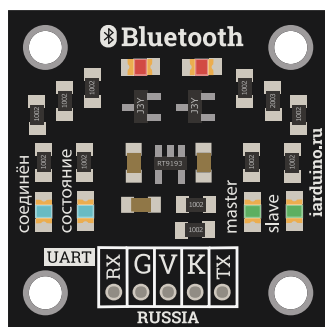


Bluetooth BLE 4.0 HM-10 (Trema-модуль v2.0)



Общие сведения:

[Trema-модуль Bluetooth BLE 4.0](#) — это модуль беспроводной связи, позволяющий передавать и принимать данные по радиоканалу на разрешённом ISM (Industry, Science and Medicine) диапазоне частот, от 2.4 ГГц до 2.5 ГГц, предназначенном для использования в промышленных, научных и медицинских целях, используя метод AFH (Adaptive Frequency Hopping Feature) - адаптивной скачкообразной перестройки несущей частоты.

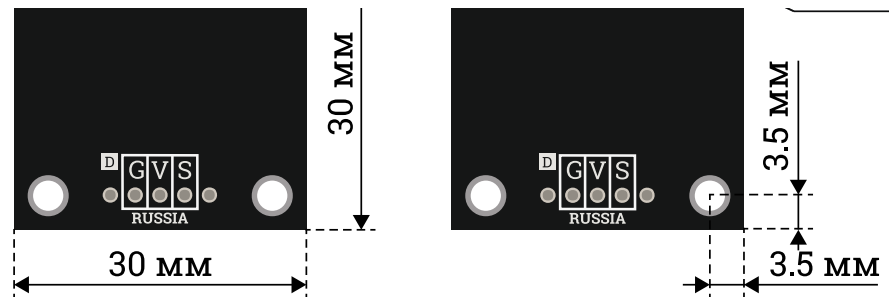
Дополнительным преимуществом данного модуля является то, что он совместим не только с оборудованием на базе ОС Android, но так же с оборудованием фирмы Apple (macOS).

Спецификация:

- Напряжение питания: 3,3 ... 5 В
- Потребляемый ток при подключении: до 40 мА (поиск, сопряжение, подключение к другим Bluetooth устройствам)
- Потребляемый ток при передаче данных: от 0,2 до 8 мА
- Частотный диапазон: ISM 2,4 ... 2,48 ГГц
- Мощность передатчика: до +6 дБм
- Дальность связи: до 10 м
- Интерфейс: UART (с программируемой скоростью передачи данных)
- Максимальное напряжение на выводах TX и RX не должно превышать напряжение питания модуля.
- PIN-код по умолчанию: 000000;
- Настройки UART по умолчанию: Скорость 9600 бит/сек, 8 бит данных, 1 стоп бит, ведомое устройство;
- Поддерживаемые скорости UART: 1200,2400,4800,9600,19200,38400,57600,115200,230400 бит/сек.

Все модули линейки "Тема" выполнены в одном формате





Подключение:

[Trema-модуль Bluetooth BLE 4.0](#) подключается к [Arduino](#) по шине UART (можно использовать как аппаратную, так и программную шину).

- Вывод модуля TX подключается к аппаратному (фиксированному) или программному (назначенному) выводу RX [Arduino](#). Это линия шины UART для передачи данных от модуля к [Arduino](#).
- Вывод модуля RX подключается к аппаратному (фиксированному) или программному (назначенному) выводу TX [Arduino](#). Это линия шины UART для передачи данных в модуль от [Arduino](#).
- Вывод модуля K подключается к любому выводу [Arduino](#), номер которого указывается в скетче. Это линия перевода модуля в режим AT-команд. Модуль в обычном режиме будет воспринимать AT-команды, только после того как на этот вывод временно подать высокий уровень. Модуль перейдёт в режим AT-команд (на скорости 38400 бит/с и не будет соединяться с другими модулями) если на нём будет установлен высокий логический уровень при подаче питания или перезагрузке.

