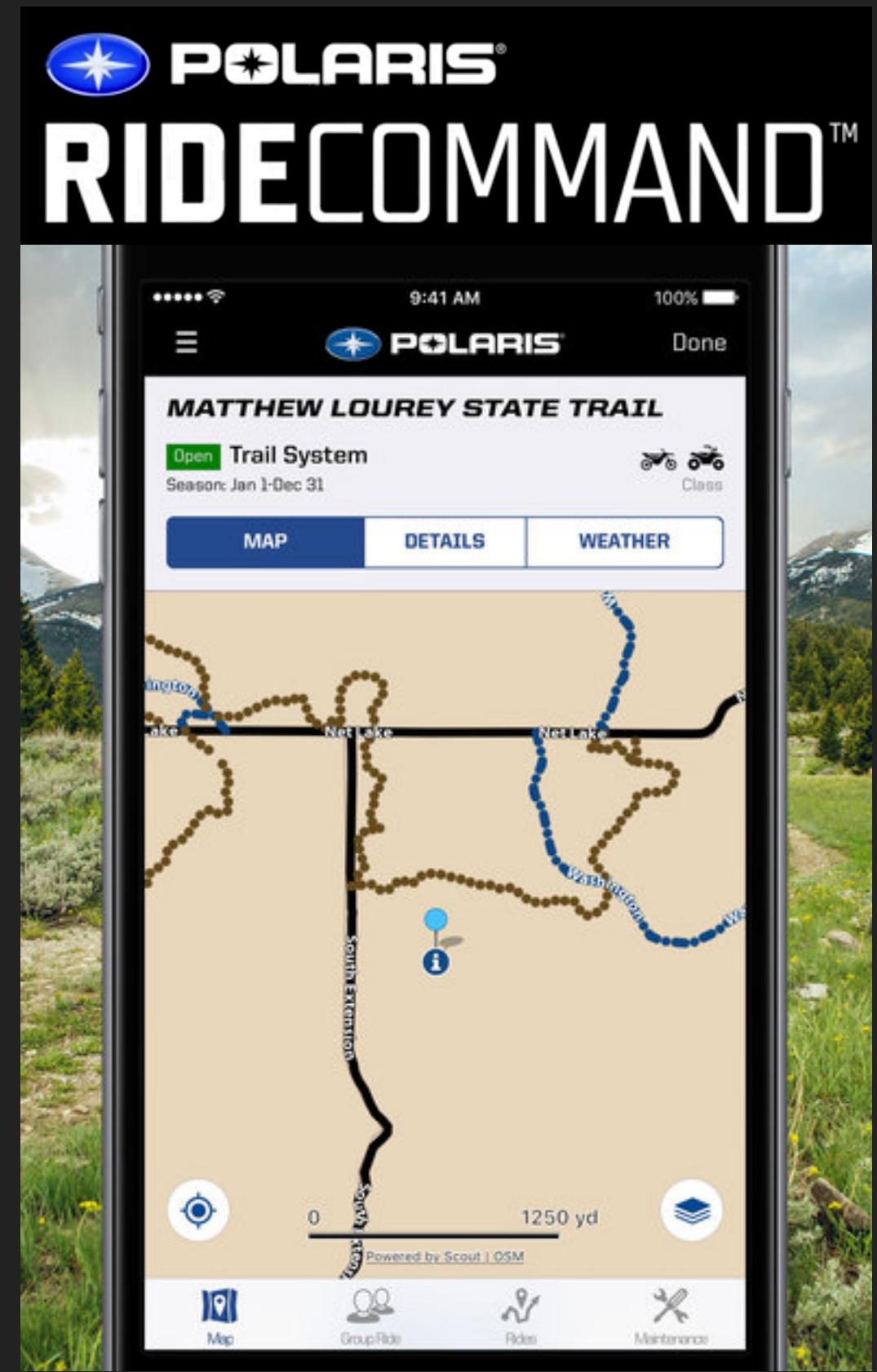
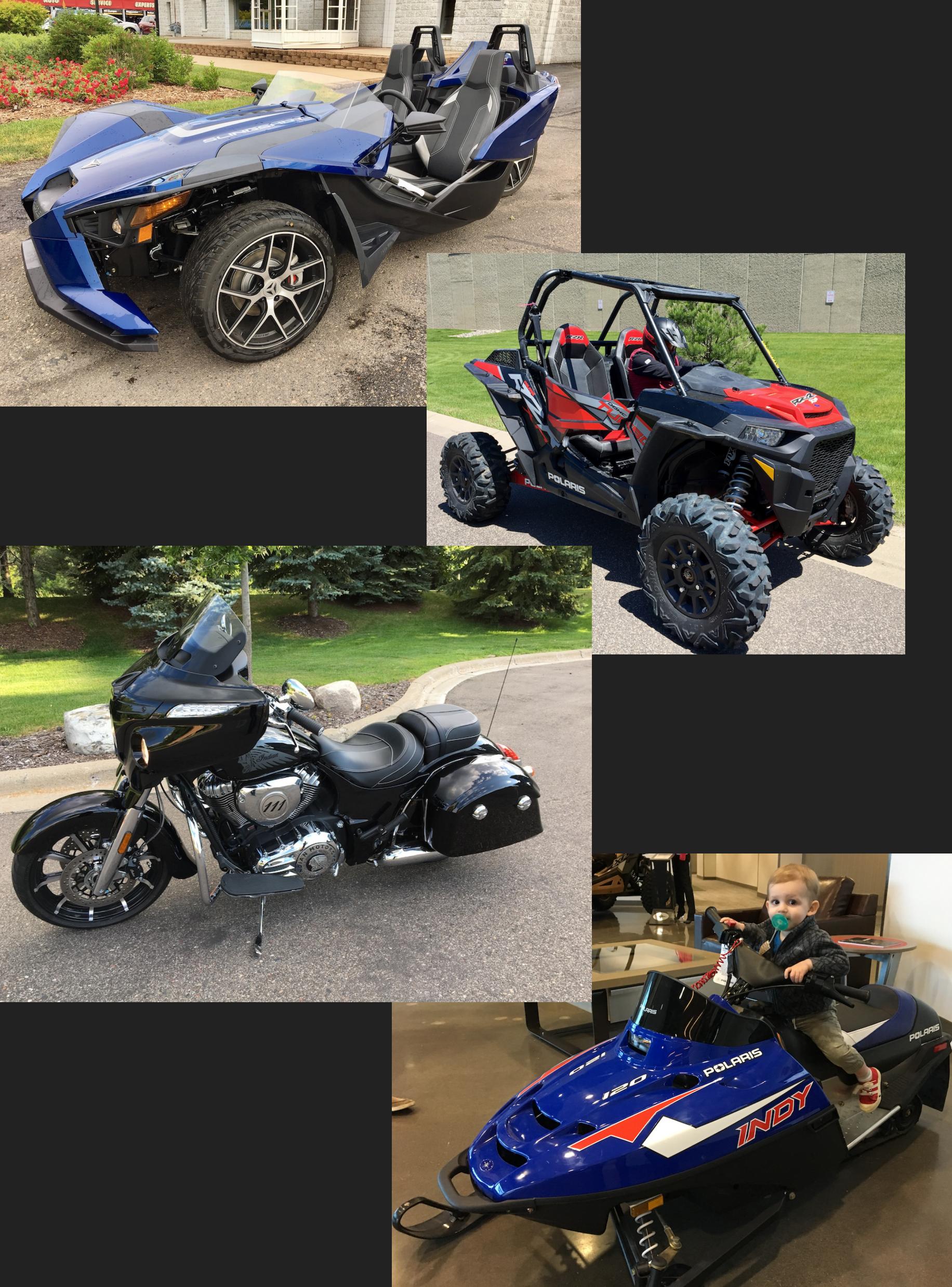


