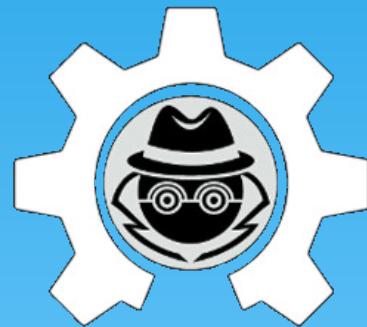


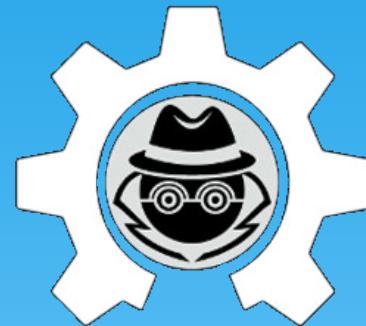
PatternAgents

Open Source Design Patterns for Electronic Systems



Tom Moxon
@PatternAgents

Beacons for Bees



*Applying Geo-Location to
Wild and Domestic Bee Hives

Agenda

- Honey Bees / Colony Collapse Disorder
- Hardware for Beehives
- The Physical Web
- Beacons
- Proximity Beacon API
- Hardware for Beacons
- Beacon Reference Designs

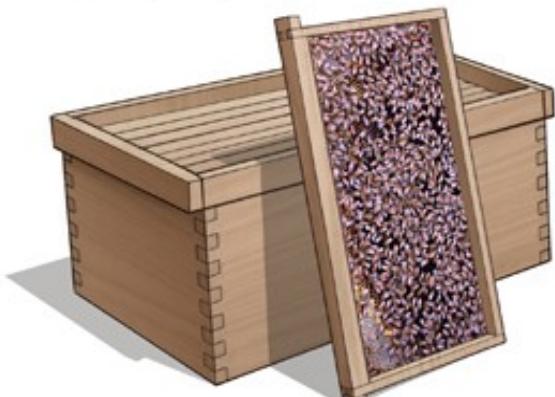
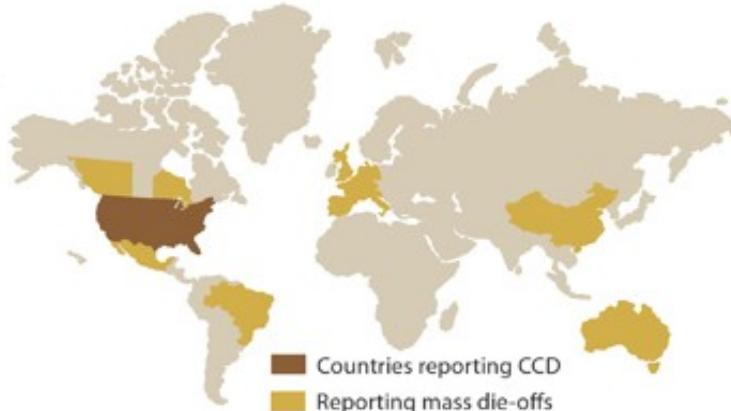


Honey Bee CCD

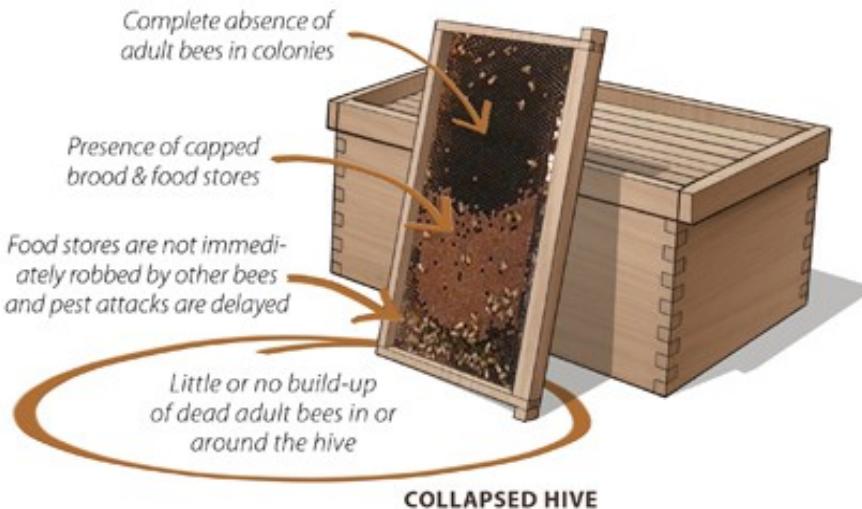
"Never in 40 years had I witnessed the symptoms I was seeing."

- DR. DEWEY CARON, UNIVERSITY OF DELAWARE

What is Colony Collapse Disorder? It's a phenomenon in which worker bees from a hive or colony abruptly disappear. "Stress, immunocompromise, and unusually serious infections by common pathogens and/or otherwise and usually benign organisms, seem to have combined to produce this devastating condition, CCD" (Kevan, 2007).



HEALTHY HIVE



Graphics by Tony Linka

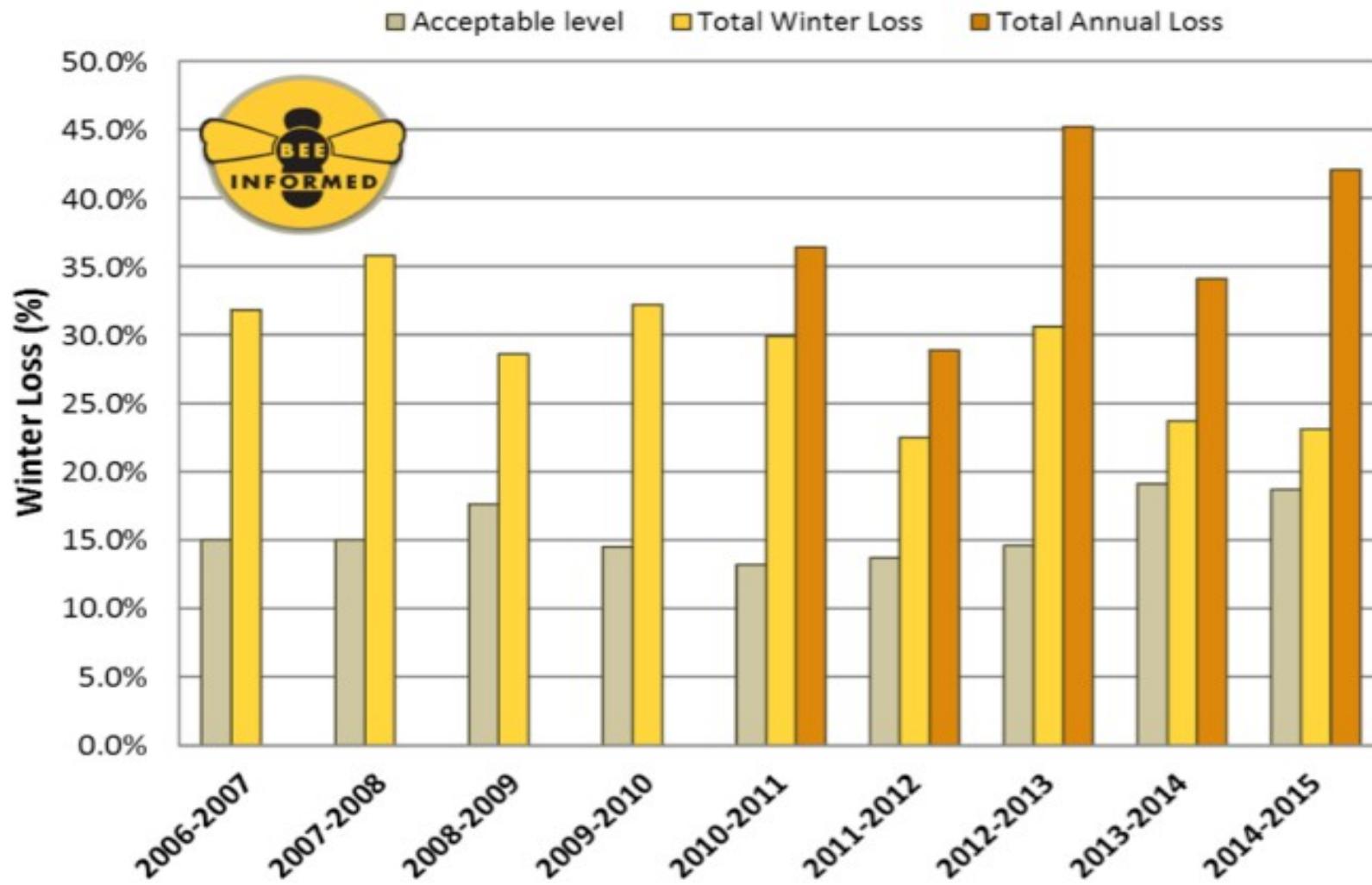
Suspected CCD Causes

There are many causes that have been attributed to Honey Bee Colony Collapse Disorder (CCD), including :

- Wild Habit Destruction
- Mono Crop (single crops lack diverse food supplies)
- Neonicotinoids and general pesticide overuse
- Mites and other parasites
- Fungal, Viral, or other infections (Nosema, IAPV)
- General Climate Change Issues

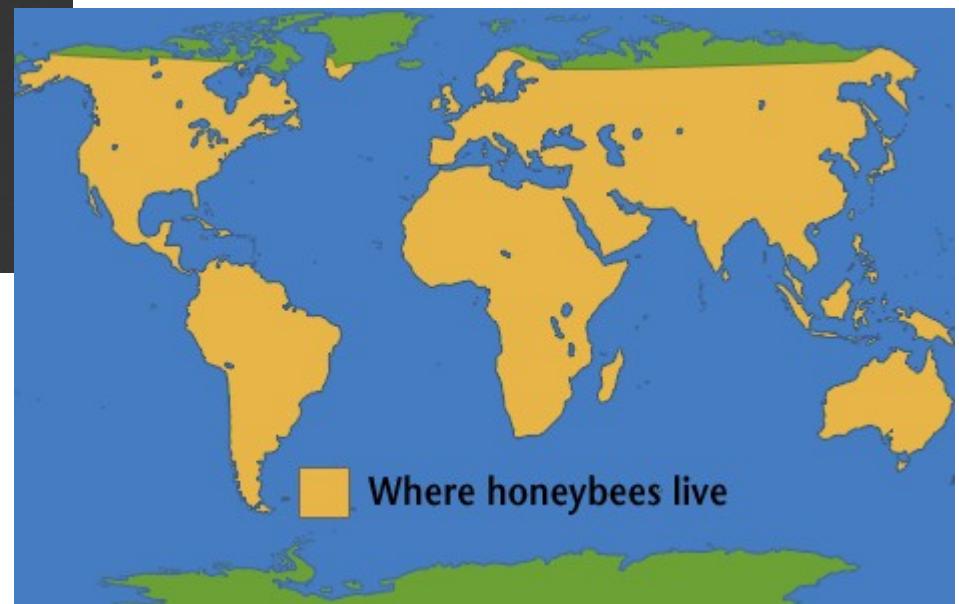
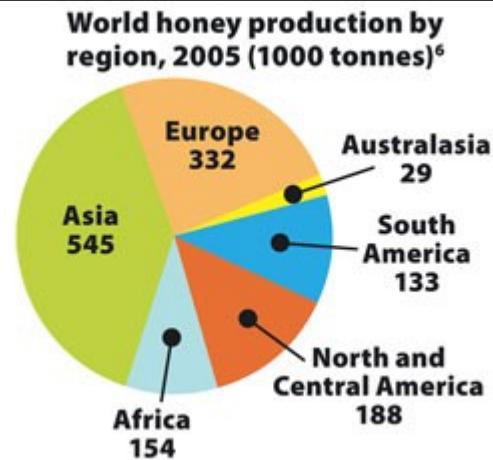
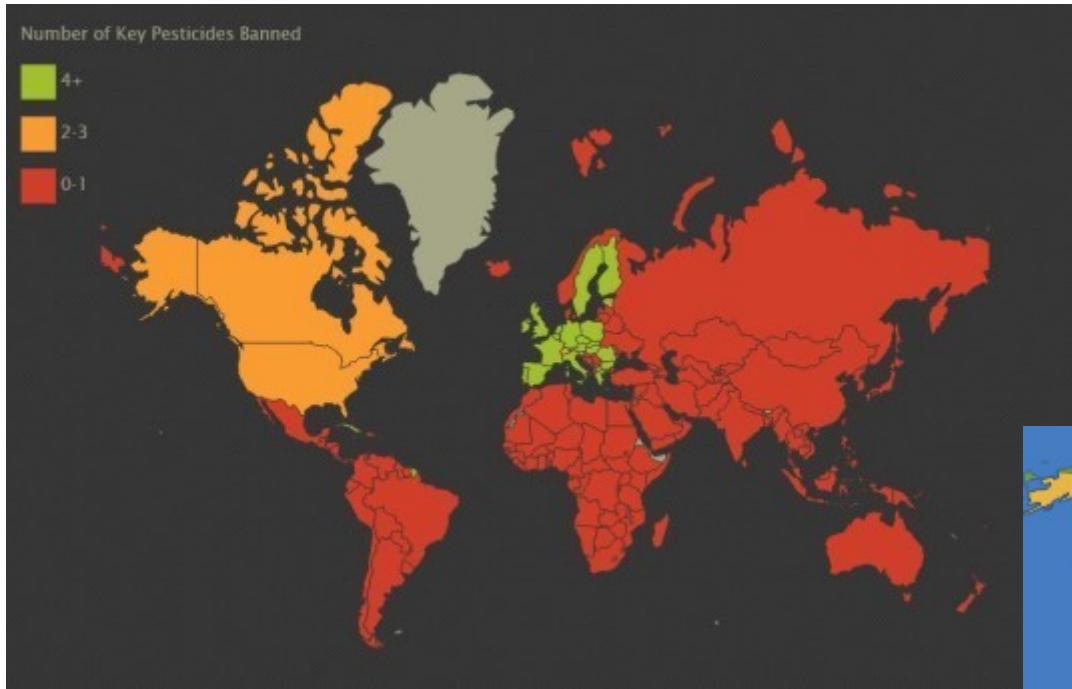
U.S. Honey Bee Surveys

Total US managed honey bee colonies Loss Estimates



Source: USDA

Worldwide Impact



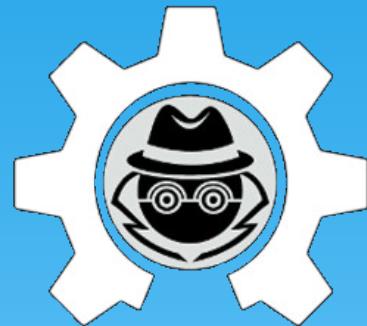
Source: National Geographic

How can we help the Bees?