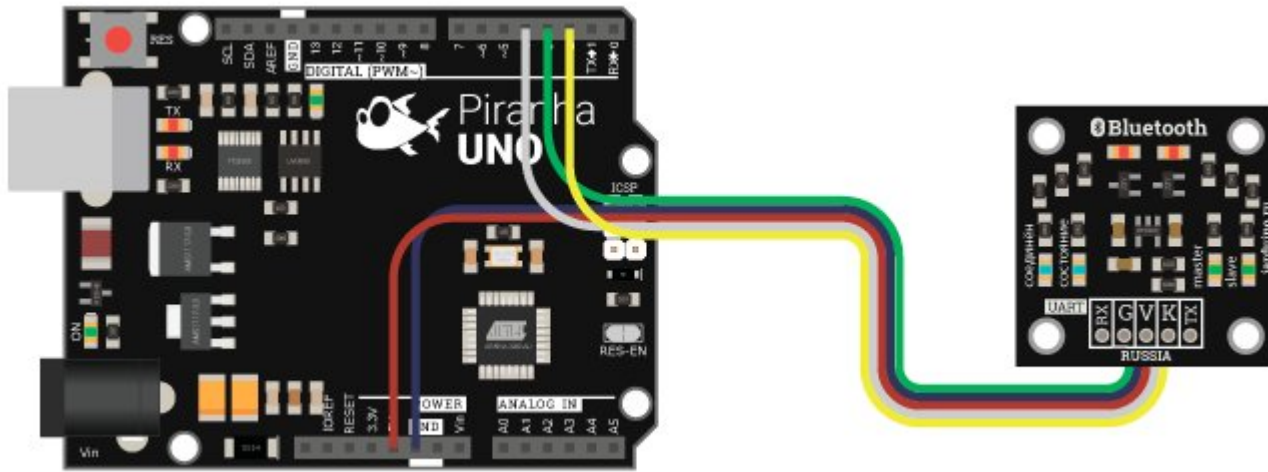
Модуль удобно подключать 3 способами, в зависимости от ситуации:

Способ - 1 : Используя проводной шлейф и Piranha UNO

Используя провода «[Папа – Мама](#)», подключаем напрямую к контроллеру Piranha UNO.

Выводы модуля Bluetooth	Выводы Piranha Uno	Цвет
RX	D3	Зелёный
TX	D2	Жёлтый

Выходы модуля Bluetooth	Выходы Piranha Uno	Цвет
G	GND	Синий
V	5V	Красный
K	D4	Серый



Способ - 2 : Используя Trema Set Shield

Модуль можно подключить к UART входу Trema Set Shield.

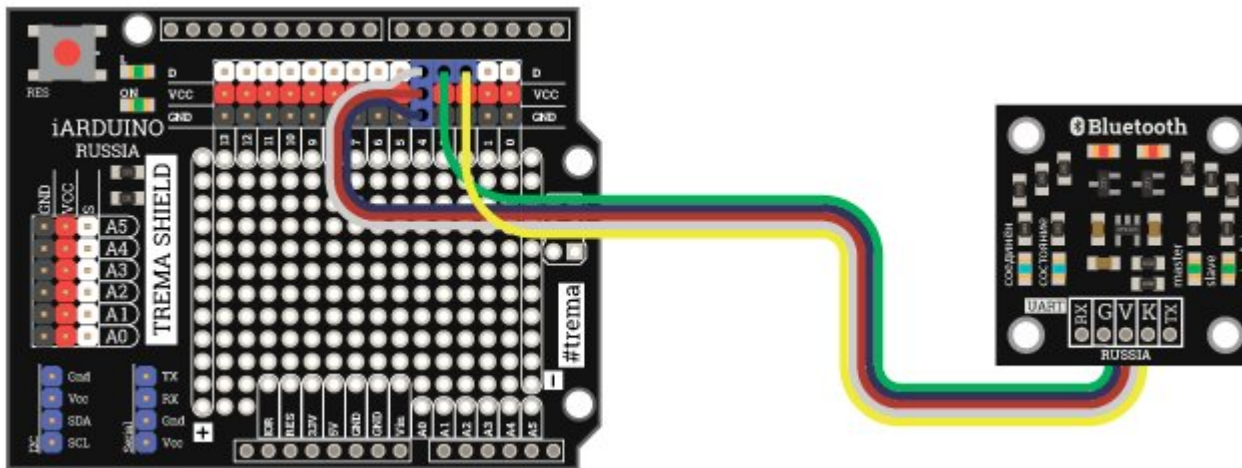




Способ - 3 : Используя проводной шлейф и Shield

Используя 2-х и 3-х проводные шлейфы, к Trema Shield, Trema-Power Shield, Motor Shield, Trema Shield NANO и тд.

Выводы модуля Bluetooth	Выводы Trema Shield	Цвет
RX	D3	Зелёный
TX	D2	Жёлтый
G	GND	Синий
V	VCC	Красный
K	D4	Серый



Способ - 4 : Используя проводной шлейф и адаптер USB-UART

Используя 5 1-х проводных шлейфов и [адаптер USB-UART](#), подключаем модуль напрямую к ПК.

Выходы модуля Bluetooth	Выходы адаптера	Цвет
RX	TX	Зелёный
TX	RX	Жёлтый
G	GND	Синий
V	VCC	Красный
K	DTR	Серый



Питание:

Входное напряжение питания 3,3 или 5 В постоянного тока, подаётся на выводы Vcc и GND модуля.