# BLUETOOTH LOW ENERGY

---

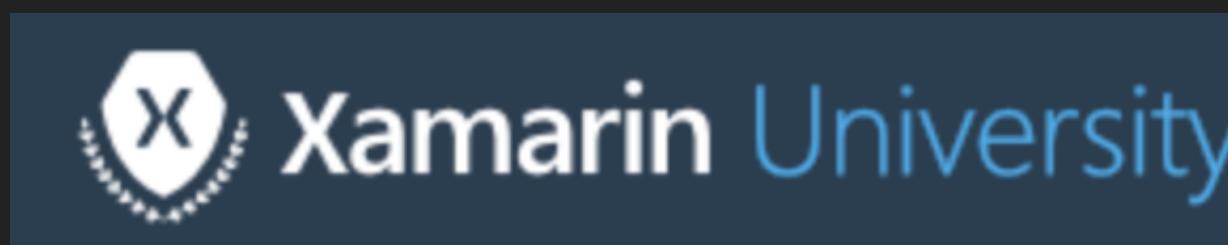
FOR XAMARIN DEVELOPERS

# ABOUT ME



Blog: <https://tomsoderling.github.io>

Twitter: @tomsoderling



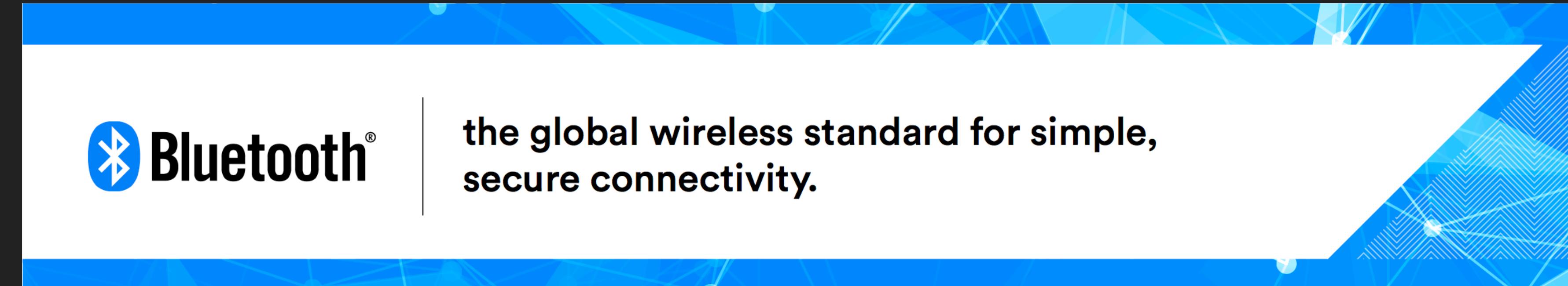
# OUTLINE

---

1. Bluetooth Basics
  - A. Two Flavors
  - B. How It Works (somewhat) + Trivia
2. BLE Basics
  - A. What's the Difference?
  - B. How Fast?
  - C. GAP
  - D. GATT
3. Handy BLE Tools
4. Building a BLE Xamarin App
5. Designing your own BLE Peripherals

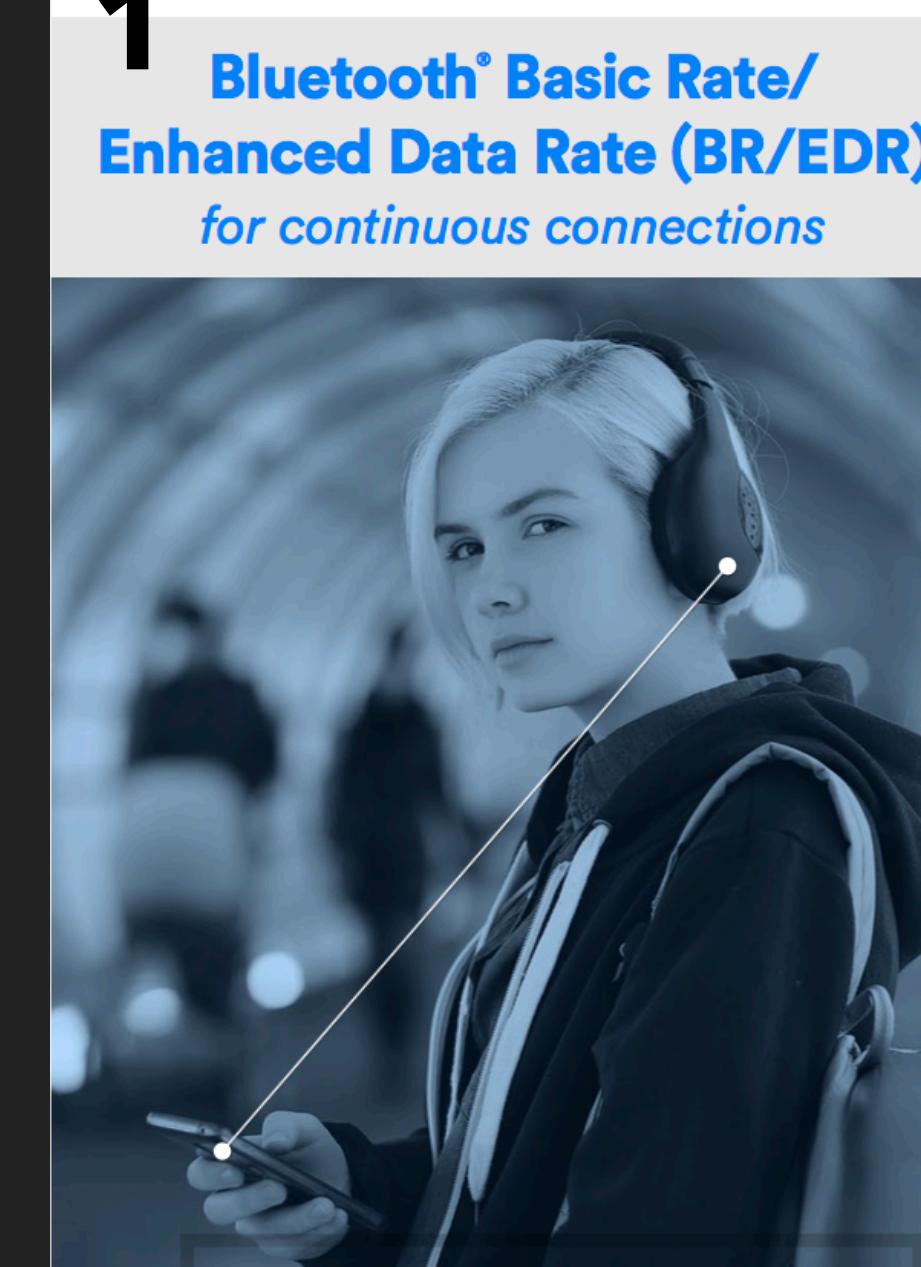
# BLUETOOTH BASICS: TWO FLAVORS

- ▶ Basic Rate/  
Enhanced Data Rate  
(BR/EDR)
- ▶ Low Energy (LE)



