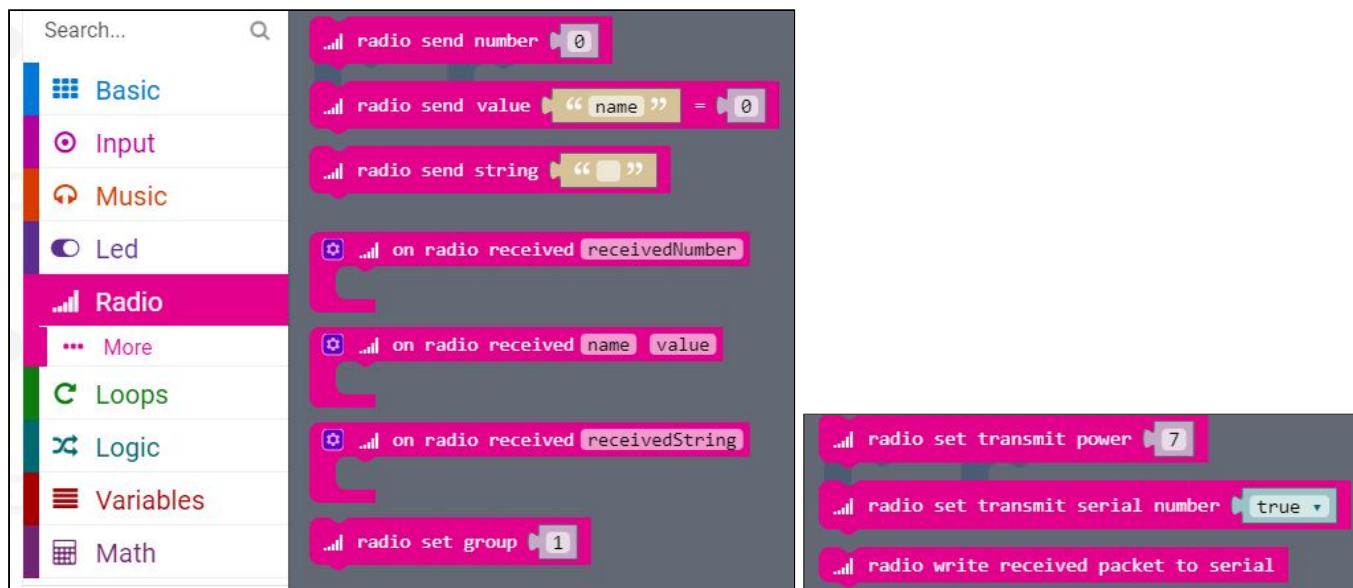


## Программируем BBC micro:bit

### Шаг 7. Беспроводное соединение устройств

На этом шаге мы познакомимся с возможностями, которые нам предоставляет встроенный в микроконтроллер BBC micro:bit Bluetooth модуль (точнее, BLE - [Bluetooth Low Energy](#)) и группой команд **Radio**, которые эти возможности реализуют.



С помощью беспроводного соединения можно связать между собой несколько микроконтроллеров, а также микроконтроллер и смартфон.

Установив на смартфон необходимое ПО, можно получать и выводить на экран информацию от датчиков микроконтроллера, и наоборот, посылать управляющие сигналы на микроконтроллер.

#### Примеры:

- [Take The BBC micro:bit To The Next Level With Bluetooth Low Energy](#)
- [BBC microbit basic bluetooth capabilities demo](#)
- [How to use a BBC micro:bit as a music remote for your phone](#)

Протестировать в эмуляторе связь смартфон — микроконтроллер мы не можем, поэтому ограничимся знакомством с примерами. А вот тестировать связь двух микроконтроллеров эмулятор позволяет, и мы рассмотрим несколько задач.

#### Назначение команд



Устанавливает мощность излучения, которая может меняться от 0 до 7. [Разработчики утверждают](#), что при значении transmit power, равном 7, в открытом пространстве дистанция устойчивого соединения может достигать 70 метров.



С помощью этой команды несколько микроконтроллеров объединяются в группу и могут “слушать” друг друга. Микроконтроллеры, не входящие в группу, сигналы группы “не слышат”.



Выполняя эти команды, микроконтроллер передаёт в эфир числовое значение (первая команда) или строку (вторая команда).

## События

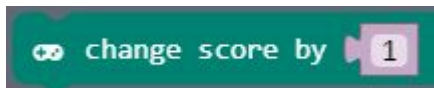


Первое событие наступает при получении числового значения, второе — при получении строкового сообщения.

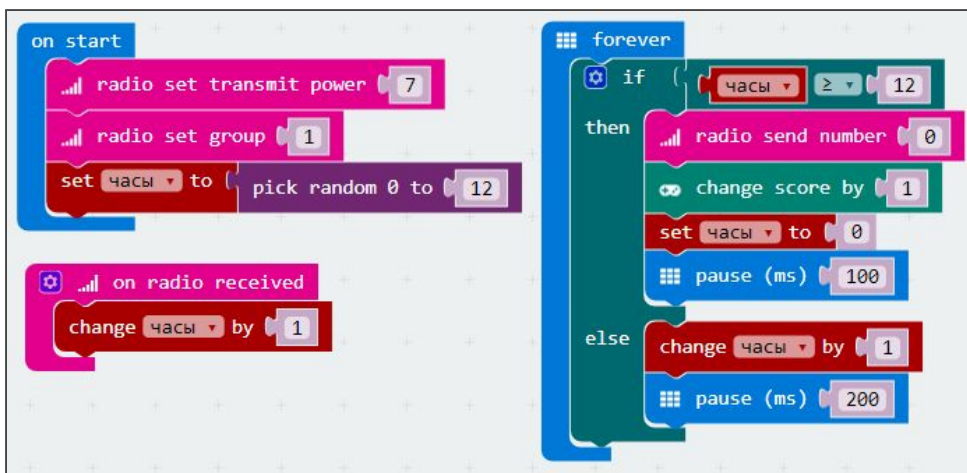
Рассмотрим интересный [проект “Светлячки”](#) в качестве примера взаимодействия нескольких микроконтроллеров с использованием Bluetooth соединения.

Всем известно, что светлячки способны излучать мерцающий свет, но не всем известно, что колония светлячков синхронизирует отправку своих сообщений. В алгоритме, который мы рассмотрим, процесс отправки / приёма сообщения с постепенной синхронизацией реализован следующим образом.

1. У светлячка есть внутренние часы, и отправка сообщения выполняется на каждый 12-й тик этих внутренних часов.
2. В начале сессии светлячок отправляет в эфир сигнал в случайный момент времени.
3. Для синхронизации светлячок корректирует свои внутренние часы: в момент приёма сигнала от другого светлячка он “подтягивает” свои часы на 1 тик для уменьшения запаздывания. Такая корректировка постепенно приводит к групповой синхронизации отправки.
4. Анимация самого сигнала (свечения) выполняется с помощью команд группы **Game**. На самом деле, команды этой группы — не отдельные команды, а эффектные процедуры, которые могут значительно облегчить процесс кодирования при разработке игры. Из всего многообразия этих процедур для проекта “Светлячки” выбирается анимация



## Код программы (можно проверить в эмуляторе)



## Другие примеры

- [Micro:bit bluetooth communication \(Turn LED on and off\)](#)
- [Using the 'Radio' service to connect micro:bit to micro:bit](#)

**Пробуем, придумываем, радуемся успеху!**