Управление светодиодной индикацией

Модуль имеет 4 светодиода на корпусе:

- **Соединён** — синего цвета;
- **Состояние** — синего цвета;
- **Master** — зелёного цвета;
- **Slave** — зелёного цвета;

Данными светодиодами возможно управлять с помощью AT-команд.

Пример ниже демонстрирует, каким образом можно управлять работой светодиодов, расположенных на модуле.

```
#include <SoftwareSerial.h> // Подключаем библиотеку для работы с программным UART
uint8_t TX = 2; // Указываем порт, к которому подключен вывод TX модуля BLE
uint8_t RX = 3; // Указываем порт, к которому подключен вывод RX модуля BLE
uint8_t Key = 7; // Указываем порт, к которому подключен вывод KEY модуля BLE
char A; // Задаём переменную для считывания значений от модуля
String B = ""; // Задаём переменную для хранения строки данных, которые отправил модуль

SoftwareSerial BT(TX, RX); // Создаём объект BT, указывая номера выводов TX и RX

void setup() {
  BT.begin(115200); // Иницируем работу с последовательным портом модуля BLE на скорости 9600
  Serial.begin(9600); // Иницируем работу с монитором последовательного порта на скорости 9600
  digitalWrite(Key, HIGH); // Подаём высокий сигнал на вывод Key для перевода модуля в режим работы
  digitalWrite(Key, LOW); // Подаём низкий сигнал на вывод Key
  //-----//
  BT.write("AT+MODE0"); delay(500); // Отправляем AT-команду перевода модуля в режим работы MODE 0 и ждём 500 мс
  while (BT.available()) { // Если в последовательном порту модуля есть данные, то
    Serial.write(BT.read()); // считываем их и выводим в монитор последовательного порта
  }
  Serial.println(); // Следующее сообщение будет отображаться с новой строки
  //-----//
}
```

```

BT.write("AT+BEFC008"); delay(500); // Отправляем AT-команду для подачи питания на светодиод "Состояние" с
while (BT.available()) { // Если в последовательном порту модуля есть данные, то
  Serial.write(BT.read()); // считываем их и выводим в монитор последовательного порта
}
Serial.println(); // Следующее сообщение будет отображаться с новой строки
//-----//
BT.write("AT+ROLE0"); delay(500); // Отправляем AT-команду перевода модуля в режим работы slave(0)/master
while (BT.available()) { // Если в последовательном порту модуля есть данные, то
  A = BT.read(); // считываем их в переменную и
  B.concat(A); // добавляем полученные данные в строку
}
if (B == "OK+Set:0") { // Проверяем, если полученный ответ на установку роли = 0 (устройство в
  BT.write("AT+AFTC204"); // отправляем AT-команду включить светодиод "Slave" на модуле после тог
  Serial.println("Slave-mode on"); // выводим текст в монитор последовательного порта
} else if (B == "OK+Set:1") { // Если же получен ответ = 1 (устройство ведущее), то
  BT.write("AT+AFTC104"); // отправляем AT-команду включить светодиод "Master" на модуле после то
  Serial.println("Master-mode on"); // выводим текст в монитор последовательного порта
}
//-----//
BT.write("AT+RESET"); delay(500); // Отправляем AT-команду для перезагрузки модуля
}

void loop() {

```

Как видно из примера выше, после загрузки скетча в плату, на [Trema-модуле Bluetooth BLE 4.0](#) будет гореть синий светодиод с надписью состояние, что означает, что модуль находится в ожидании подключения. Сразу после того, как модуль будет сопряжён с любым другим устройством, в зависимости от того, в каком режиме он настроен на работу (master/slave), на модуле загорятся синий светодиод с надписью "Соединён" и зелёный светодиод с надписью режима работы модуля (master или slave).

Работа светодиодной индикации в зависимости от роли модуля

В зависимости от указанной роли модуля (**master** или **slave**), светодиодная индикация будет работать по разному!

Если Вы используете для сопряжения друг с другом 2 [Trema-модуля Bluetooth BLE 4.0](#), то важно помнить, что модуль, настроенный на работу в режиме master, после установления соединения с устройством, работающим в режиме slave, отключает свою светодиодную индикацию на всё время соединения.