**Bluetooth®**  
the global wireless standard for simple,  
secure connectivity.

**1** **Bluetooth® Basic Rate/  
Enhanced Data Rate (BR/EDR)**  
*for continuous connections*

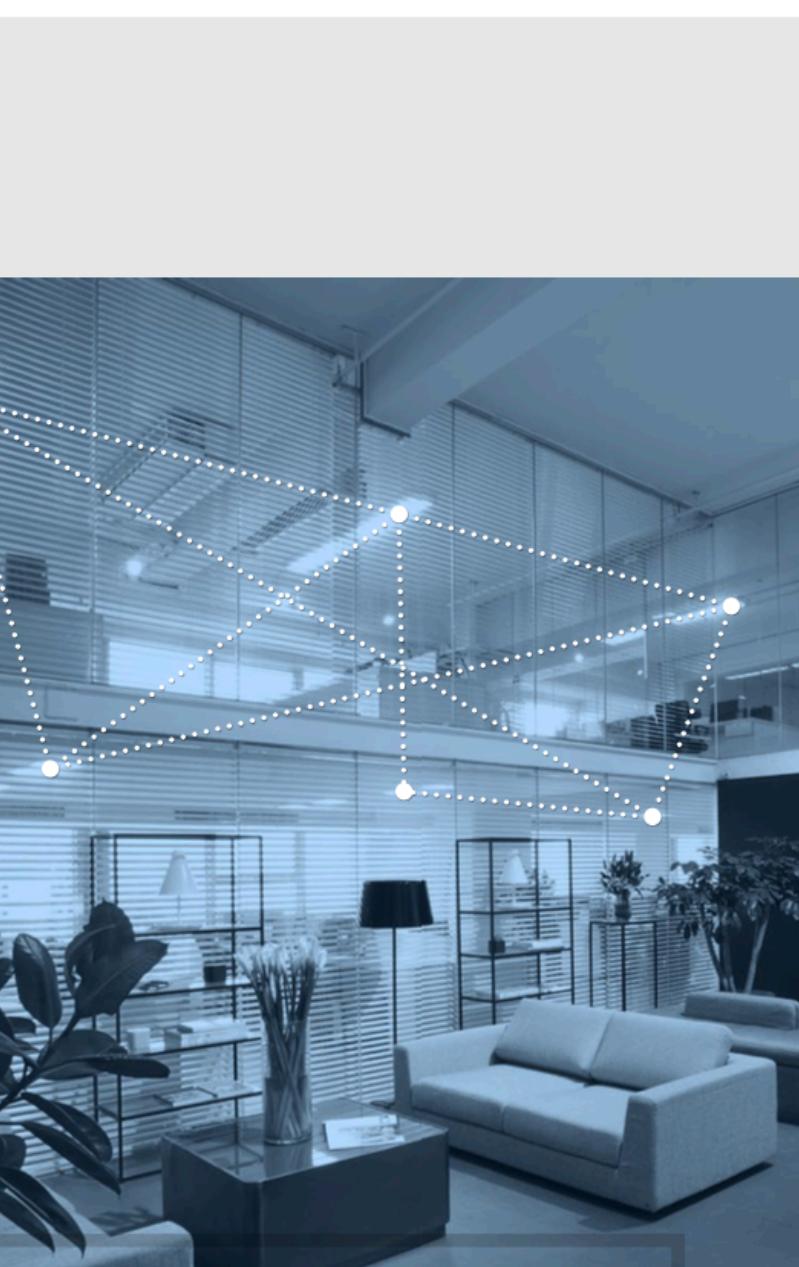
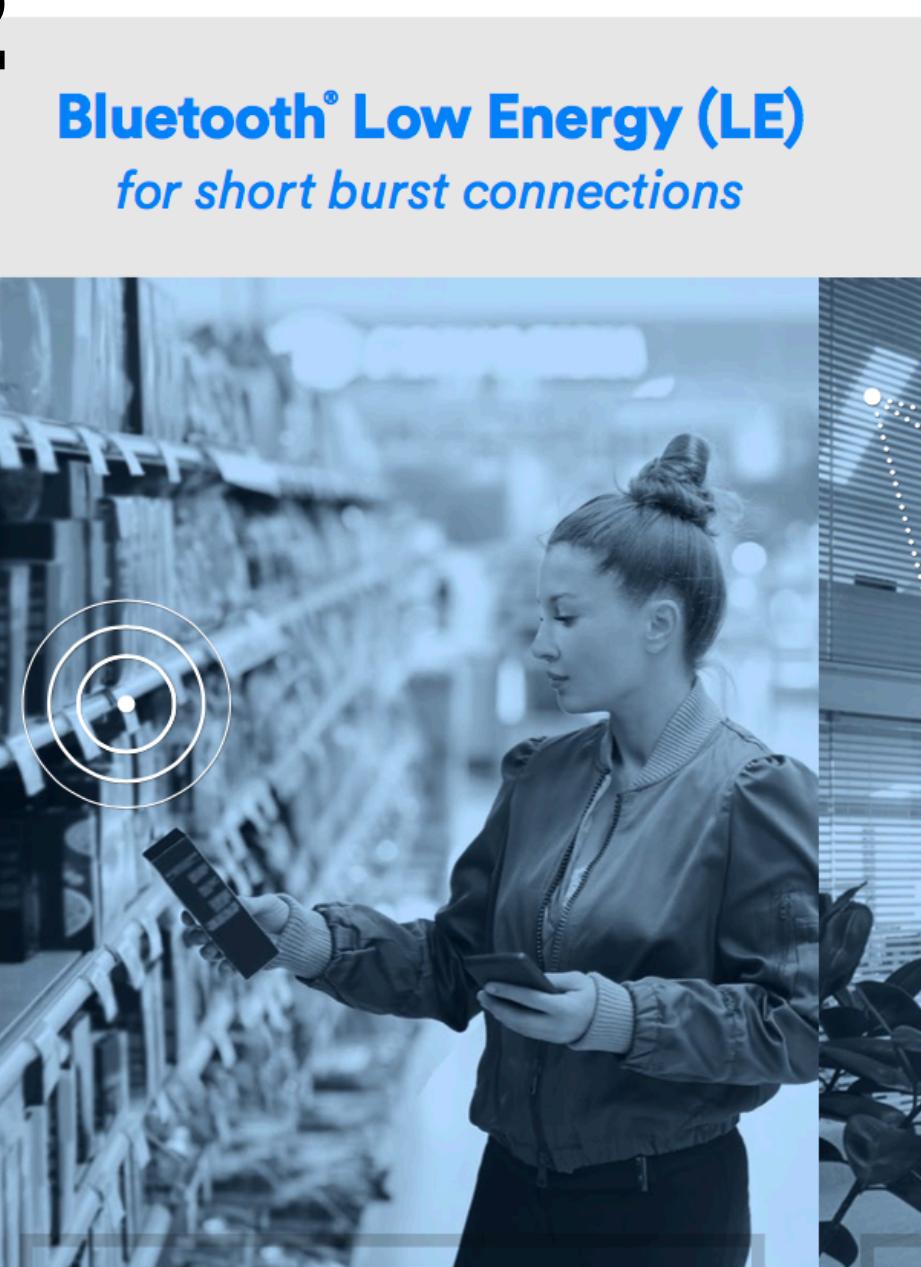
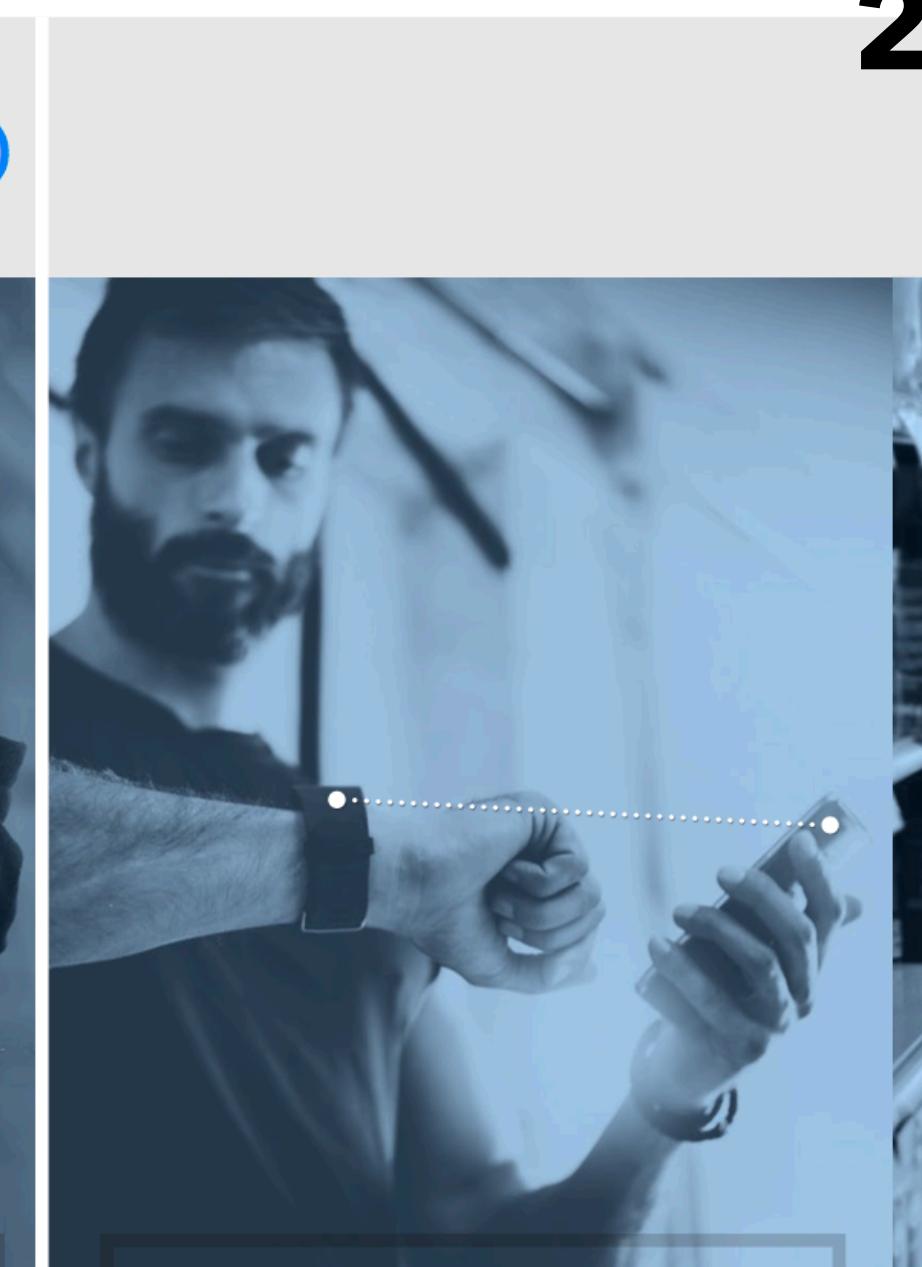