- Plant Bee Friendly Wildflowers (increase biodiversity)
- Reduce your use of pesticides (companion crops, etc.)
- Raise Awareness of the plight of Honeybees
- Consider your own backyard Hive
- Locate, measure, and monitor wild Hives (little data)

Hardware for Bees

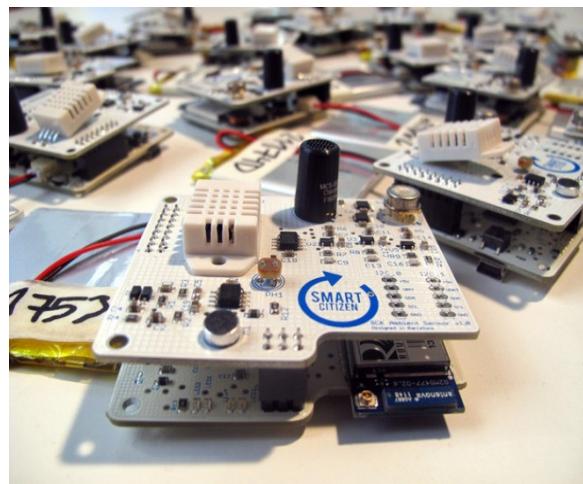


*Working with Hives and Bees

Open Source Beehives



Colorado Top Bar Hive



Smart Citizen Sensors



Barcelona Warre Hive

- * Efforts like Open Source Beehives have been providing easy to build kits and sensors packages for DIY beekeepers

Source: OpenSourceBeehives

Materials for Hives

- * Bees are very picky about what goes into the hive
- * Organic materials (wood, bamboo, paper, etc.)
- * FR4 (printed circuit board materials) and conformal coating of FR4 are not welcomed by the bees (out gassing and aging of materials is a problem)
- * 3-D Printing should use undyed, Poly Lactic Acid (PLA) materials, a corn based, biodegradable, plastic that the bees will generally accept into the hive

Maintenance for Hives

- * Bees may coat circuits, wires, and sensors with propolis, if they don't like the materials
- * If you can coat circuits, wires, and sensors in bee's wax after final test, then the bees will tend to leave them alone
- * If you can't coat it in bee's wax, consider a laser cut wooden or bamboo box, and seal the box edges and openings with bee's wax
- * Replaceable covers for in-hive sensors, paper/gauze

Local Efforts - Bee Counting

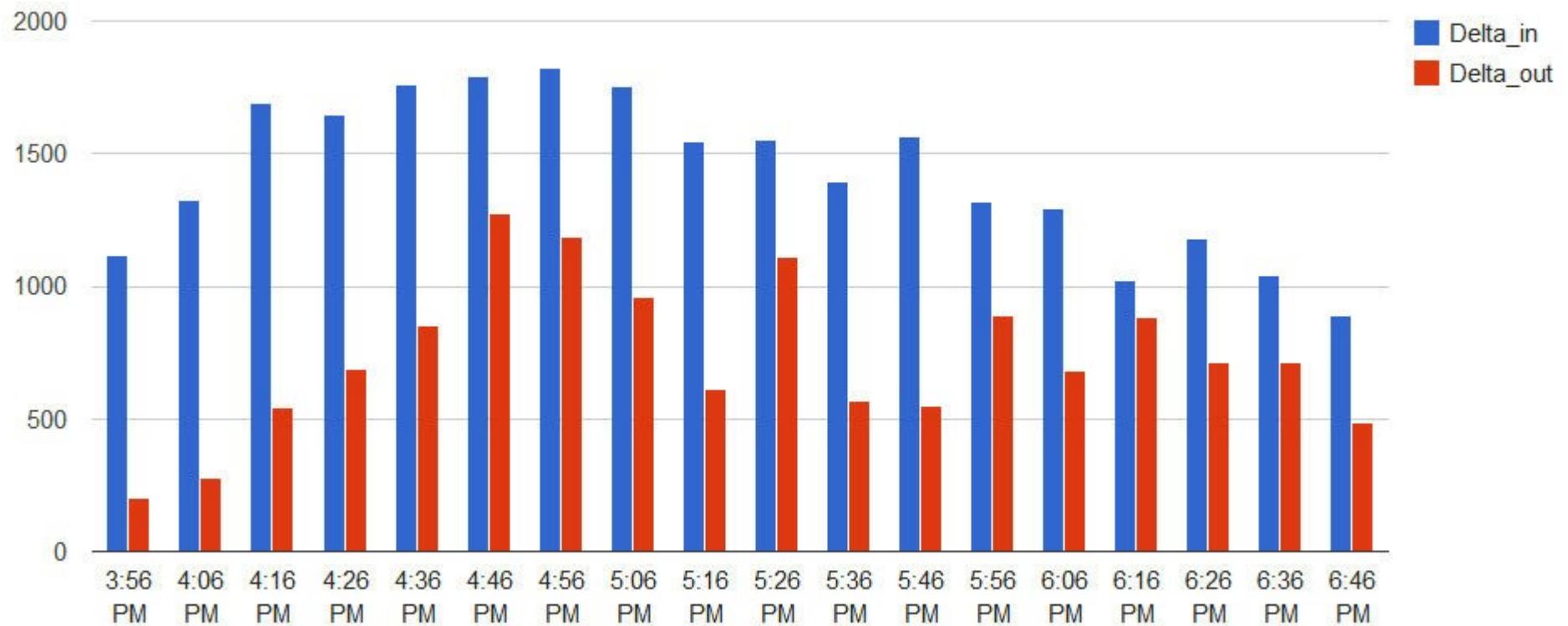
- * Open Source design for optical counter
- * <http://www.instructables.com/id/Honey-Bee-Counter/>



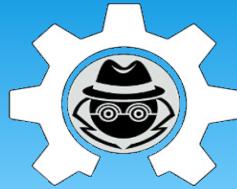
Source: Tom Hudson, PE

Bee Counting

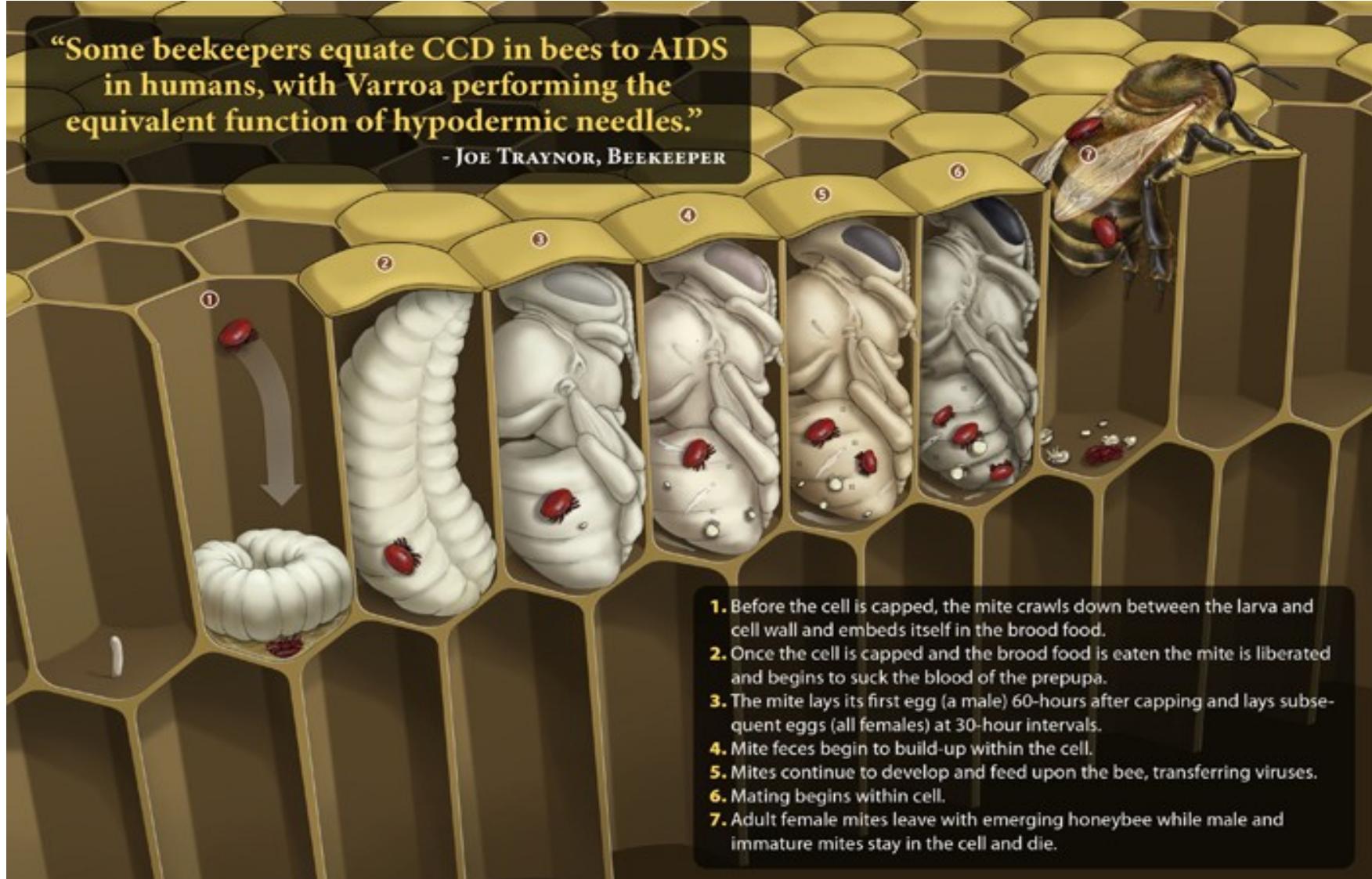
June 21, 2012 - Skidmore Street, 10 minute intervals



Source: Tom Hudson, PE



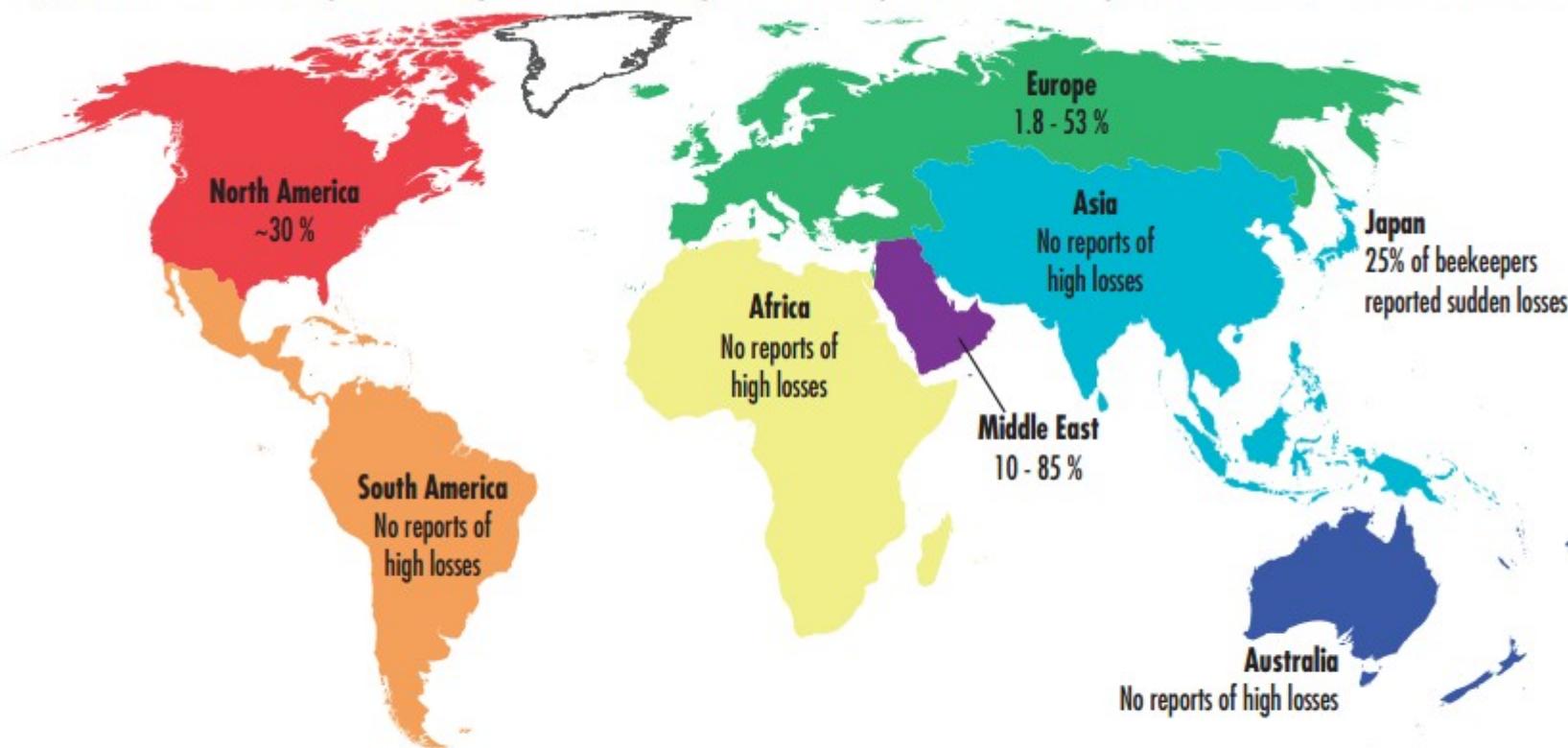
Varroa Mite Infestation



Graphics by Tony Linka

Varroa Mite Losses

Figure 8. Yearly average of managed honey bee colony losses due to Varroa mite



The Varroa mite, *Varroa destructor*, is one of the threats to managed honey bee colonies. Elevated colony losses reported from the USA, Europe, the Middle East and Japan are related to high mite infection³⁹.

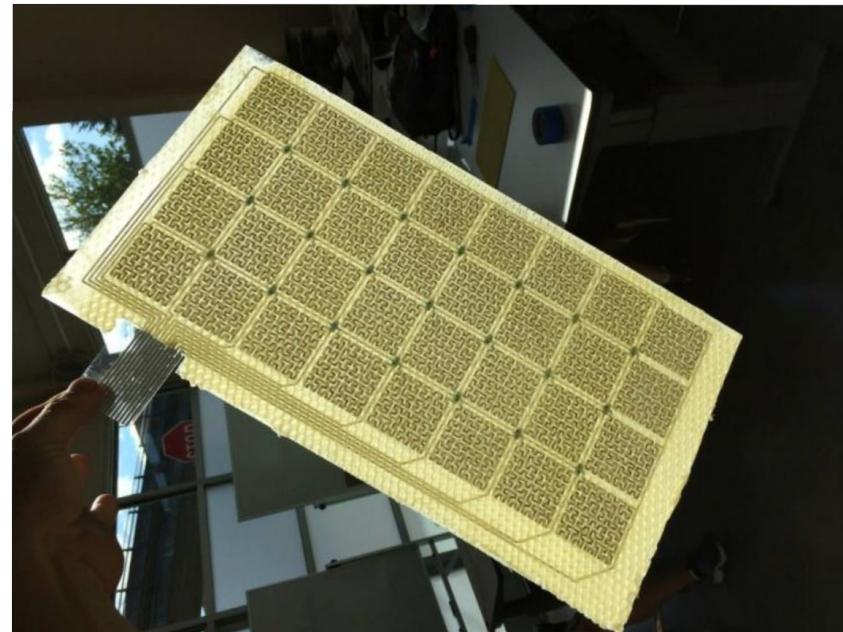
Controlling Varroa Mites

- * Killing the Mites without killing the Bees is difficult
- * Better to interrupt their life-cycle by sterilization
- * Elevated Temperatures in select areas of the hive
- * Existing & patented solutions not working well...
(brute force, difficult to implement, genetic impact)

Circuit Boards for the Bees



- * Eltopia has been developing and testing new control methods for Varroa Mite and other pests



Source: Eltopia

Hive Monitoring Resources

Intelligent Beehives
Open Energy Monitor
HiveTool
Colonymonitoring.com
Hackerbee
Honey Bee Counter
Luke Aldridge Rice
Intelligent Beehives at Yale
Arnia

The Physical Web



- * The physical web is about extending the power of the internet to everyday physical objects and places. The goal is to be able to walk up to any “smart” physical object (like a parking meter, vending machine, etc.) and be able to interact with it without having to first download an “app” (application program).