Подробнее о модуле:

В ISM диапазоне частот работают и [радио модули nRF24L01+](#), но в отличие от этих модулей, которые работают на определённой частоте диапазона, [Trema-модуль Bluetooth BLE 4.0](#) используют метод AFH (Adaptive Frequency Hopping Feature) адаптивной скачкообразной перестройки несущей частоты (он меняет свою частоту 1600 раз в секунду). Несущая частота меняется псевдослучайным образом и заранее известна только паре «ведущий - ведомый», что обеспечивает не только устойчивость к помехам (занят канал? не беда, перейдём на другой) но и сохранение конфиденциальности передаваемых данных.

Преимуществом Bluetooth модулей перед другими модулями беспроводной передачи данных заключается в простоте работы (Вам не нужно знать протоколы, работать с регистрами, отслеживать сигналы и т.д.) и широкой распространённости данного типа передачи данных (Вы можете управлять Вашими устройствами, или получать их показания, практически с любого телефона, планшета, ноутбука).

Модуль не требует подключения антенны, т.к. она встроена (присутствует на ПП модуля).

Принцип работы BLE

Помимо модуля, в ISM диапазоне частот работают многие другие устройства, такие как Wi-Fi -роутеры или микроволновые печи. Совместная работа сразу нескольких подобных устройств может мешать работе модуля. Для того, чтобы минимизировать влияние сторонних устройств, используются **широковещательные каналы объявления (Advertising channels, "рекламные каналы")**. Данные каналы имеют следующие индексы: **37, 38 и 39**.

 Рис. 6. Широковещательные объявления

Их использование не случайно:

— частоты данных каналов попадают между каналами Wi-Fi (1, 6, 11 каналы), что позволяет свести к минимуму влияние Wi-Fi сигнала на модуль;

— после того, как каналы расположены максимально далеко друг от друга, а передача **широковещательных объявлений** идёт последовательно по 3 каналам на разной частоте, можно быть уверенным в том, что объявления дойдут до получателя.

 Рис. 7. Полоса частот и каналы Bluetooth Low Energy

После того, как соединение между устройствами установлено, передача данных осуществляется уже по стандартным, **рабочим** каналам (с 0 по 36).

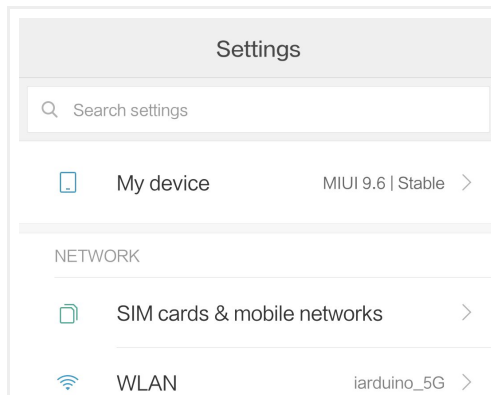
При этом принцип работы BLE сводится к тому, что устройства работают в следующем режиме: **Проснулся** — **Передал данные** — **Уснул**, за счёт чего и достигается минимальное значение энергопотребления.

Пример:

Для того, чтобы Вы могли опробовать модуль в работе, для начала Вам необходимо скачать и установить терминал, благодаря которому возможно отправлять AT-команды модулю для настройки его работы и получения необходимой информации.

Следующий пример представлен для телефонов с установленной ОС Android, но все эти действия аналогичны при работе на устройствах с ОС Apple (macOS).

Настройка Bluetooth-модуля телефона:



Зайдите в настройки телефона и выберите настройки модуля Bluetooth;



Portable hotspot Off >

Data usage >

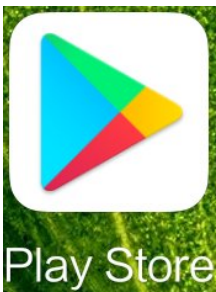
< Bluetooth

Переведите модуль Bluetooth из состояния выключен в состояние включен.

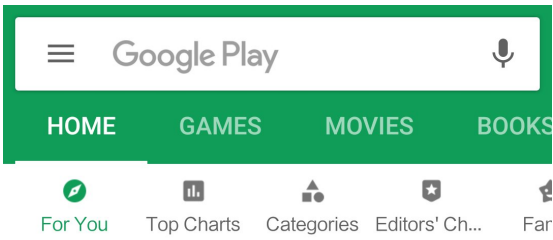
Bluetooth



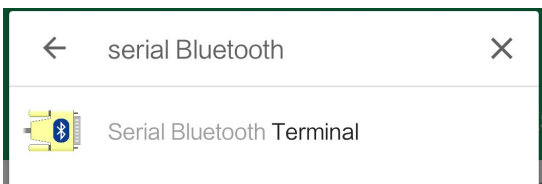
Установка приложения на телефон



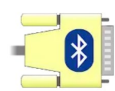
Войдите в меню магазина приложений **Google Play Market** ;