**point-to-point (1:1)**

**audio streaming**

- wireless headsets
- wireless speakers
- in-car audio

**2** **Bluetooth® Low Energy (LE)**  
*for short burst connections*



**broadcast (1:m)**

**localized information**

- point of interest beacons
- item finding beacons
- way finding beacons

**mesh (m:m)**

**large device networks**

- building automation
- wireless sensor networks
- asset tracking

# BLUETOOTH BASICS: HOW IT WORKS

---

- ▶ Operates on the unlicensed ISM (Industrial, Scientific and Medical) radio bands at 2.4 GHz (along with wifi, cordless phones, microwave ovens)
- ▶ Uses an algorithmically determined frequency hopping pattern that is a pseudo-random ordering of frequencies separated by 1 MHz.
  - ▶ BR/EDR: 79 frequencies
  - ▶ BLE: 37 frequencies
- ▶ Hopping pattern can be adapted to exclude frequencies that are used by interfering devices to improve co-existence on the ISM band.
- ▶ Named after King Harald Gormsson who was well known for two things:
  - ▶ Uniting Denmark and Norway in 958.
  - ▶ Had a dead tooth that was a dark blue/grey color, and earned him the nickname “Bluetooth”



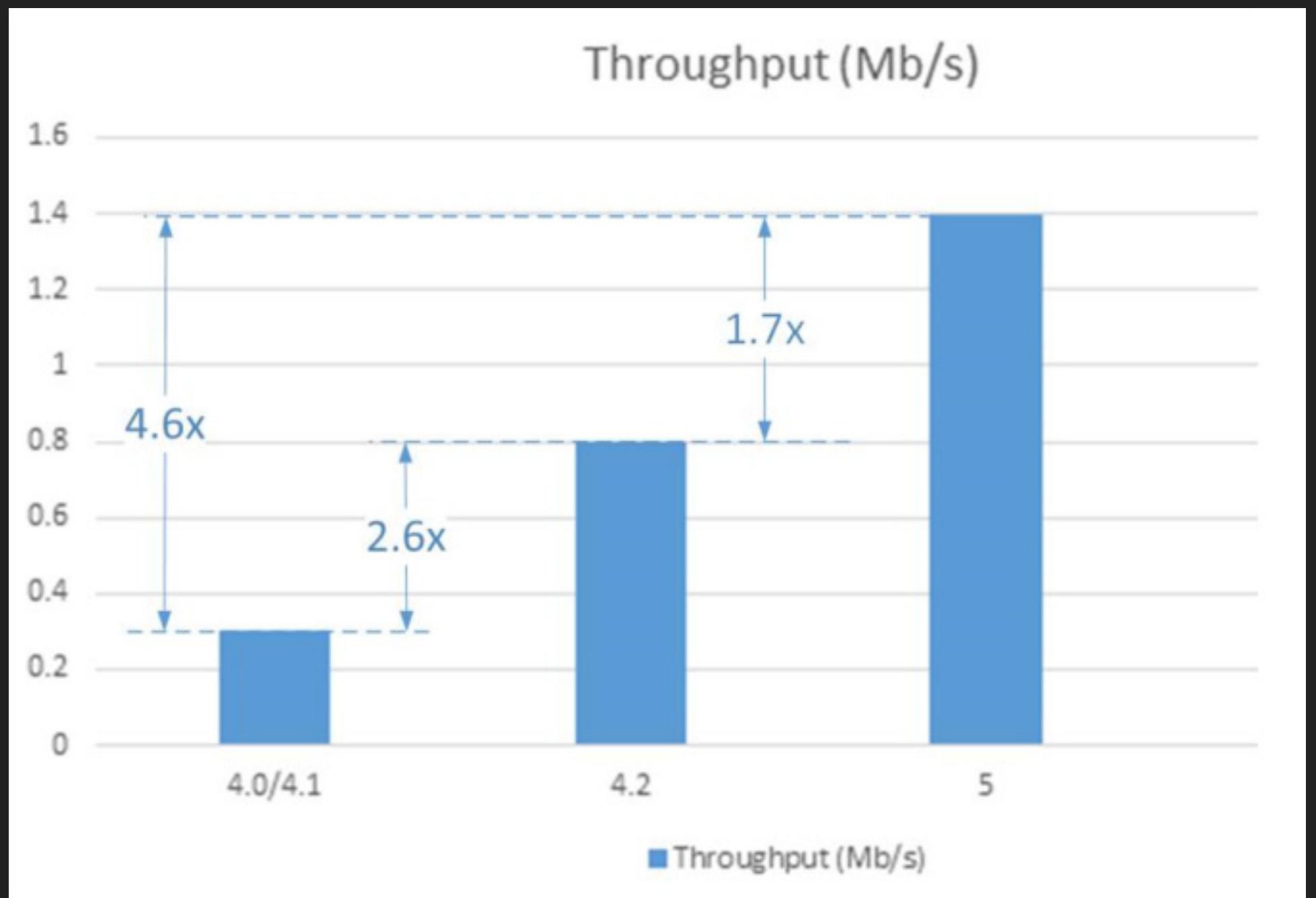
# BLE BASICS: WHAT'S THE DIFFERENCE?

- ▶ AKA "Bluetooth Smart", or "Bluetooth 4.0"
- ▶ Some overlap with classic Bluetooth, but from different source entirely: Nokia
- ▶ Adopted by the Bluetooth SIG (Special Interest Group) as part of Bluetooth 4.0 core specification (core spec 5.0 now, 2,822 pages)
- ▶ No pairing!
- ▶ Easiest way to design a device that can talk to any modern mobile platform (iOS, Android, Windows phones, etc.)
  - ▶ No Made For iPhone (MFi) certification needed
- ▶ Designed for lower power consumption, lower complexity, lower \$ than BR/EDR
- ▶ BLE consumes between 1/2 and 1/100 the power of classic Bluetooth



# BLE BASICS: HOW FAST?

- ▶ How Fast?
  - ▶ BLE throughput by core spec version



# BLE BASICS: GAP

---

- ▶ GAP (Generic Access Profile)
  - ▶ GAP controls connections and advertising in Bluetooth. GAP is what makes your device visible to the outside world, and determines how two devices can (or can't) interact with each other.
  - ▶ There are 2 main device **roles** that GAP defines:
    1. **Peripheral** devices: small, low power, resource constrained devices: Heart rate monitor, Beacon, Toaster, etc.
    2. **Central** devices: usually the mobile phone or tablet that you connect to with far more processing power and memory.



# BLE BASICS: GAP ADVERTISEMENT

- ▶ Two ways to send advertising data out with GAP:

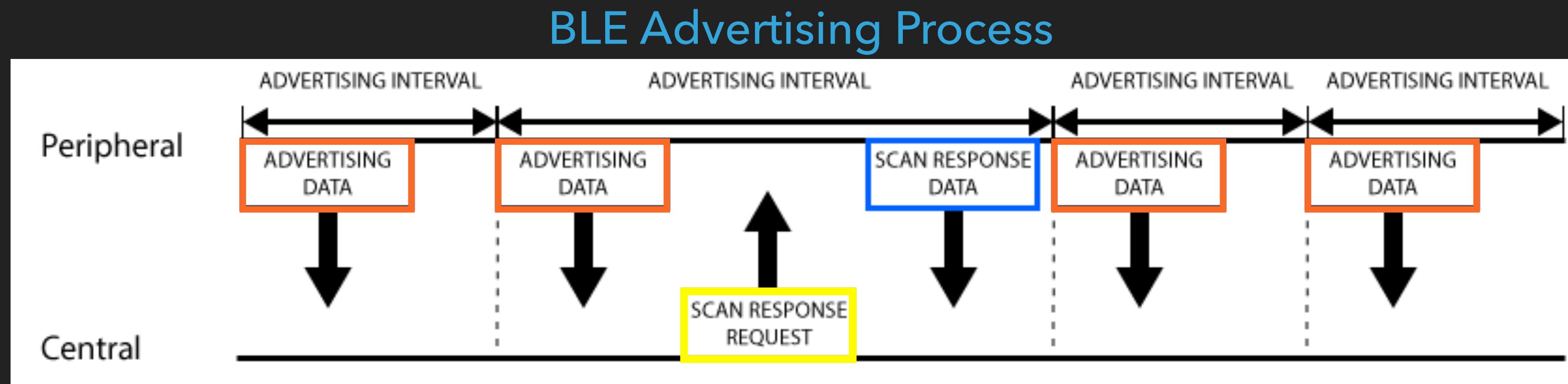
1. **Advertising Data** payload

This is mandatory. Constantly transmitted out to tell **central** devices "hey I'm here" (iBeacon)

2. **Scan Response** payload

Optional secondary payload used to send more advertising data (device name, etc.)