The Physical Web



- * You've all seen the QR-Code Symbols on posters or stickers, next to a store, on a product display, on a bus stop sign.
- * It holds an “encoded URL” or INTERNET address, using a smartphone camera and application to “SCAN” the QR-Code, and launch the web browser to that address.
- * Can yield a more seamless “physical” to “virtual” user experience

Near Field Communication



- * NFC - Near Field Communication
- * “Tap” to Join/Pair/Exchange



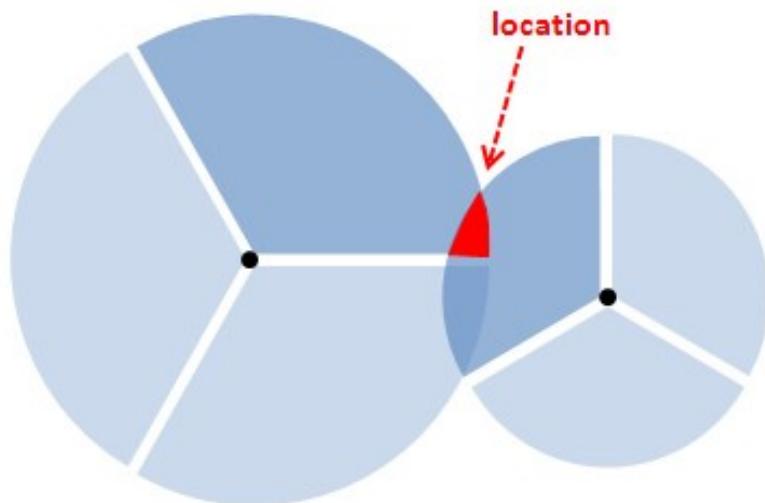
Source: Wikipedia

Global Positioning System



- * Most smart phones include GPS capability these days
- * Uses a LOT of battery power
- * Doesn't work well indoors or under the forest canopy
- * Can give fine grain position (3meters)

Cell Tower Positioning



Cell phone detected within a certain distance of two cell towers with directional antennae.

- * Triangulation/Location
- * Not very accurate
- * Uses cell phone
- * Only where cell towers are...

Indoor Positioning System



- * Uses Wi-Fi signal strength (RSSI)
- * Triangulation from known location of several Wi-Fi access points
- * Popular in Malls, Campus, etc.
- * Needs Wi-Fi radio on (uses power)

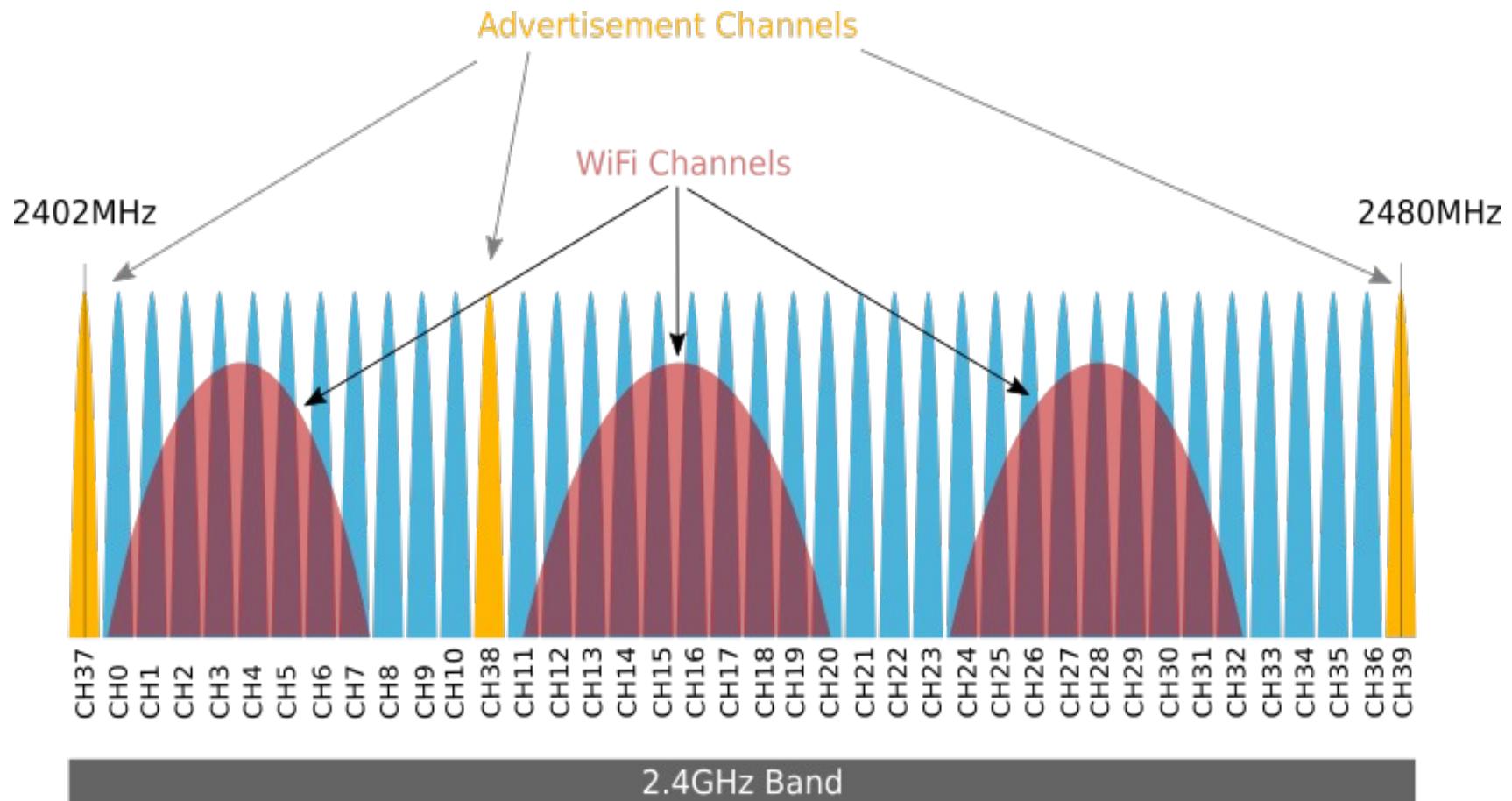
Bluetooth Communication



Low Energy

- * A Wireless Personal Area Network
- * Originally Nokia “Wibree” (2006)
- * Merged into Bluetooth SIG (2010)
- * Very low power transmitter
- * Ubiquitous now...

Bluetooth vs. WiFi



Okay, so what is a Beacon?

Imagine that each Lighthouse could flash its light in a unique pattern that would identify each lighthouse individually...



- * A Bluetooth Beacon functions similarly to a Lighthouse, it transmits information about a location, using a Bluetooth Low Energy radio, programmed to emit a known pattern, or universally unique identifier.

(i.e. “You are near here!”)

Lighthouse Photo by
Dennis Jarvis

Why Beacons for Bees?

- * Raise awareness of pollinator health problems
- * Proximity warning for those allergic to Bees, Wasps, etc.
- * Locate, monitor, and protect Wild Hives
- * New approach to Geo-Tagging wildlife habitat
- * More appropriate technology for remote areas
- * Lower power than existing monitoring solutions
(Wi-Fi uses more power and transmits longer)

What is an iBeacon?



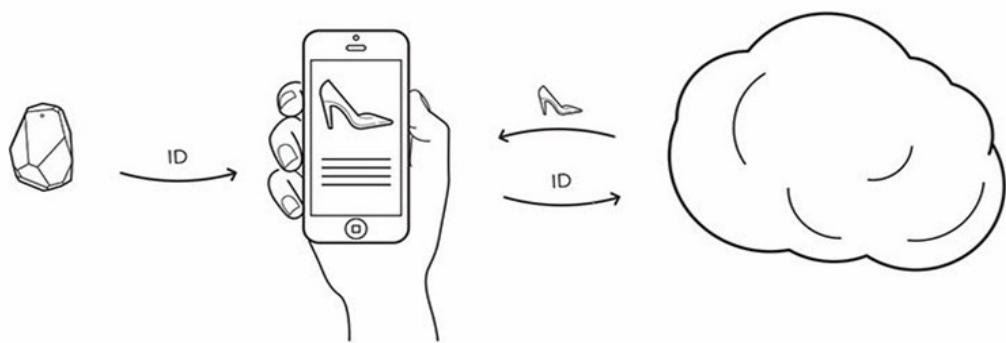
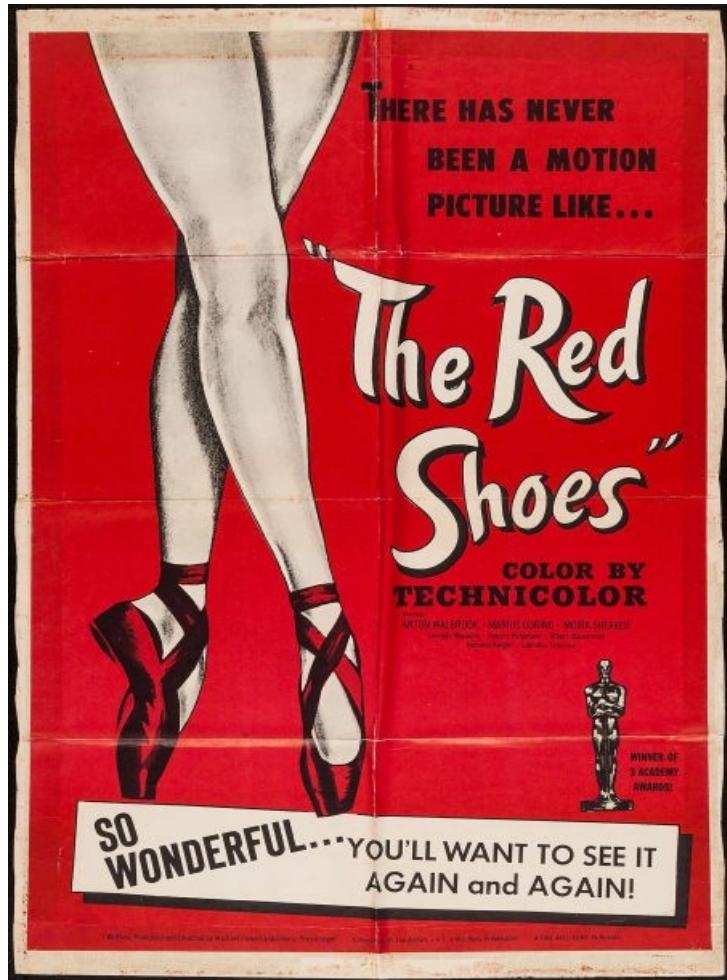
- * A proprietary and licensed Beacon system developed by Apple in 2013 to create location aware services for Apple Pay.
- * iBeacon uses Bluetooth low energy proximity sensing to transmit a universally unique identifier (UUID), which is picked up by a compatible application or operating system.

The unique identifier can be used to determine the device's physical location, track customers, or trigger a location-based action on the device such as a check-in on social media, or a push notification.

First Use Case

- * Originally, Apple envisioned the iBeacons to be run on iPad Tablets used as Point-of-Sale Systems.
- * When you are close enough to the Point-of-Sale System, a coupon or membership offer might appear in the application (“the app”) running on the users iPhone
- * Didn't need any dedicated hardware, as the Bluetooth Radios were already present in the iPhones and iPad tablets

How iBeacons Work



Source: estimote.com

iBeacon - First

- * First Beacon Specification/Most widely installed base
- * Requires the Implementer to develop an “App”
- * Requires the User to install that “App”
- * Uses Vendor/Manufacturer Specific BT Frame
- * “One Beacon, One App”
- * Hard Codes UUID to Application (Maintenance...)

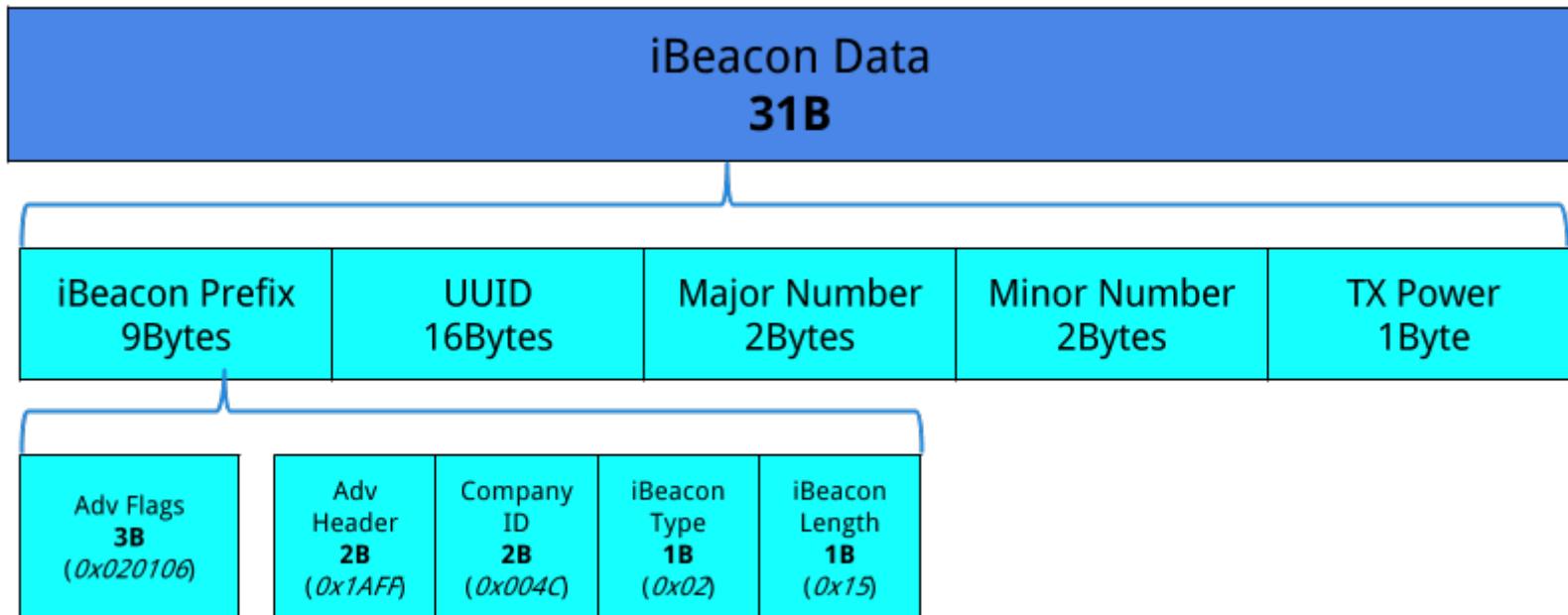
iBeacon - Apple Pay

- * Can be very costly to implement and maintain
- * Costs : \$10K typical startup (small event)
 - : > \$250-\$500K (large event)
 - : Apple Passbook can be less expensive
- * Users : Apple, Sephora, Dunkin' Donuts,



Source: Apple Computer

Advertising Packet iBeacon



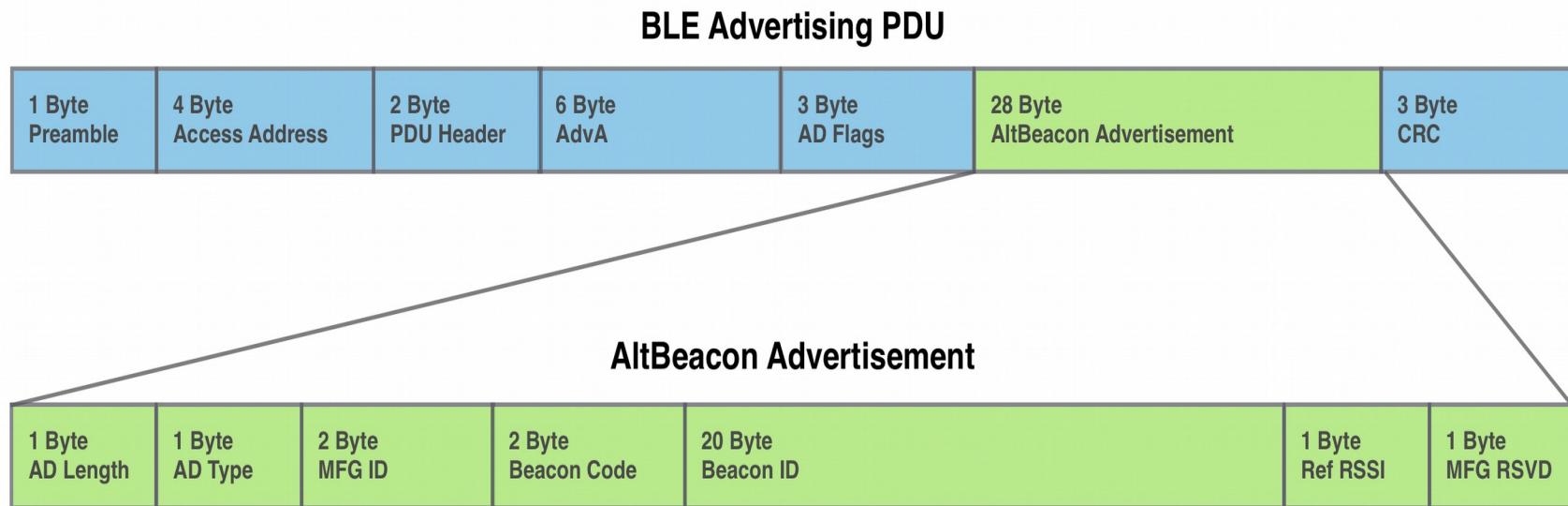
- * Custom Prefix (Apple Specific) 10 Packets per Second (100ms)
- * (UUID) Universal Unique Identifier is an assigned number and needs to be interpreted by the application to be useful
- * Only 31 Bytes!

