mobile computing and networking*.

P. Ranjana, D. J. (2016). Optimized CAR based Adaptive Routing in Mesh Network. *Middle-East Journal of Scientific Research, 24*, 10-14.

Philipp Zenker, S. K. (2016). Evaluation of BLE Mesh Capabilities: A Case Study Based on CSRMesh. *ICUFN*.

Raniwala, A. (б.д.). Centralized Channel Assignment and Routing Algorithms for Multi-Channel Wi reless Mesh Networks. *50 Mobile Computing and Communications Rev*.

Shruthi Sirur, P. J. (2015). A mesh network for mobile devices using Bluetooth low energy. *IEEE SENSORS*.

Yang Cao, Z. L. (2006). A Centralized Scheduling Algorithm based on Multi-path Routing in WiMAX Mesh Network. *International Conference on Wireless Communications, Networking and Mobile Computing*.

# Для заметок