- ▶ Both can contain up to 31 bytes of data

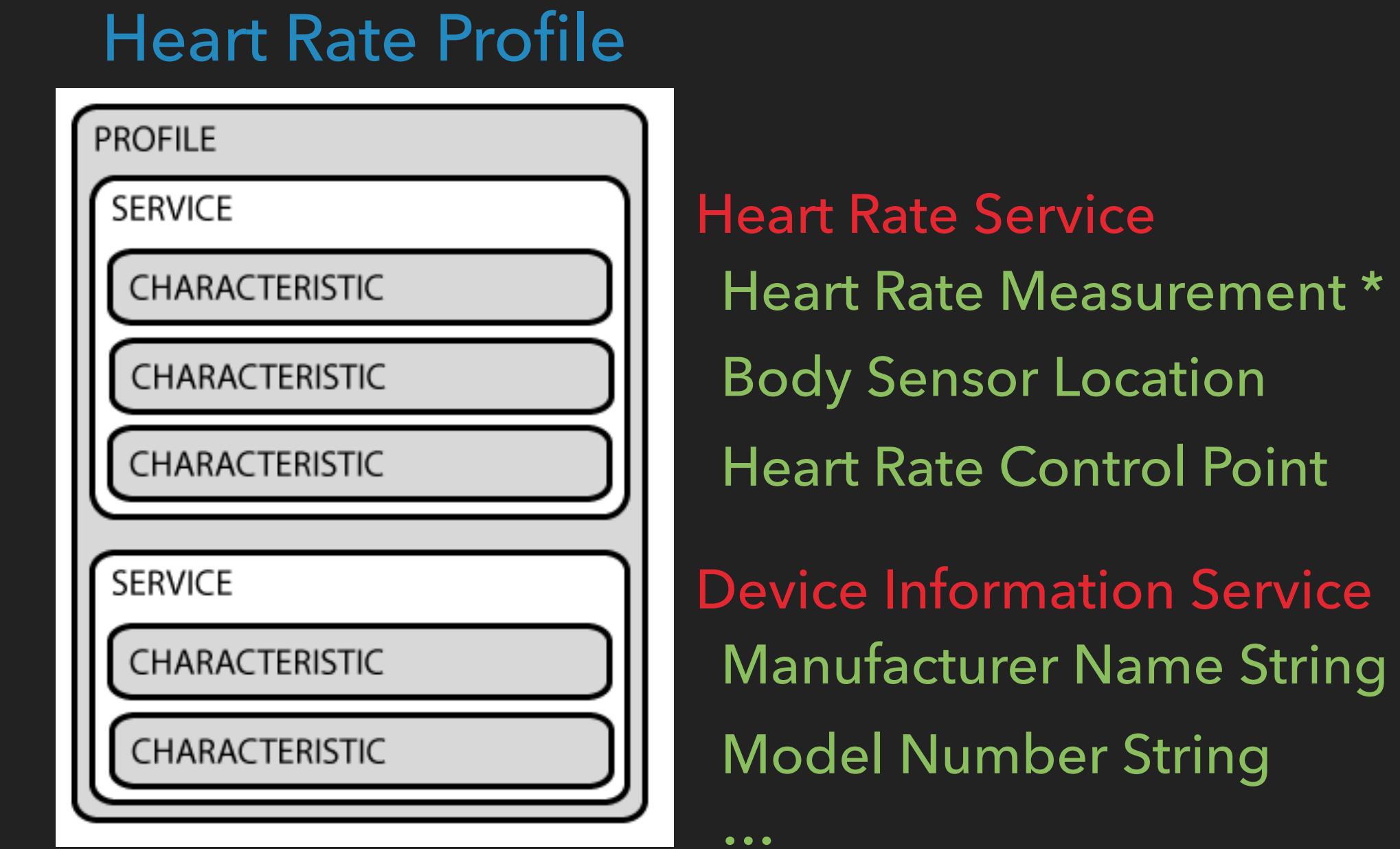


- ▶ Once a connection is made (peripheral to central), generally the advertising process will stop and you won't be able to send advertisement packets anymore. Instead...

# BLE BASICS: GATT

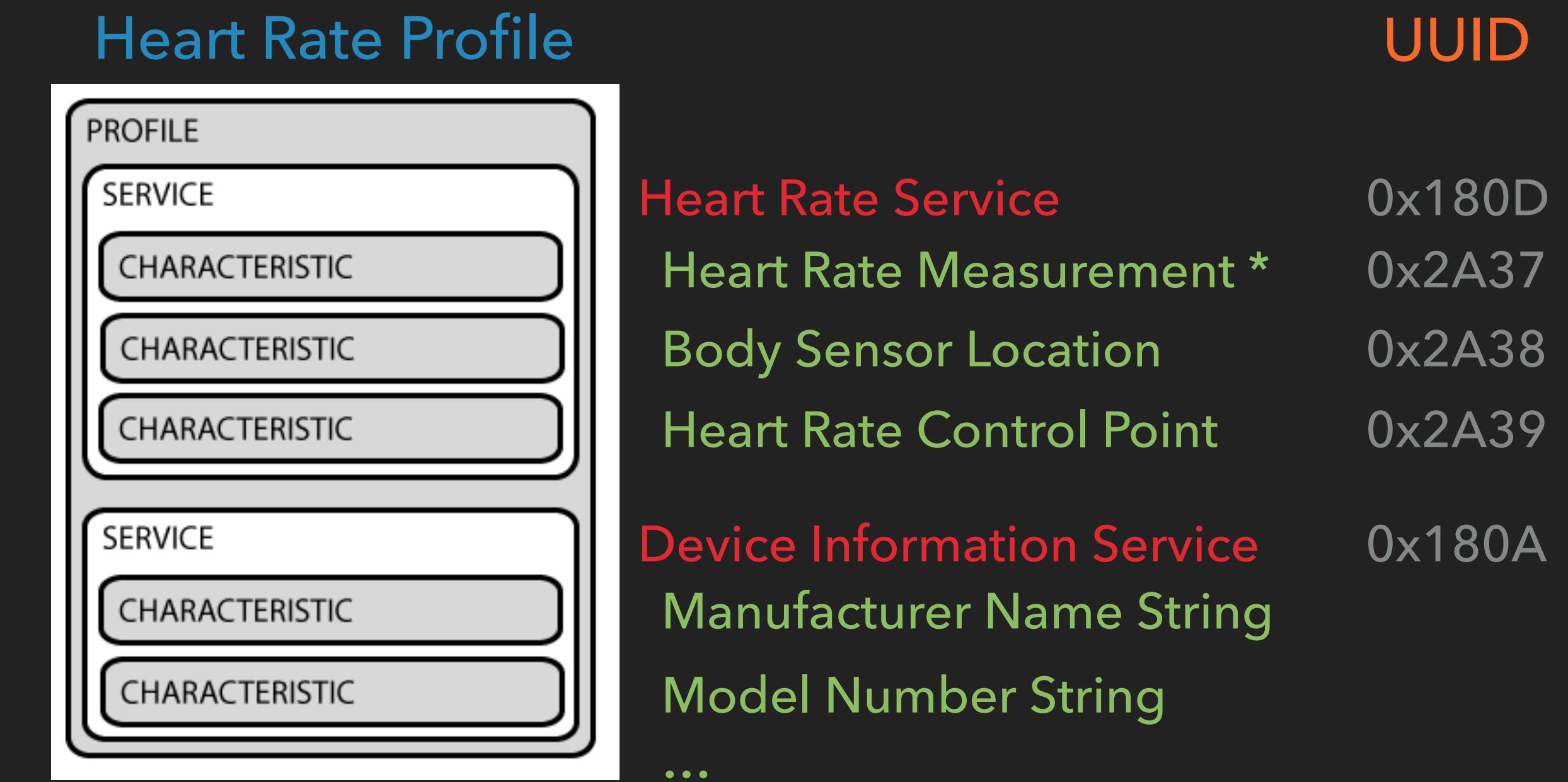
---

- ▶ You'll communicate back and forth with GATT (Generic ATTribute Profile)
- ▶ GATT Profile specifies the structure in which data is exchanged:
  - ▶ **Profile**: a predefined set of **Services** (one or more). [Link](#)
  - ▶ **Service**: logical grouping of chunks of data called **Characteristics**
  - ▶ **Characteristic**: each contains a value (single data point)



# BLE BASICS: GATT

- ▶ UUIDs
  - ▶ **Service & Characteristic:** 16-bit (for officially adopted), or 128-bit (custom)
  - ▶ Use the adopted ones: interoperability across BLE-enabled HW/SW
  - ▶ SIG: [Services, Characteristics](#)