What is an AltBeacon?



- * Developed as an Open Source alternative to Apple's closed and proprietary iBeacon technology.
- * Specification by Radius Networks
- * Adds Manufacturer ID, Beacon Code, Reserved Code
- * Started gaining steam until EddyStone Beacons were introduced, now loosing share to EddyStone

AltBeacon Packet



- * Interesting Use Cases with Manufacturer ID, and MFG reserved codes
- * Limited installed base

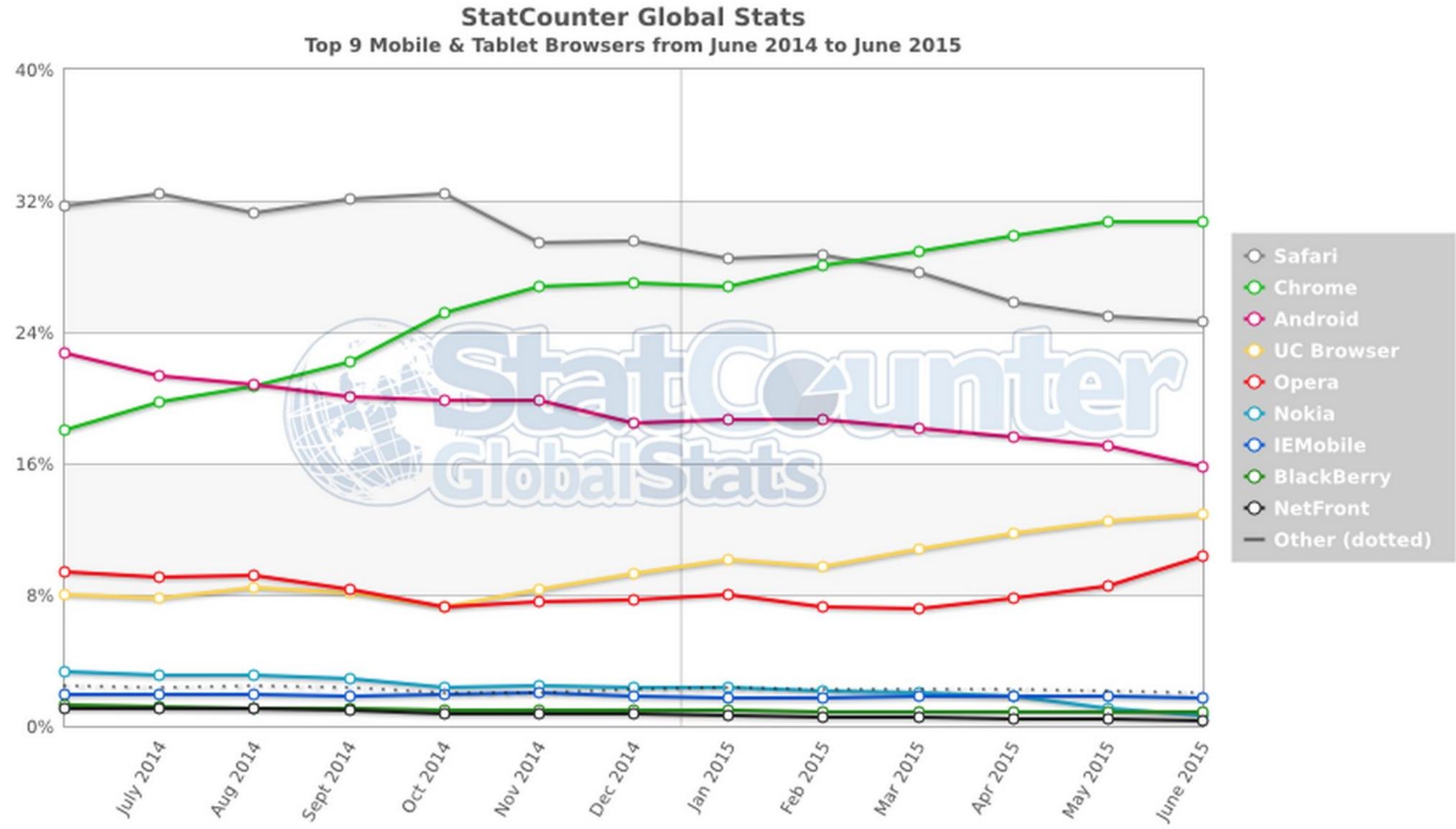
What is Eddystone?



Eddystone

- * Eddystone is an open beacon format developed by Google and designed with transparency and robustness in mind
- * Eddystone can be detected by both Android and iOS devices
- * Eddystone doesn't require a dedicated “app” to function, as it can deliver a target URL directly
- * Google Chrome **IS** the “app”!

Chrome App Share



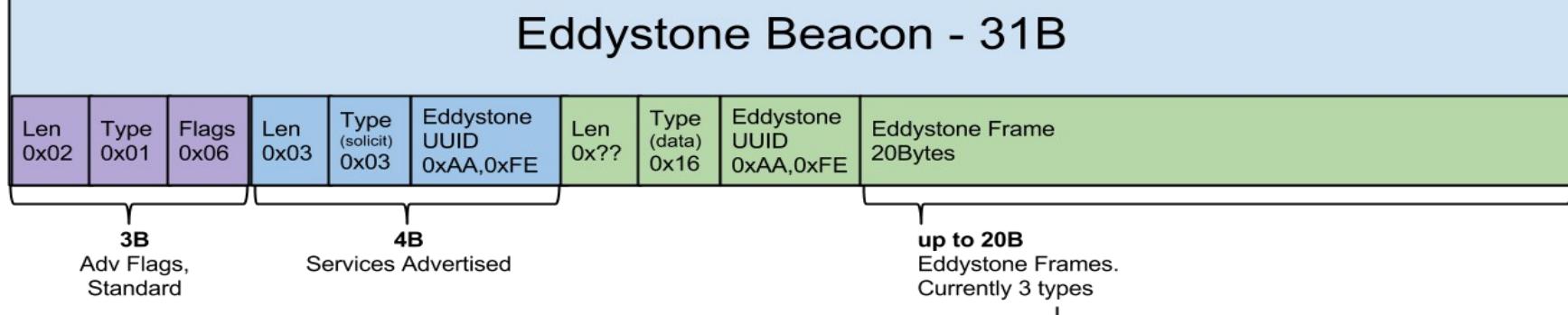
Source: StatCounter

Eddystone Frame Types

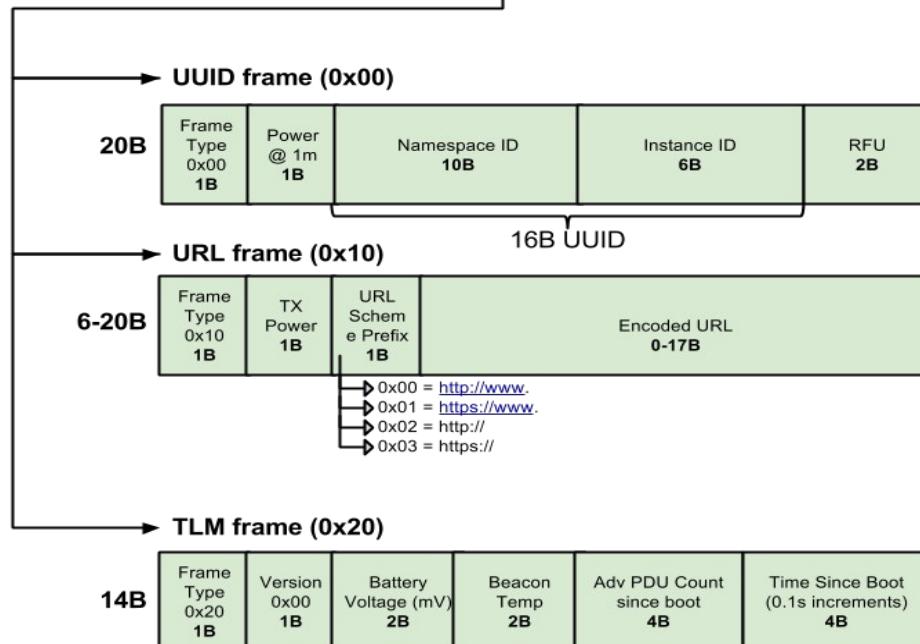
- * **Eddystone-URL** : delivers a “short” URL directly to the user when they are in range
- * **Eddystone-UID** : delivers a UUID, similar to how iBeacon works, but 16 bytes in length
- * **Eddystone-TLM** : delivers Beacon Telemetry data for Beacon management (battery level, uptime, temperature, etc.)

Eddystone Packet

Eddystone Beacon - 31B



- Uses Service Frames
- 1-2 Packets per Second
- 10X less than iBeacon
- Much longer battery life
- Includes Telemetry packet
- No Licensing Required



Source: Google

Eddystone-UUID

- * Functions similar to iBeacon, AltBeacon (app required)
- * Beacon UUID is registered with Google Beacon Dashboard
- * Available for Google Developer Project once registered
- * Open Standard, no licensing, fees, Apple taxes...
- * Use free “Beacon Tools” app to register beacons
- * Almost NO barrier to entry (low cost, open source, etc.)
- * “One Beacon, Many Apps...”

Eddystone-UUID

- * Recommended Practice
- * Use a SHA1 Hash of your company FQDN

sha1(www.patternagents.com) =
fba4fd8d86e4726632962dfff9f21eafbbe93ed2

- * UUID = fba4fd8d86e472663296
- * Use sub-domains if you need more than one

foo.patternagents.com

Eddystone-URL

- * The Eddystone Beacon provides an encoded URL
- * The URL is 17 bytes long (max), and uses an encoded format (UriBeacon format) :

Prefix

0x00 - http://www.

0x01 - https:/www.

0x02 - http://

0x03 - https://

Suffix

0x00 - .com/

0x01 - .org/

0x02 - .edu/

0x03 - .net/

- * Works great with a URL shortener, like bit.ly, git.io, etc.
- * Must be an “https://” URL for Google to index it...

Eddystone-URL

- * Eddystone-URL beacons are NOT registered, unless they also transmit UUID frames (which ARE registered)
- * Any HTTPS address can be used and will give a pop-up notification, if notifications are enabled
- * Gives control back to the user/implementer, you can direct the user to any valid, secure URL and they can opt-in/visit
- * No tracking generally, unless the user has a login/account with the referred URL, can be redirected to user page...

Eddystone-URL

- * IMHO, the costs associated with developing apps for iBeacon and AltBeacon are prohibitive unless your company (or application/event/etc.) is large enough to warrant it
- * Eddystone-URL is an equalizer here, it is a much lower cost solution to implement and deploy, and there are no application development costs or Apple licensing costs to incur.
- * Eddystone-URL makes the design, and deployment of location aware services, faster, easier, and less expensive

Eddystone TLM

- * Telemetry Frame
- * Great for Beacon fleet management
- * Battery Level, Temperature, Packet Count, etc.
- * Interleaved with other frames occasionally
- * Can be uploaded to cloud via your “app”
(is it stealing my data? Paranoia runs deep...)

Eddystone EID

- * Empheral ID, changes every few minutes
- * A secure key is exchanged during beacon registry
- * Meant to be more secure, not easily spoofed
- * Useful for wearable, medical data devices