Обратите внимание на строку поиска в верхней части экрана, где написано **Google Play** ;



Нажмите на строку поиска и наберите **Serial Bluetooth Terminal** и выберите первую появившуюся строку с именем приложения;



Serial Bluetooth Terminal

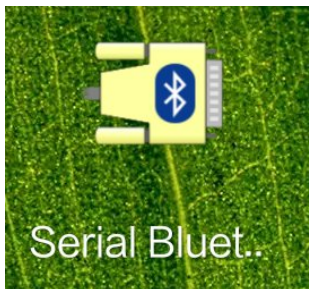
Kai Morich

Tools

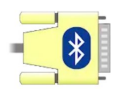
INSTALL

In-app purchases

В окне установки приложения нажмите кнопку **Установить** ;



После успешной установки иконка программы появится у вас на рабочем столе телефона;



Serial Bluetooth Terminal

Kai Morich

Tools

UNINSTALL

OPEN

In-app purchases

Для запуска приложения достаточно один раз нажать кнопку **Открыть** или выбрать иконку приложения на рабочем столе;

Скетч для Arduino

Загрузите следующий скетч в вашу плату.

```
#include <SoftwareSerial.h> // Подключаем библиотеку для работы с программным UART
uint8_t TX = 2; // Указываем порт, к которому подключен вывод TX модуля BLE
uint8_t RX = 3; // Указываем порт, к которому подключен вывод RX модуля BLE
uint8_t KEY = 7; // Указываем порт, к которому подключен вывод KEY модуля BLE
char A; // Задаём переменную для считывания значения, отправленных с телефона
```

```

uint8_t LED13 = 13; // Задаём переменную для вывода, к которому подключен светодиод на плате

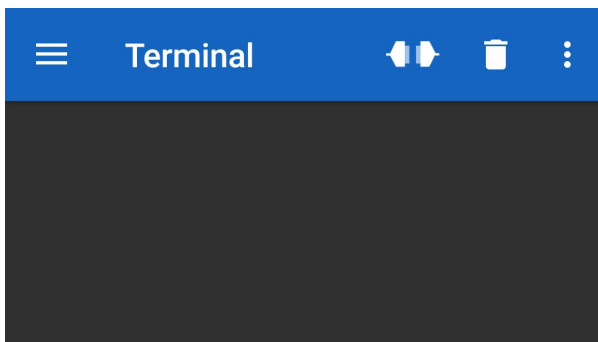
SoftwareSerial BT(TX, RX); // Создаём объект BT, указывая номера выводов TX и RX

void setup() {
  BT.begin(9600); // Инициуруем работу с последовательным портом модуля BLE на скорости 9600 бод
  BT.write("AT+NAMETest"); // С помощью AT-команды задаём имя устройства, которое будет видно в эфире
  delay(100); // Задержка в 100мс
}

void loop() {
  if (BT.available()) { // Если в последовательном порте модуля появились данные, то
    A = BT.read(); // считываем их в переменную A
  }
  if (A == '1') { // Проверяем, если значение A = 1, то
    digitalWrite(LED13, HIGH); // включает светодиод
    BT.write("Светодиод включен"); // отправляем уведомление на телефон
  } else if (A == '0') { // если же значение A = 0, то
    digitalWrite(LED13, LOW); // гасим светодиод и
    BT.write("Светодиод выключен"); // отправляем уведомление на телефон
  }
}

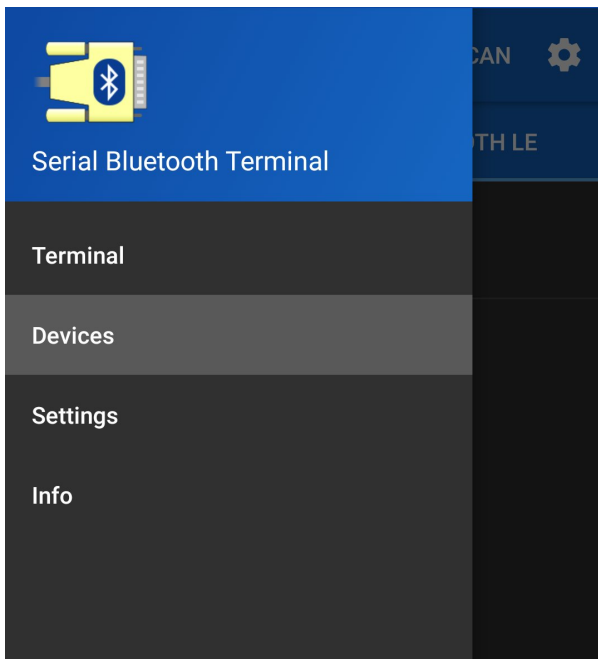
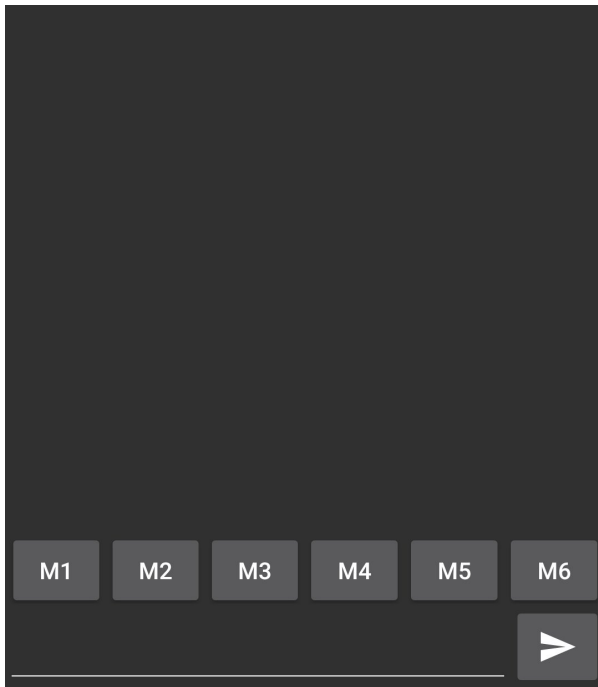
```