# HANDY BLE TOOLS

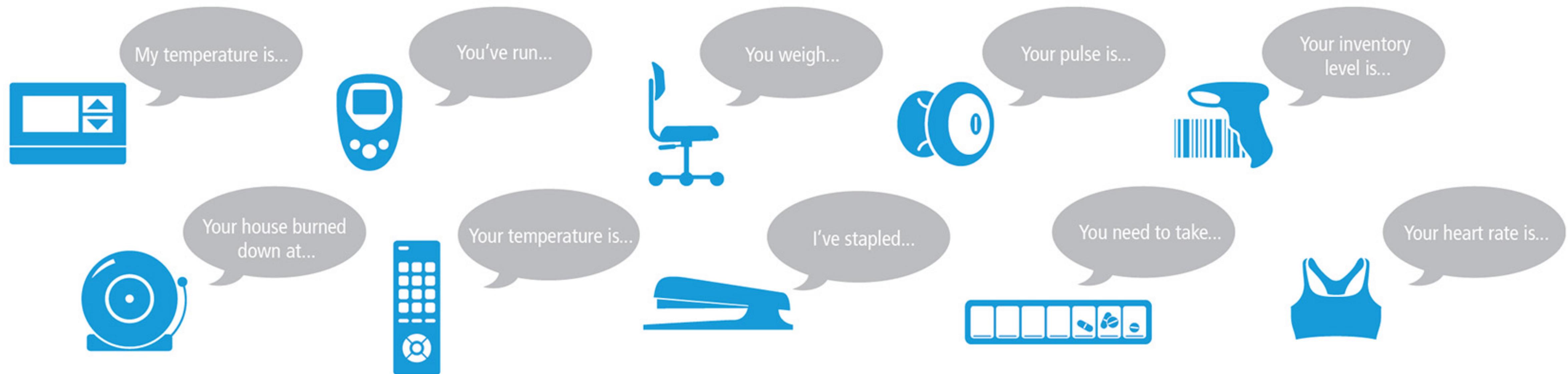
- ▶ Light Blue, by Punch Through
- ▶ MacOS, Windows, iOS