Google Beacon Tools App

* Install from Google Play Store

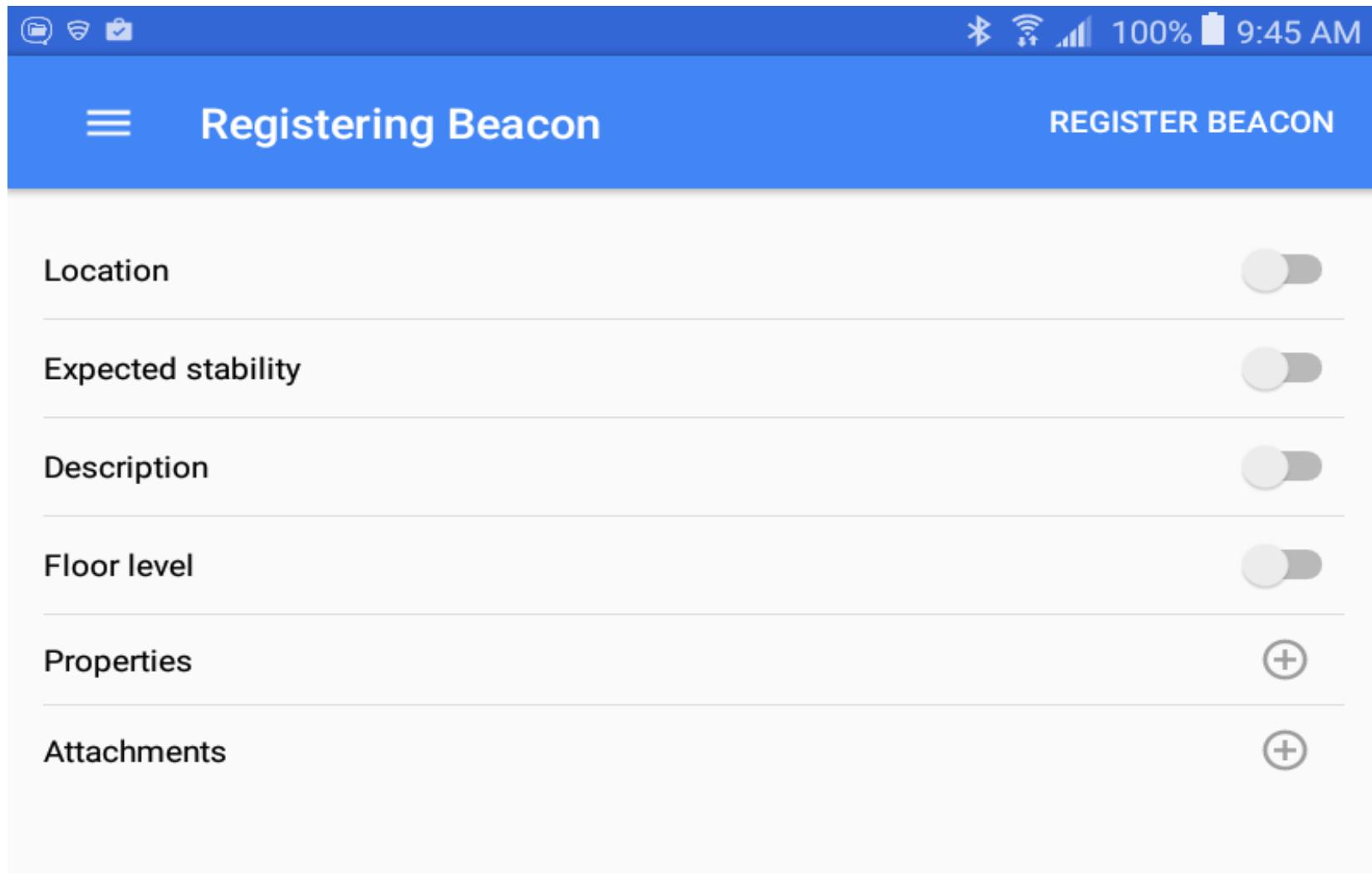


The screenshot displays the "Beacons Near Me" screen of the Google Beacon Tools app. At the top, there are tabs for "UNREGISTERED" and "REGISTERED". Below these tabs, the "Eddystone-URL" section is shown, listing two entries: "bit.ly/2b4edZk" with Tx: -6 dBm and Rx: [blue progress bar], and another entry with the same URL and Tx: -47 dBm and Rx: [blue progress bar]. Below this, the "Eddystone-UID" section is shown, displaying the hex code "c4 5b be 11 45 75 ff ae 54 89" and an entry with Tx: -47 dBm and Rx: [blue progress bar]. The top right corner of the screen shows the device's status bar with icons for Bluetooth, Wi-Fi, signal strength, 100% battery, and the time 9:36 AM.

Category	Identifier	Tx (dBm)	Rx (Signal Strength)
Eddystone-URL	bit.ly/2b4edZk	-6	[Blue Bar]
	bit.ly/2b4edZk	-47	[Blue Bar]
Eddystone-UID	c4 5b be 11 45 75 ff ae 54 89	-47	[Blue Bar]

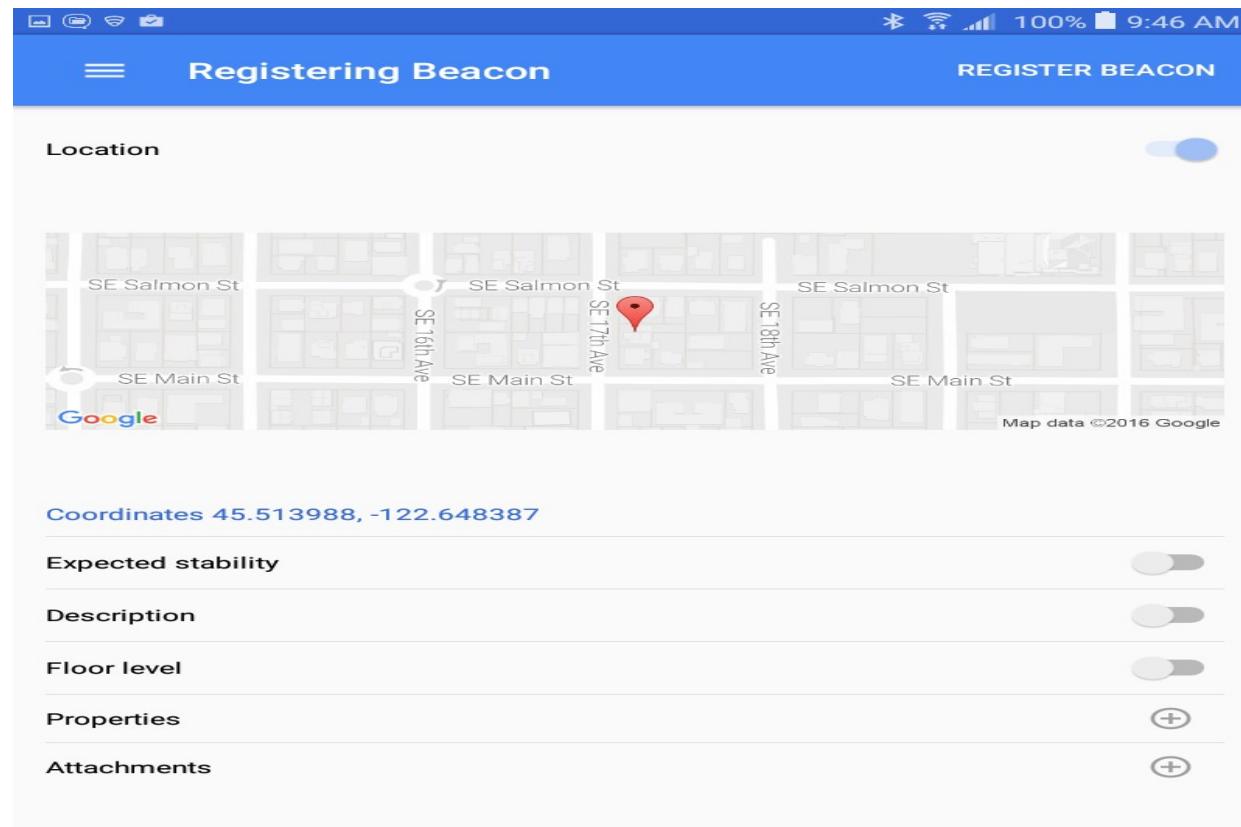
Source: Google

Register Your Beacons



Source: Google

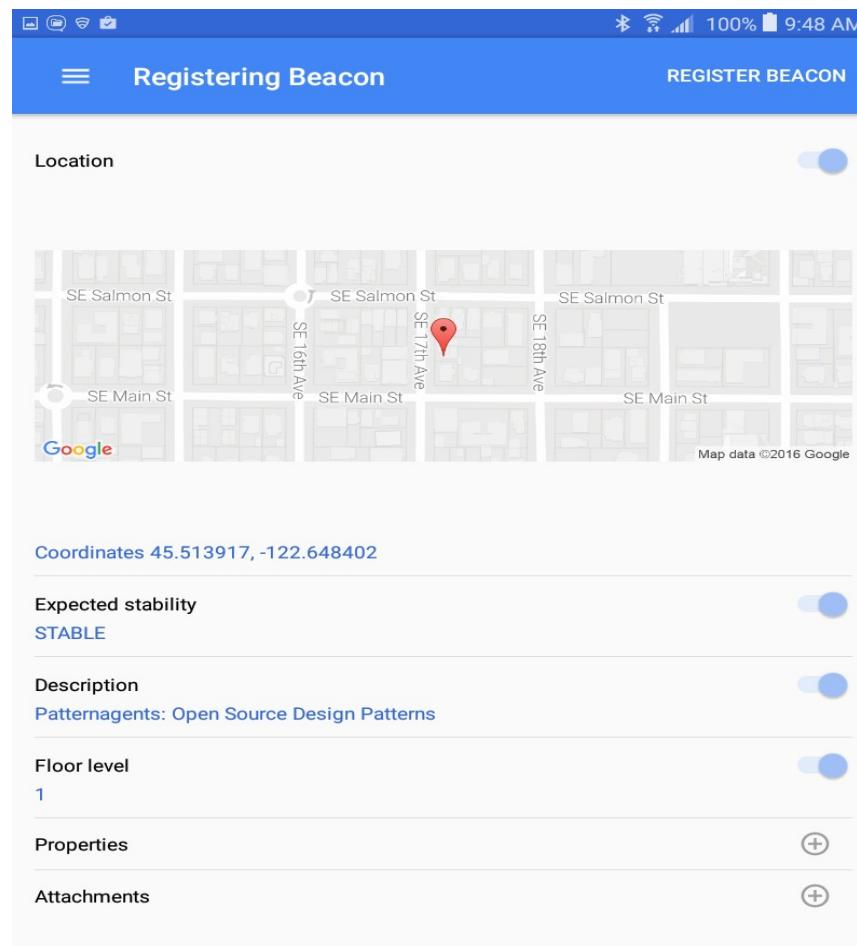
Enter Location



Source: Google

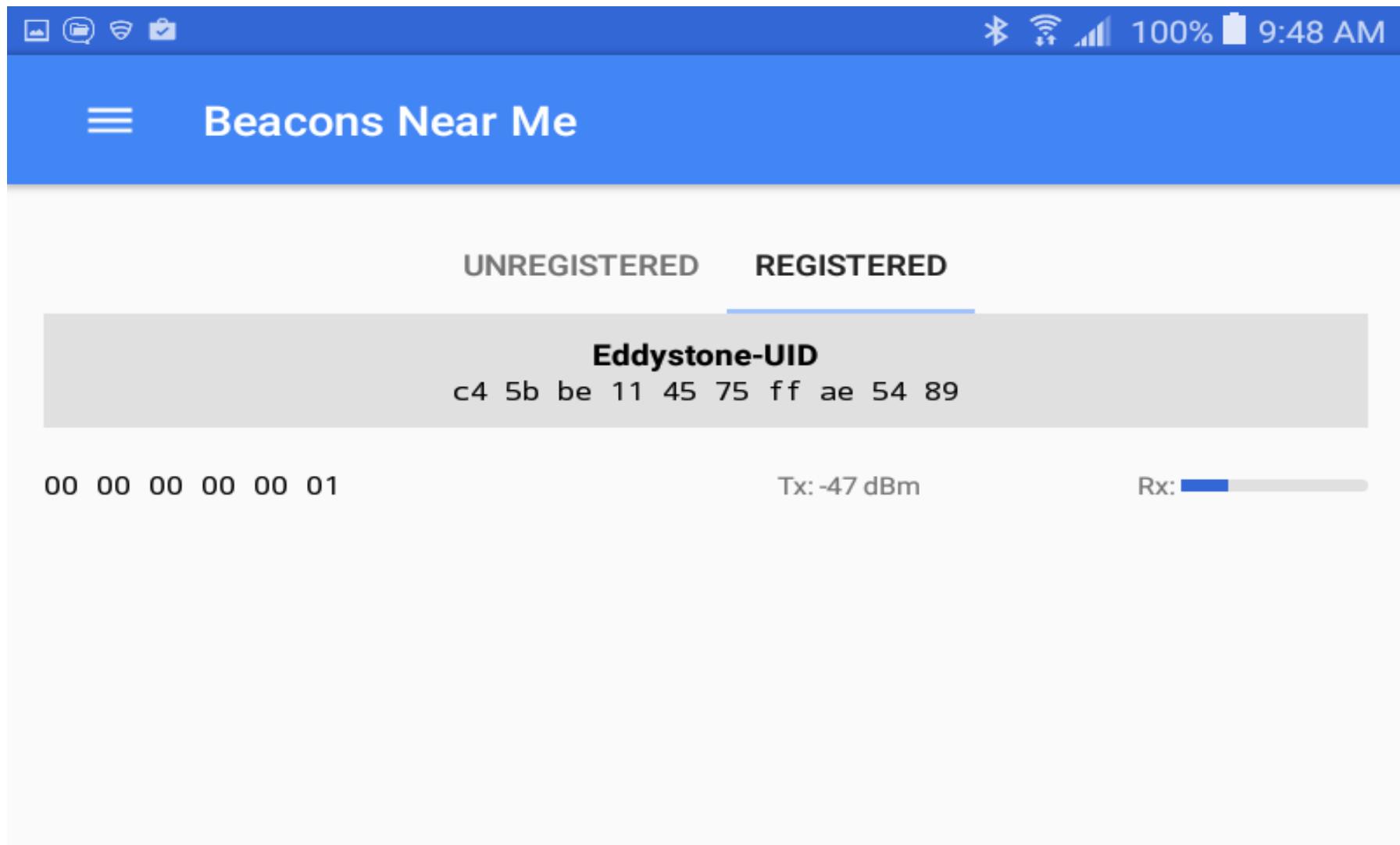
Stability

- * Mark your beacon as stable, moving, etc.



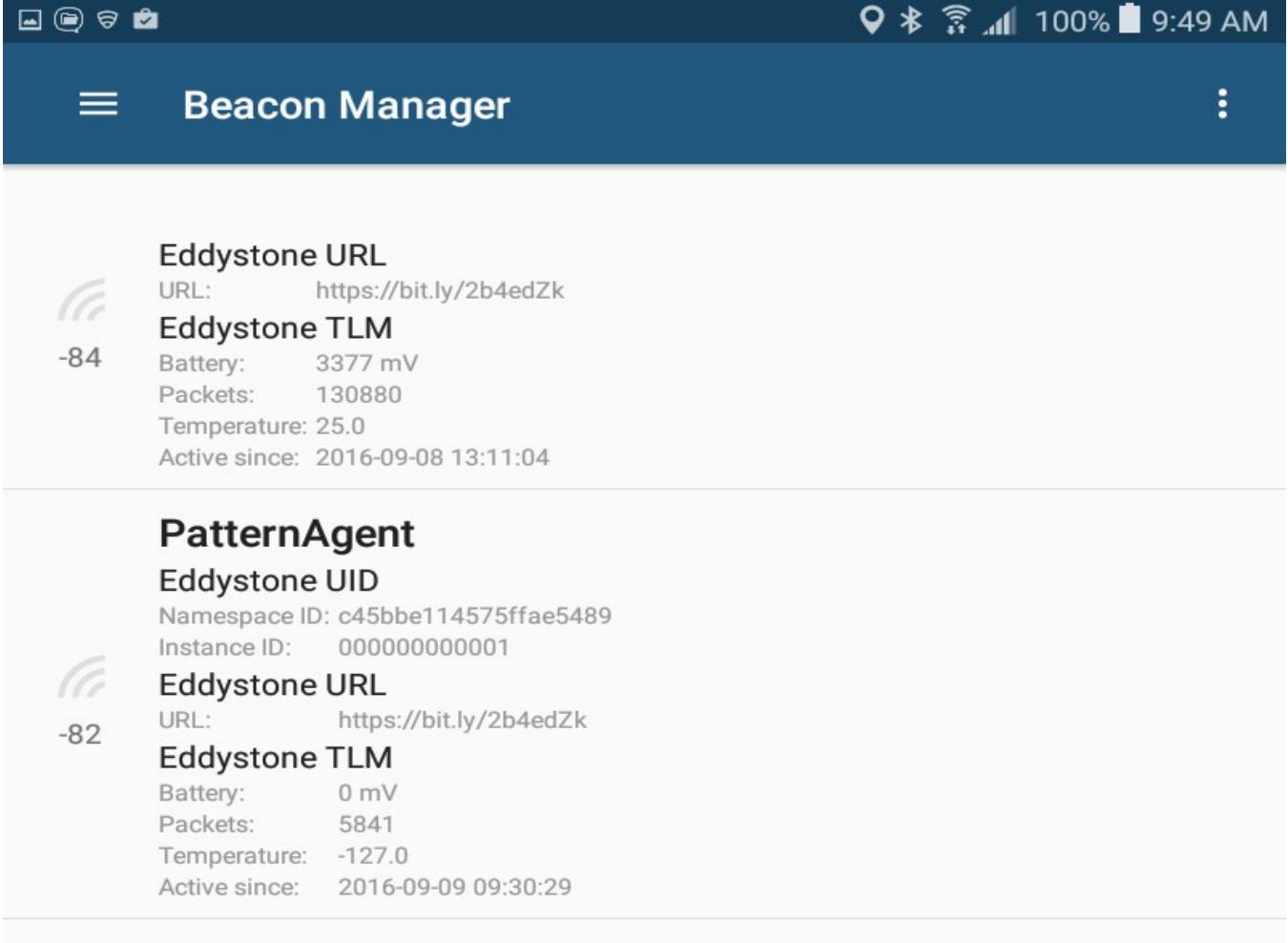
Source: Google

Registered



Source: Google

Beacon Manager



The screenshot shows the Beacon Manager app running on an Android device. The top status bar displays various icons and the time 9:49 AM. The battery level is at 100%. The main screen has a dark blue header with the title "Beacon Manager". Below the header, there are two entries for beacons:

Eddystone URL
URL: <https://bit.ly/2b4edZk>

Eddystone TLM
Battery: 3377 mV
Packets: 130880
Temperature: 25.0
Active since: 2016-09-08 13:11:04

PatternAgent

Eddystone UID
Namespace ID: c45bbe114575ffae5489
Instance ID: 000000000001

Eddystone URL
URL: <https://bit.ly/2b4edZk>

Eddystone TLM
Battery: 0 mV
Packets: 5841
Temperature: -127.0
Active since: 2016-09-09 09:30:29

Source: BEACONinside GmbH

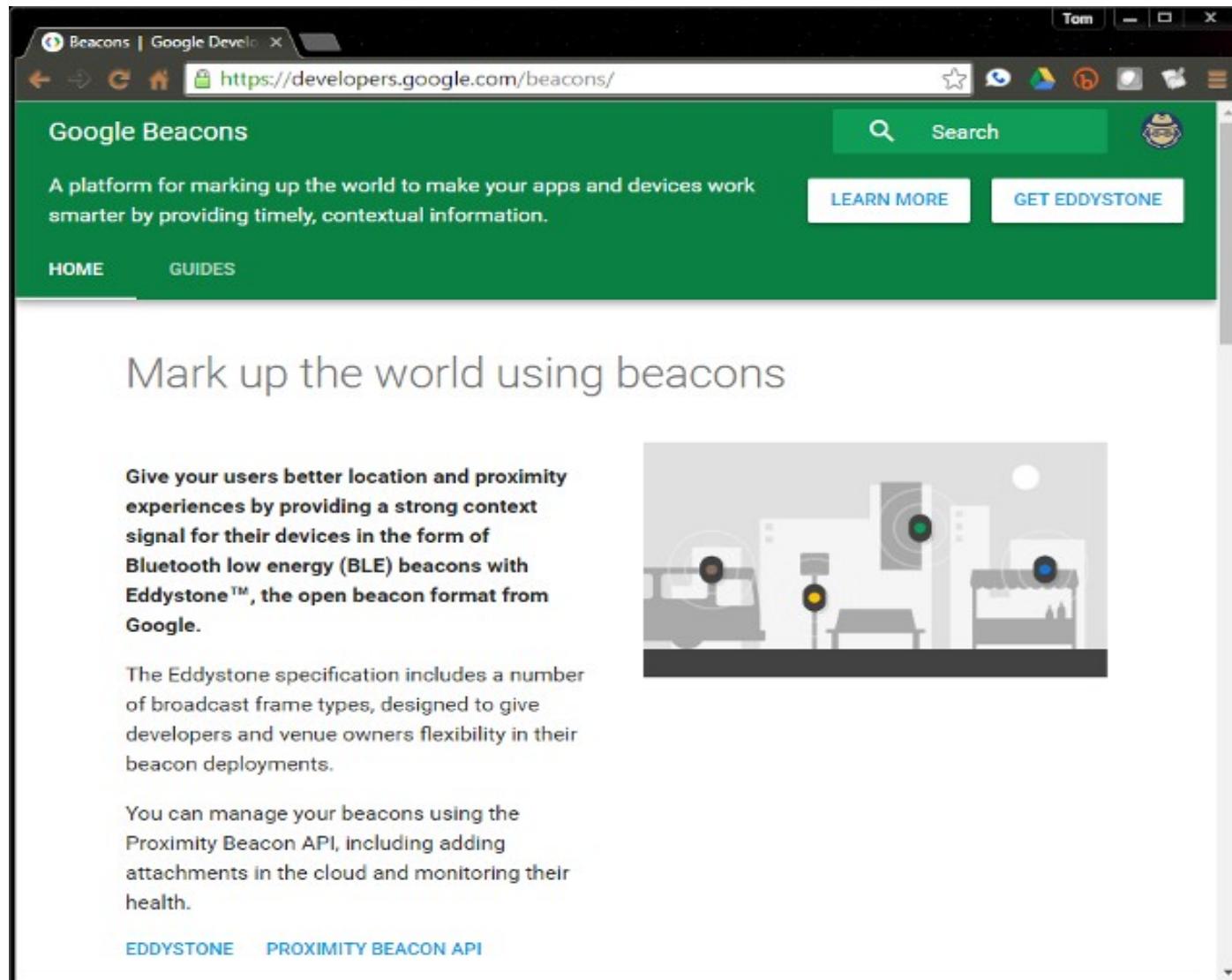
Beacon Toy



- * Turn your phone into a Beacon!
- * Great for impromptu testing, etc.
- * Free app

Source: Adrian Cretu

Google Developer - Beacons



The screenshot shows a web browser window displaying the Google Beacons developer page at <https://developers.google.com/beacons/>. The page has a green header with the title "Google Beacons". Below the header, there is a sub-header: "A platform for marking up the world to make your apps and devices work smarter by providing timely, contextual information." To the right of this text are two buttons: "LEARN MORE" and "GET EDDYSTONE". Below the header, there are two navigation links: "HOME" and "GUIDES". The main content area features a large heading "Mark up the world using beacons". To the left of this heading is a block of text explaining the purpose of beacons: "Give your users better location and proximity experiences by providing a strong context signal for their devices in the form of Bluetooth low energy (BLE) beacons with Eddystone™, the open beacon format from Google." Below this text is another block of text: "The Eddystone specification includes a number of broadcast frame types, designed to give developers and venue owners flexibility in their beacon deployments." At the bottom of the main content area, there are two links: "EDDYSTONE" and "PROXIMITY BEACON API". To the right of the main content area is a small image showing a cityscape with several beacons represented by circles with dots, illustrating how they can be deployed in the real world.

Source: Google

Google Beacon Dashboard

The screenshot shows a web browser window titled "Beacon Dashboard | Google". The address bar displays the URL <https://developers.google.com/beacons/dashboard/#/beacons?pn=10>. The main content area is titled "Google Beacons > Dashboard". On the left, there is a section labeled "Beacons" with a search bar containing "Add search term" and a help icon. Below this is a list item for "patternagents". The list item includes a green circular icon with a white "B" symbol, the text "1 beacon.", and a download arrow icon. To the right of the list item are two columns: "Description" and "Level Status". The "Description" column contains the text "Patternagents: Open Source Design Patterns". The "Level Status" column contains the text "1 Active".

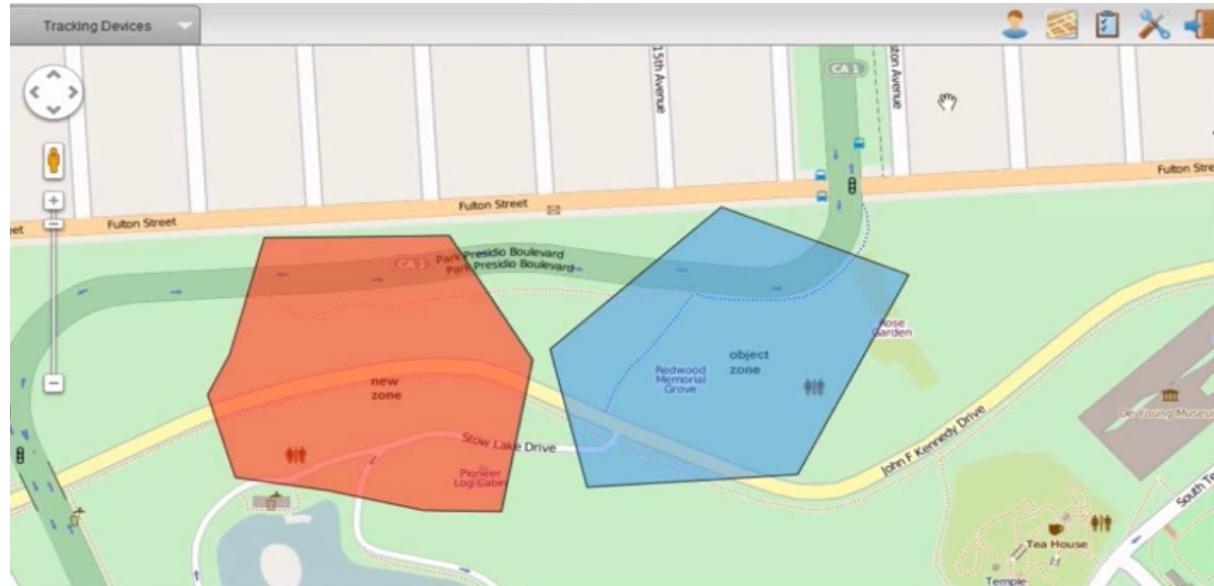
Description	Level	Status
Patternagents: Open Source Design Patterns	1	Active

Source: Google

Proximity Beacon API

- * Eddystone is much more than just a Beacon Format
- * Registered Beacons can have additional Properties and Attachments that get delivered to the app by the Proximity Beacon API
- * Also supports iBeacon, AltBeacon
- * Attachment can indicate functional location (i.e. Store X Front Door, Store Y Aisle 19, etc.)
Makes it easier to replace or retask a beacon
- * Integrates with other Google APIs, Nearby, etc.

What is a Geo-Fence?



- * A Geo-Fence is a virtual perimeter for a real-world geographic area.
- * Geo-Fences can be implemented using Cell Tower Positioning, GPS location, Wi-Fi Signal Strength measurement or any combination of those.

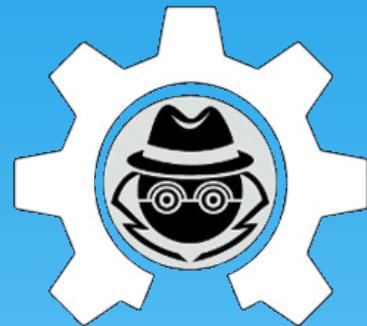
Source : Wikipedia

eBeacons vs. Geo-Fences

- * Chrome is the App
- * Self-Contained
- * User enables Phy. Web
- * Typical 70m area max
- * Best for small areas
- * Needs dedicated App
- * Needs GPS, Cell, Wi-Fi
- * User must Opt-In
- * Typical 100m area min
- * Best for large areas

Source: @PatternAgents

Hardware for Beacons



*A quick How-To Overview...

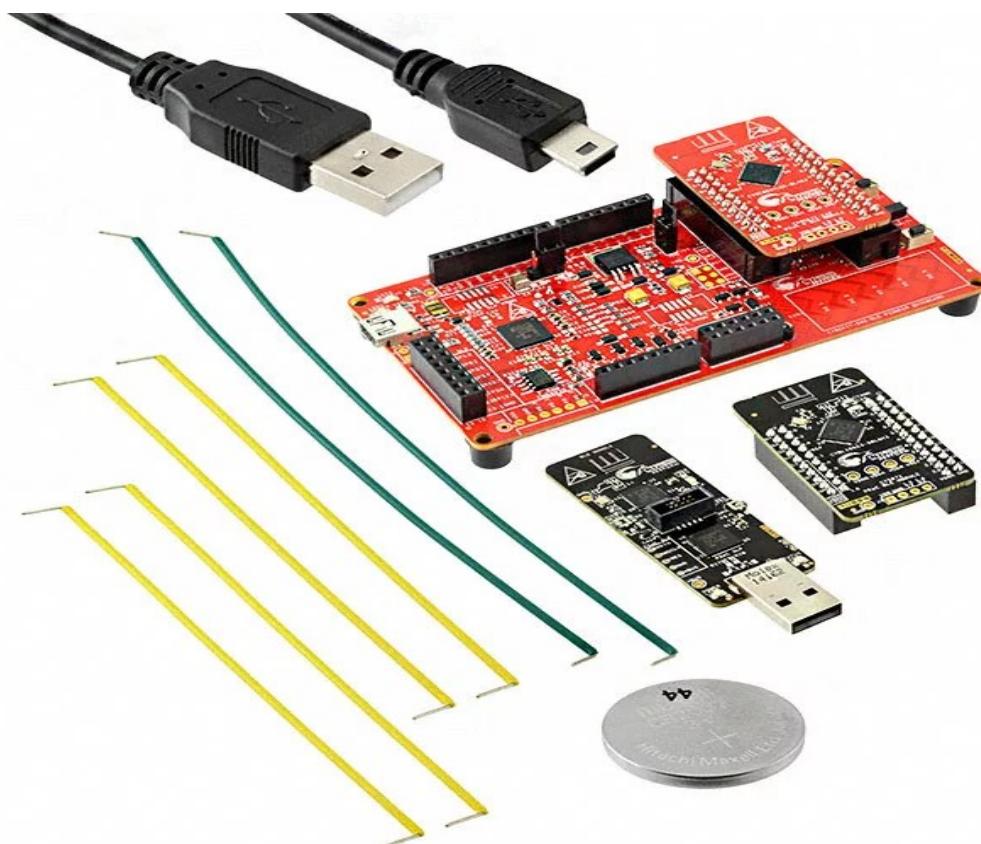
Bluetooth Hardware



- * Several Solutions available, including your phone!
- * Cypress Semiconductor, Nordic Semiconductor, TI, Silicon Labs, etc. all have Bluetooth Radios
- * Almost all of them have Eddystone Reference Designs that are freely available