Настройка приложения Serial Bluetooth Terminal на телефоне:

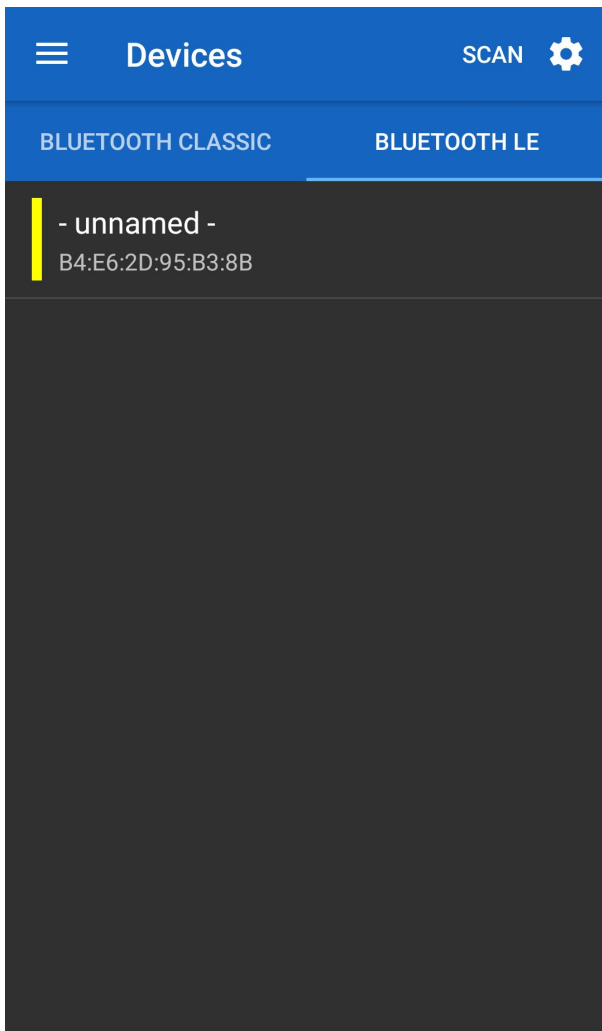


При запуске приложения вы попадёте на главный экран.