MacOS

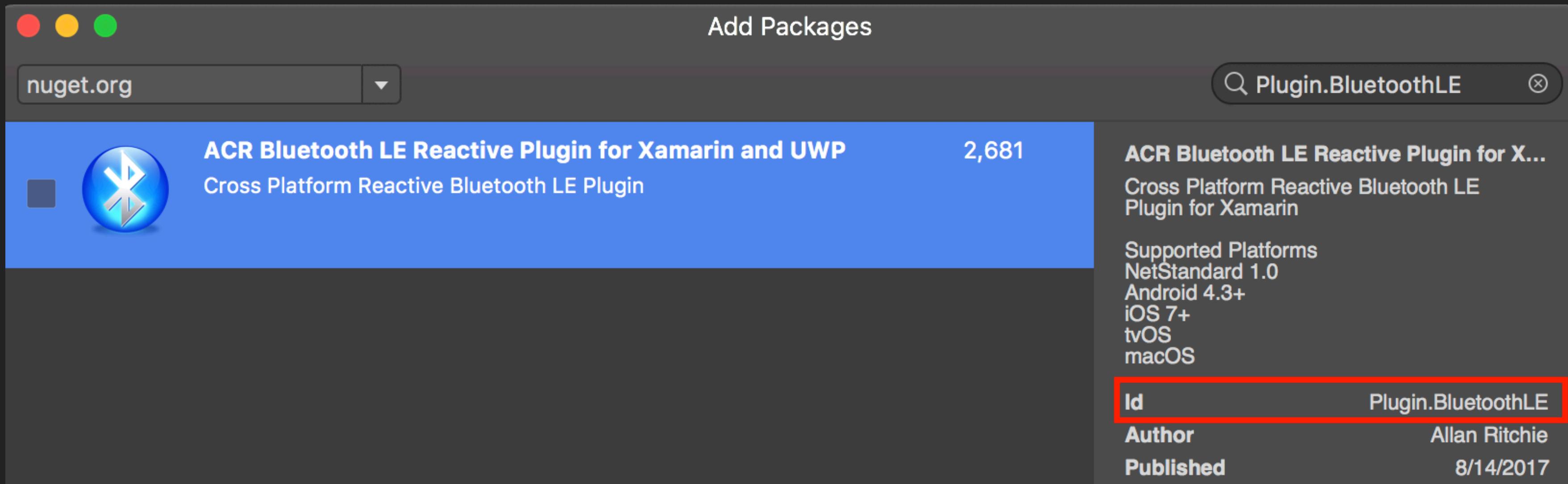


iOS



# BUILDING A BLE XAMARIN APP

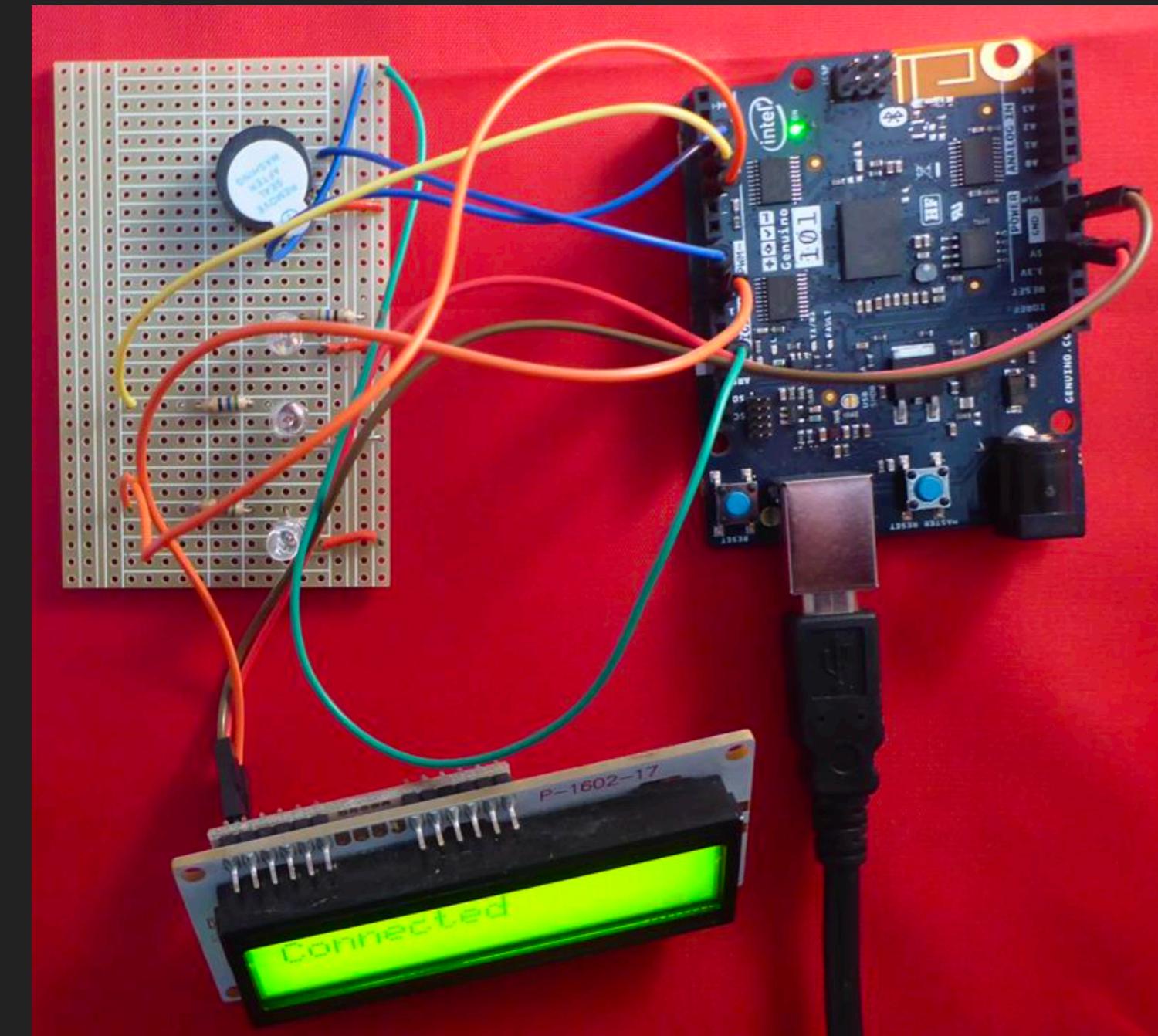
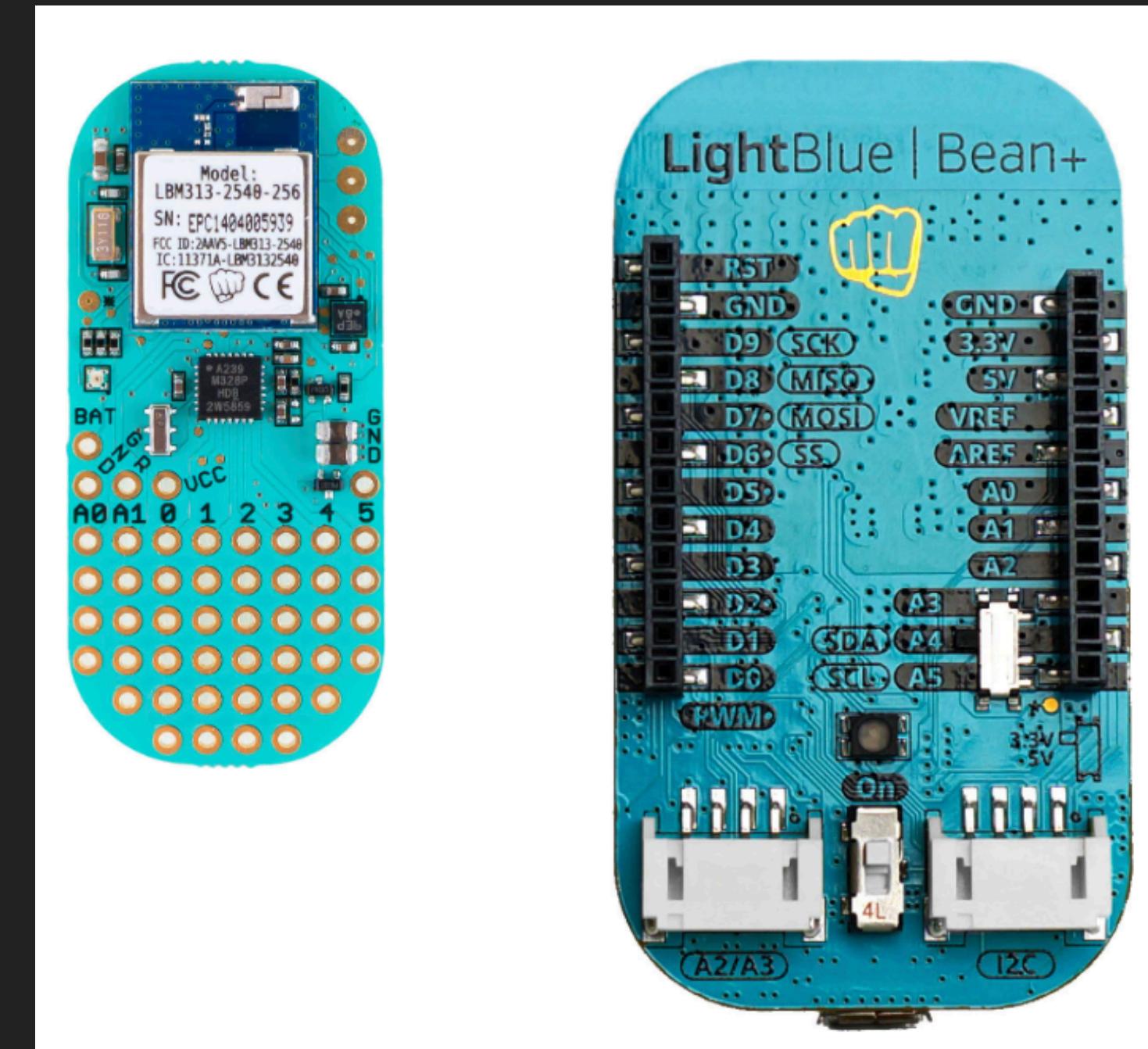
- ▶ Pretty good cross-platform library: **ACR Reactive BluetoothLE Plugin for Xamarin**
  - ▶ Uses the Reactive Extensions for .NET <https://github.com/reactive-extensions/rx.net>
  - ▶ Library by Microsoft that extends LINQ Queries to apply to asynchronous events and callbacks
- ▶ Search for “Plugin.BluetoothLE”



- ▶ Use up to version 3.1.1 of the plugin if you’re using traditional PCL profiles (not .NET Standard)

# DESIGNING YOUR OWN BLE PERIPHERALS

- ▶ Kits available
  - ▶ SIG: Bluetooth Developer Starter Kit + Arduino 101 board
- ▶ **bleno**: A Node.js module for implementing BLE peripherals
  - ▶ <https://github.com/sandeepmistry/bleno>
- ▶ Demo on Mac



# RESOURCES

---

- ▶ Intro to BLE

<https://learn.adafruit.com/introduction-to-bluetooth-low-energy/introduction>

- ▶ Bluetooth SIG

<https://www.bluetooth.com>

- ▶ ACR Reactive BluetoothLE Plugin for Xamarin and Windows

<https://github.com/aritchie/bluetoothle>

- ▶ Blog Post on Reactive Extensions

<https://blog.xamarin.com/why-every-mobile-developer-should-use-reactive-programming/>

- ▶ bleno, echo example project

<https://github.com/sandeepmistry/bleno/tree/master/examples/echo>

- ▶ SIG: Bluetooth Development Starter Kit

<https://www.bluetooth.com/develop-with-bluetooth/developer-resources-tools/developer-kits/bluetooth-starter-kit>

- ▶ LightBlue Bean BLE development board

<https://punchthrough.com/bean>

- ▶ iTAG bluetooth 4.0 peripheral (used in my demos)

<https://www.amazon.com/Bluetooth-Wireless-Anti-lost-Compatible-Smartphones/dp/B017BDPMYC>