Cypress CY8CKIT-042-BLE

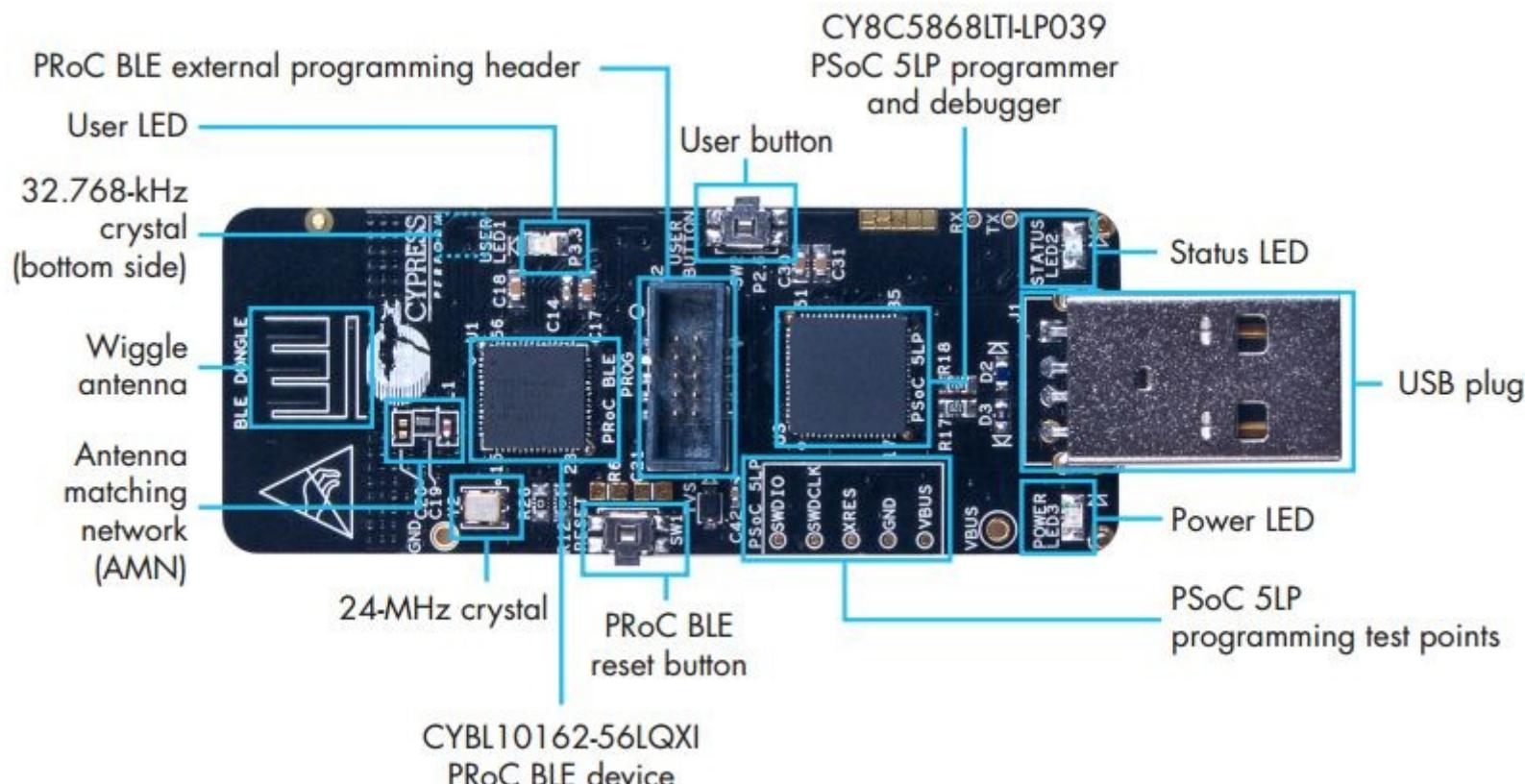
- * Cypress Programmable System-on-Chip
- * Malleable hardware that you can program!