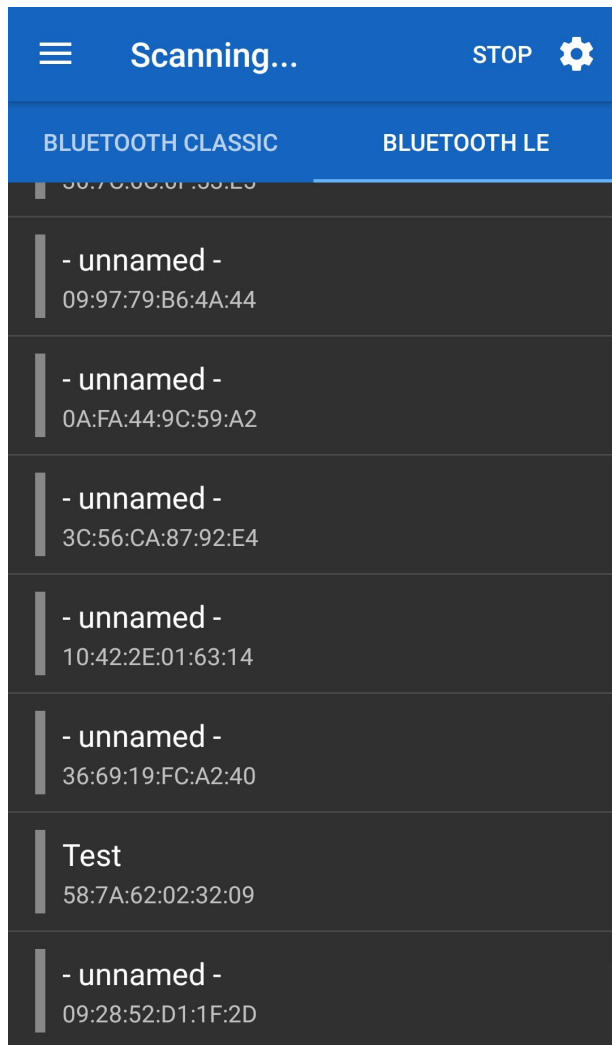
Для поиска устройств нажмите на изображение трёх горизонтальных линий в левом верхнем углу.



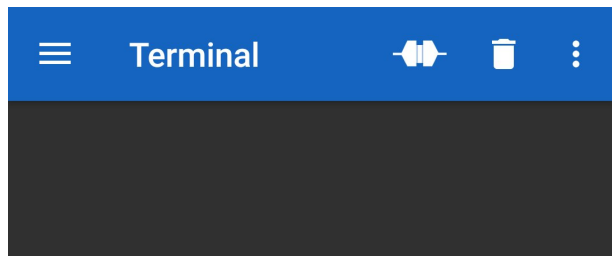
После нажатия на кнопку в появившемся меню Вам необходимо нажать на строку **Devices** ;



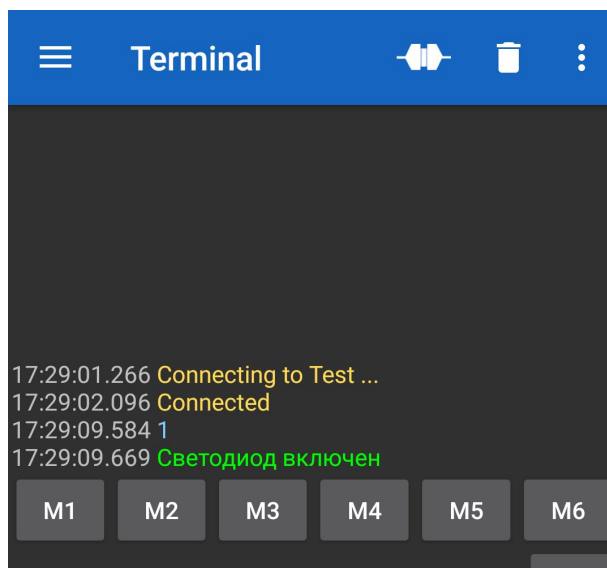
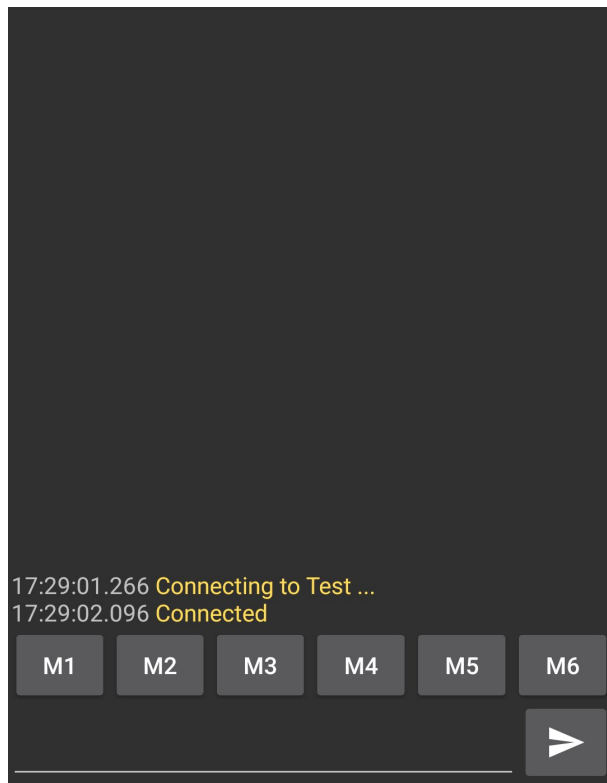
В появившемся окне будет представлен список всех Bluetooth-устройств в радиусе действия Bluetooth-адаптера смартфона. Для начала работы нажмите на строку с название в верхнем правом углу и дождитесь окончания поиска.



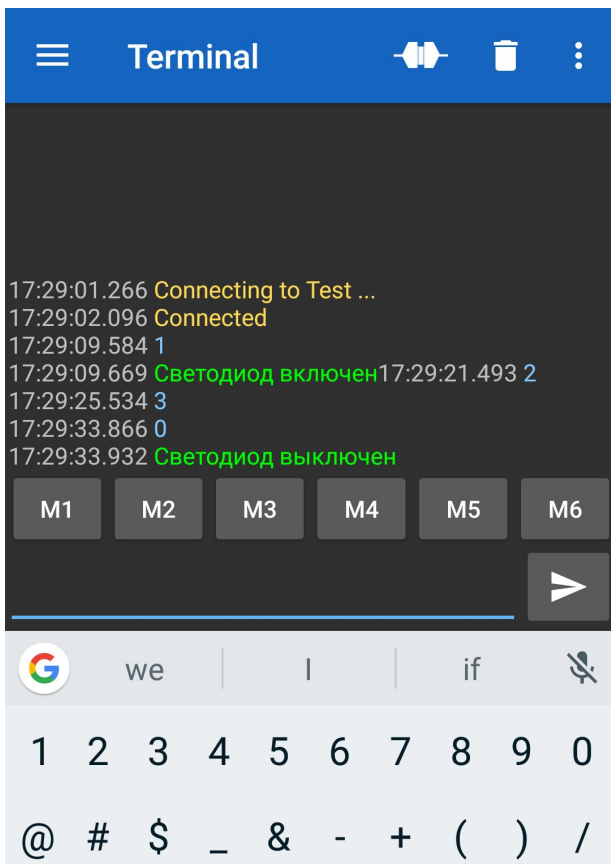
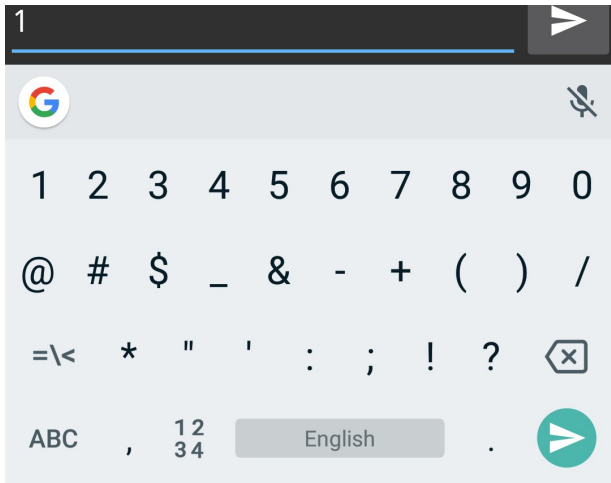
После того, как сканирование будет завершено, найдите в списке устройство с именем `Test` и нажмите на него. Произойдёт сопряжение устройств и Вас вернёт на главный экран приложения.



На главном экране будет написано `Connected`, что означает, что сопряжение произошло удачно и соединение установлено. Теперь, для того, чтобы отправить в модуль команду на включение светодиода, нажмите на строку в нижней части экрана, введите `1` и отправьте в модуль, нажав на кнопку в правом нижнем углу.



После того, как команда будет отправлена, вам на телефон придёт ответное сообщение о том, что светодиод включен. Отправляя любые другие символы, кроме 0 и 1, модуль ни как не будет реагировать на них. Отправив 0, вы выключите светодиод.



Таким образом, отправляя команды модулю, Вы можете управлять устройствами, подключенными к Arduino.



АТ-команды

Для предварительной или более детальной настройки модуля, используются АТ-команды. Список команд Вы найдёте [тут](#) или в [DataSheet](#)'е к модулю.

Применение:

- Создание связи между двумя [Arduino](#).
- Создание связи между Arduino и другими ведущими Bluetooth устройствами: телефонами, планшетами, компьютерами и т.д.
- Создание связи между Arduino и другими ведомыми Bluetooth устройствами: гарнитурами, клавиатурами, мышками и т.д.
- Дистанционное управление роботами, устройствами, проектами и т.д.
- Дистанционное получение данных от датчиков, детекторов, сигнализаций и т.д.
- Создание Bluetooth ретрансляторов для увеличения дальности беспроводной связи.

Ссылки:

- [DataSheet](#);
- [Wiki - Установка библиотек в Arduino IDE](#).
- [Wiki - АТ-команды Bluetooth BLE](#)