Source: Cypress

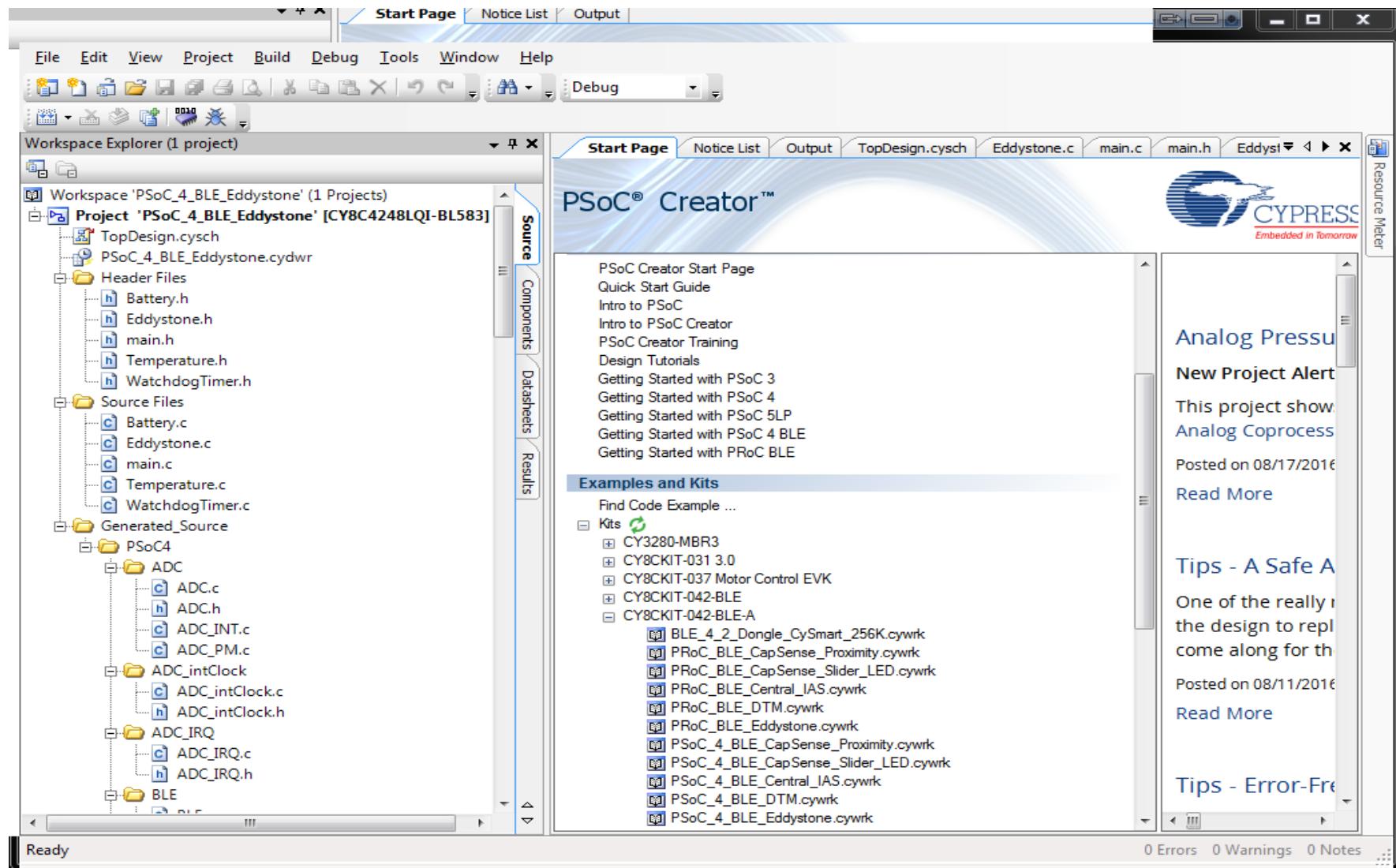
Cypress Smart Dongle

- * Cypress Programmable System-on-Chip
- * Built-in JTAG Programmer



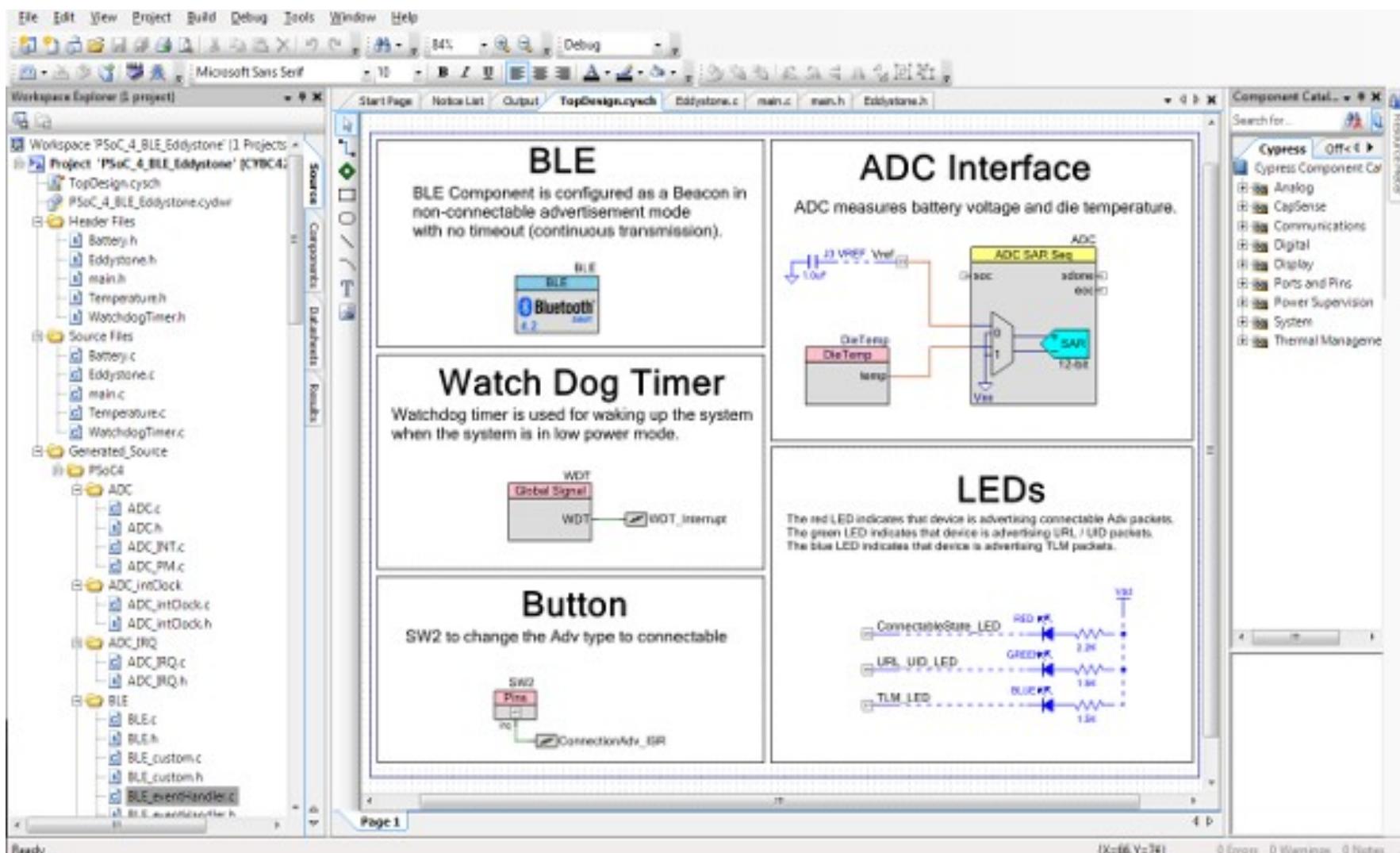
Source: Cypress

PSoC Creator



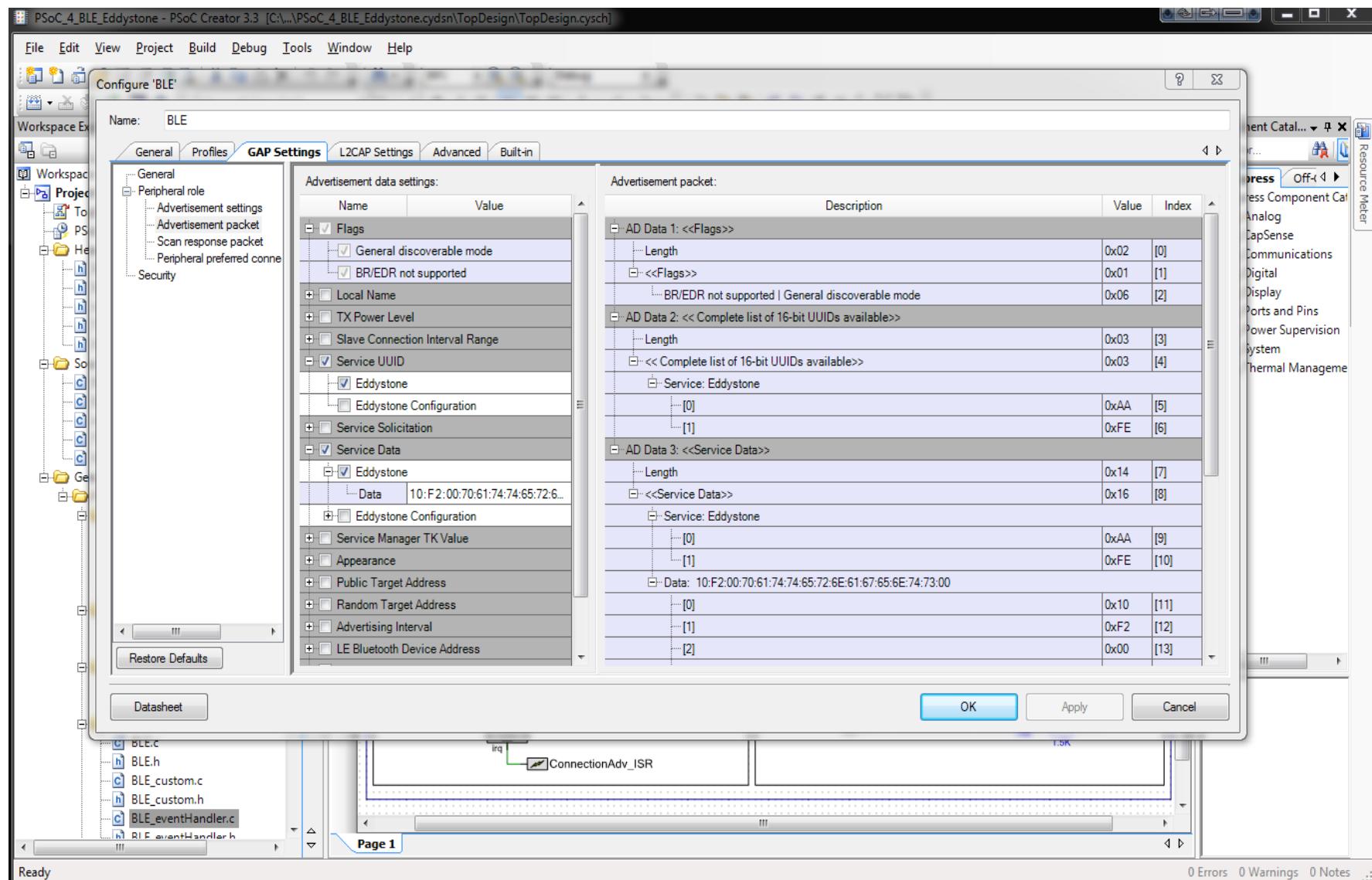
Source: Cypress

Creator Schematic



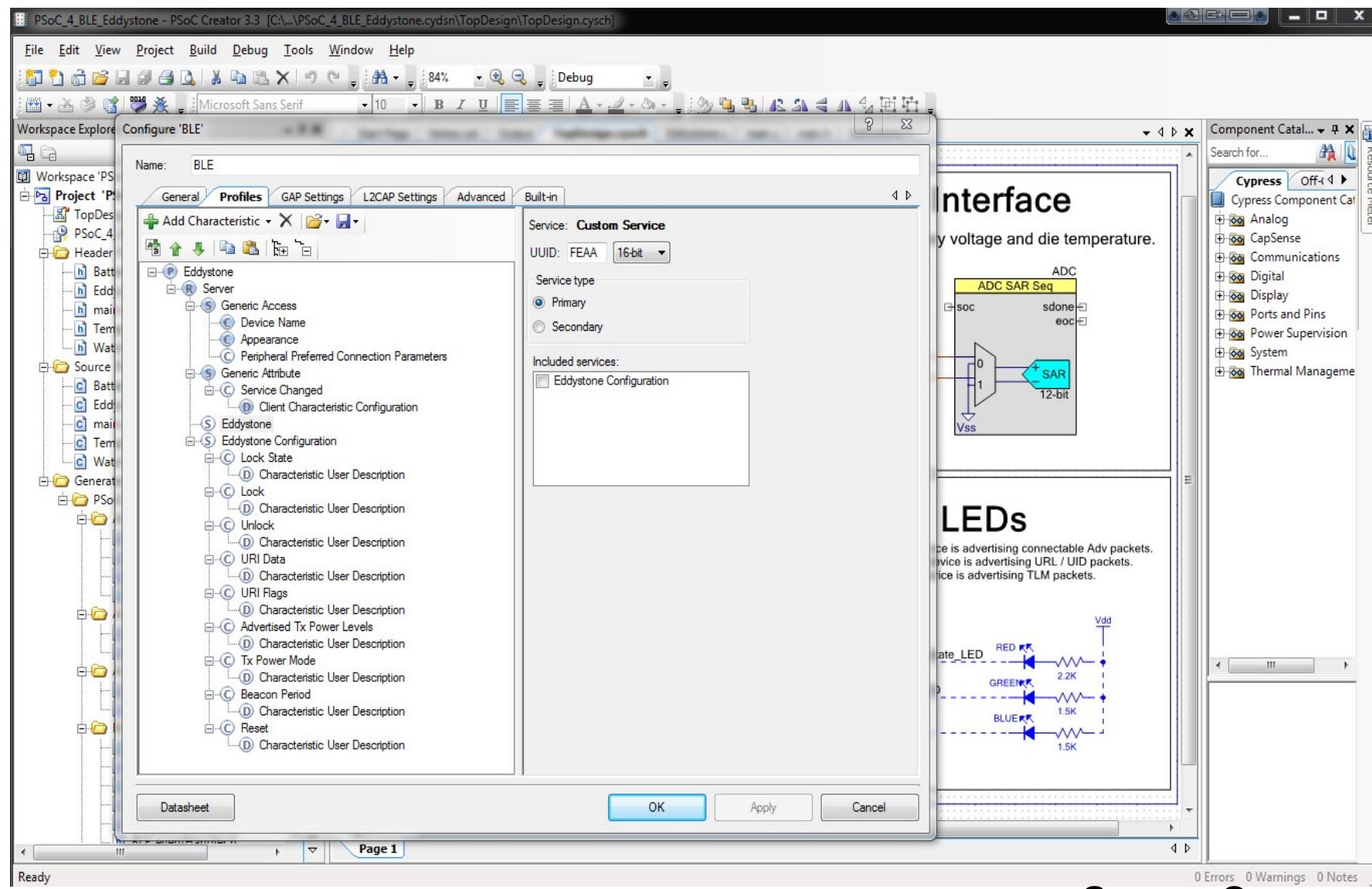
Source: Cypress

BLE Component Design



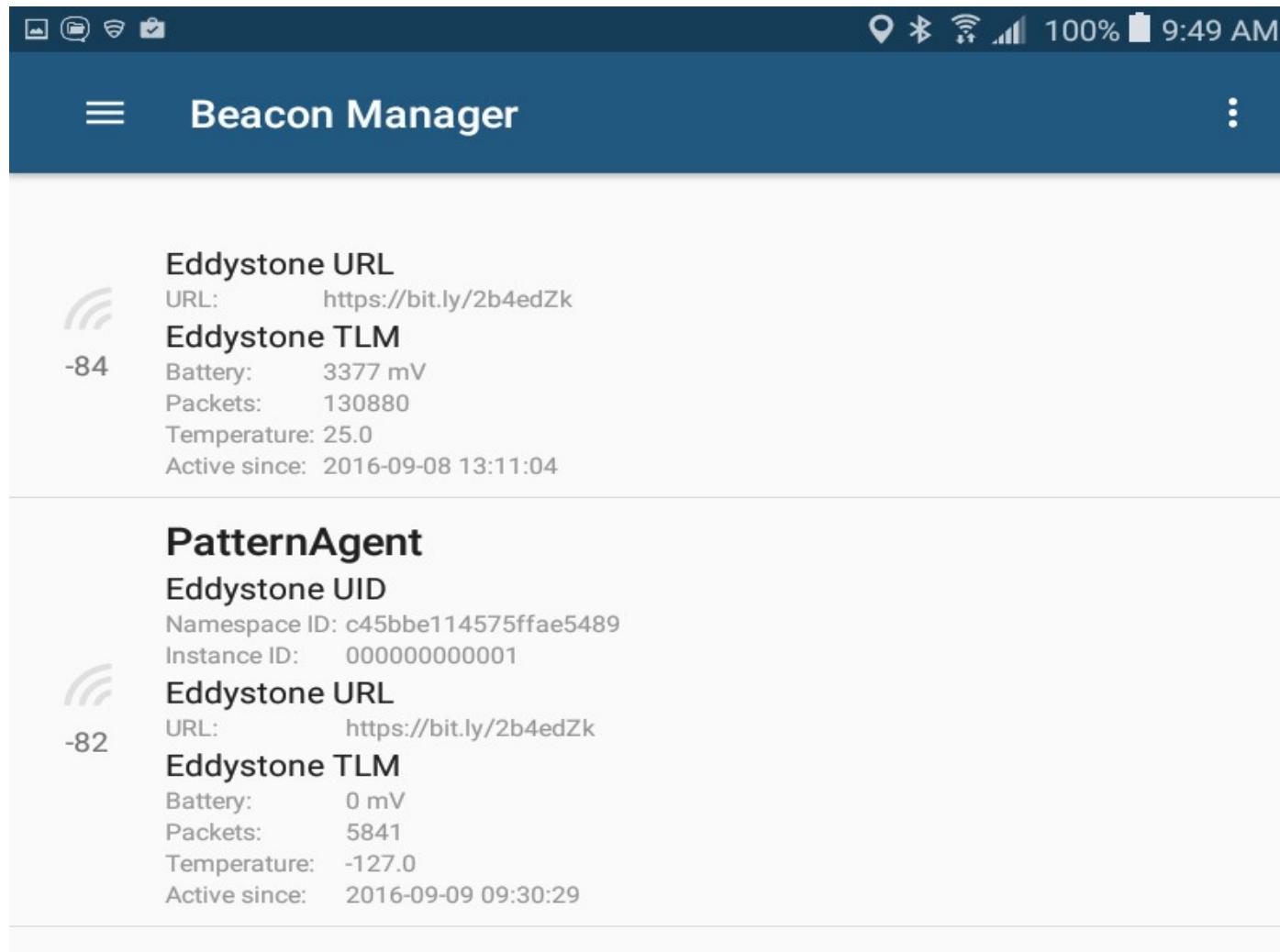
Source: Cypress

GAP/GATT Configuration



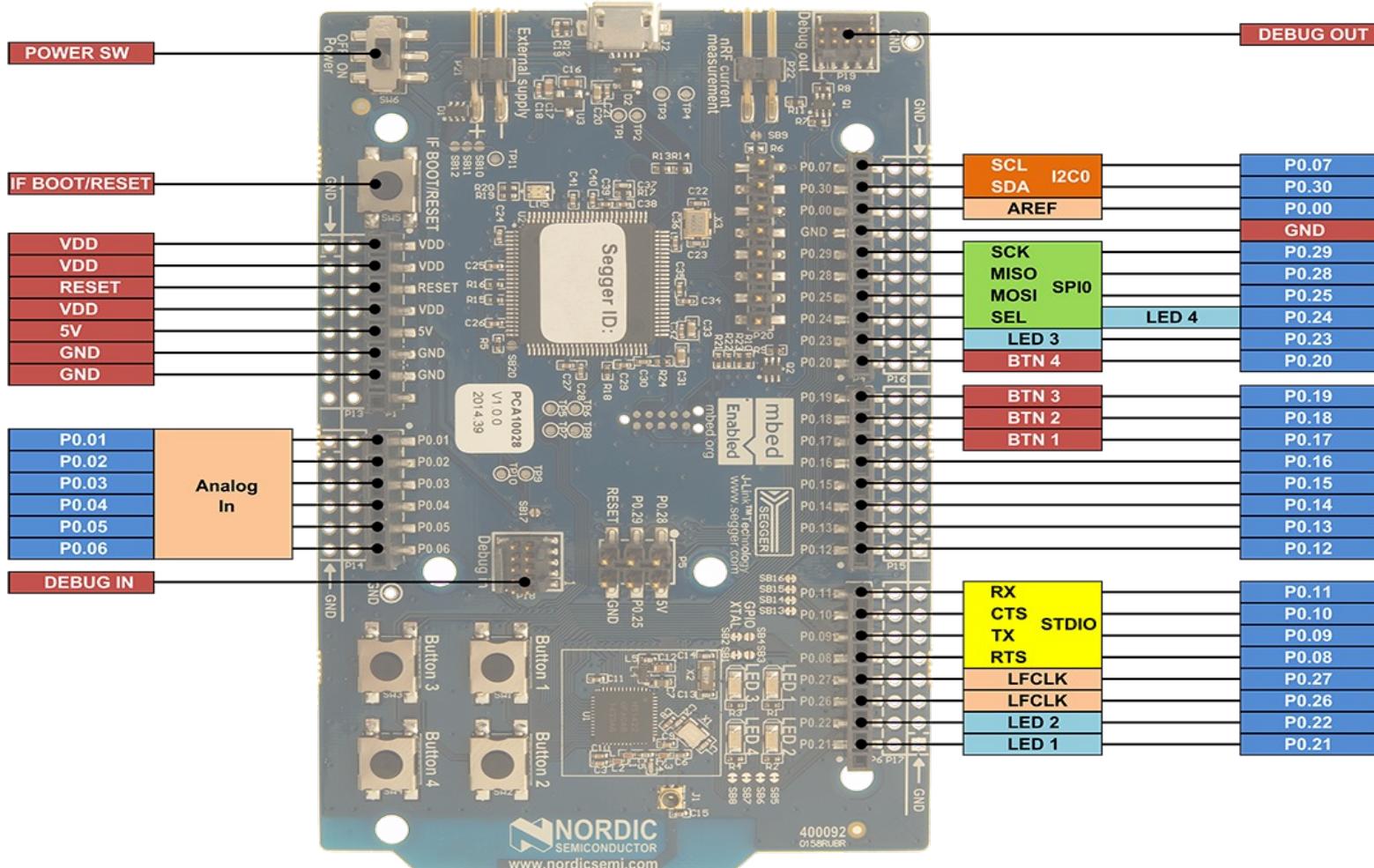
Source: Cypress

Test the Beacon Firmware



Source: @PatternAgents

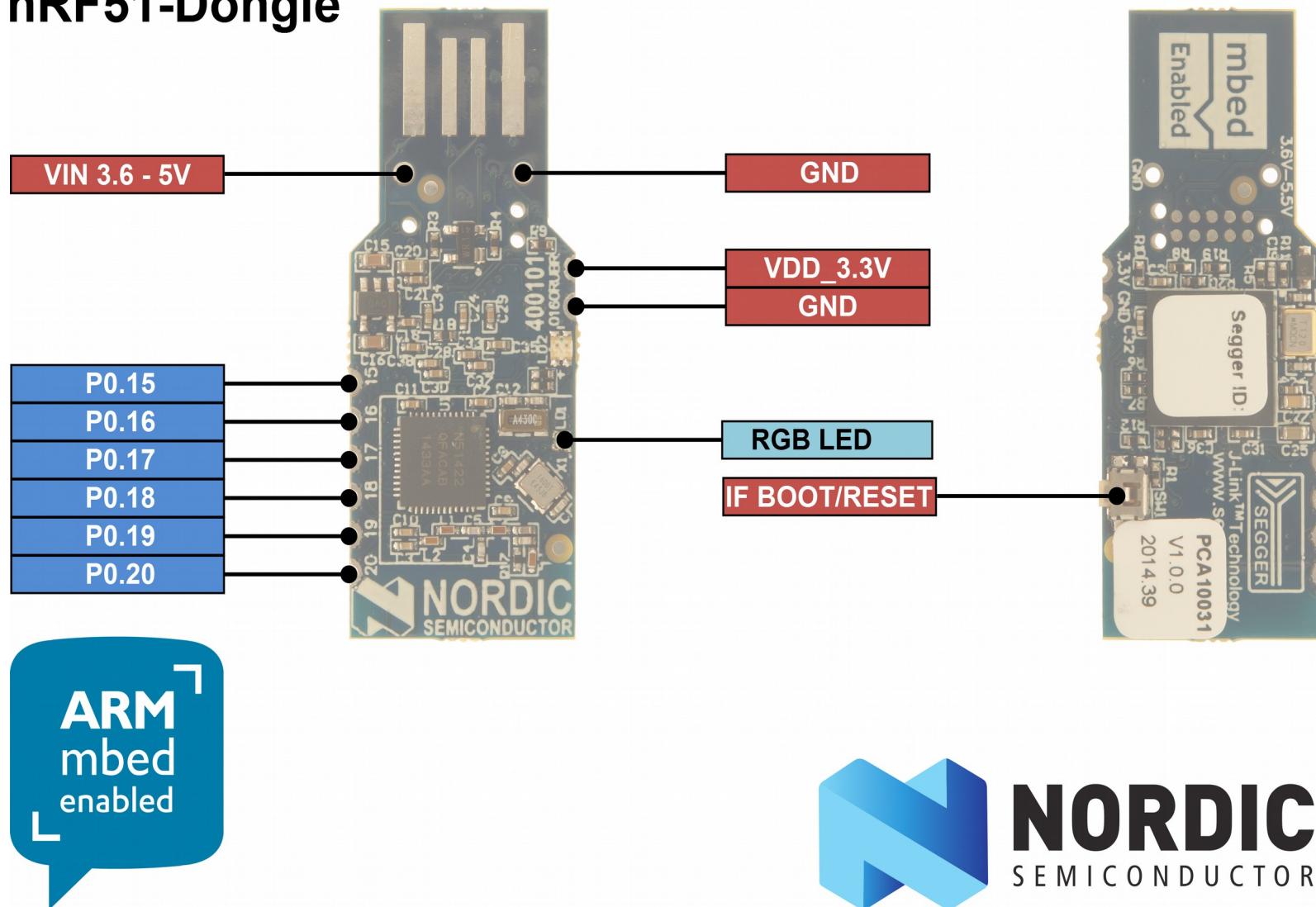
Nordic nRF51 Series



Source: NordicSemi

Nordic nRF51 Series

nRF51-Dongle



ARM mBed Online

- * We'll use the ARM mbed on-line tools
- * Nothing to install, just create account
- * We'll import the Google Reference code
- * The boards enumerate via USB as Mass Storage device class
- * Just download the compiled hex file to the Mass Storage class device (i.e. E:) and it programs the firmware for you, Easy!

ARM mbed Example

The screenshot shows a web browser displaying the ARM mbed developer website. The URL in the address bar is <https://developer.mbed.org/teams/mbed-os-examples/code/mbed-os-example-ble-EddystoneService/>. The page title is "mbed-os-examples / mbed-os-example-ble-EddystoneService". The main content area describes the Eddystone Beacon example, mentioning it broadcasts URLs to nearby BLE devices. It details the two-stage process: configuration service modification and beacon advertisement. A sidebar on the right contains a "Repository toolbox" with options like "Import into Compiler", "Build repository", and "Clone repository to desktop". Below it is a "Repository details" section with statistics: Type: Program, Created: 26 Jul 2016, Imports: 16, Forks: 0, Commits: 10, Dependents: 0, and Dependencies: 0.

mbed-os-examples / mbed-os-example-ble-EddystoneService

Eddystone beacons broadcast a small amount of information, like URLs, to nearby BLE devices. The canonical source for this example lives at https://github.com/ARMmbed/mbed-os-example-ble/tree/master/BLE_EddystoneService.

Home History Graph API Documentation Wiki Pull Requests

Eddystone beacons broadcast a small amount of information, like URLs, to nearby BLE devices.

The Eddystone Beacon sample application runs in two stages:

- On startup, the Configuration Service (which allows modification of the beacon) runs for a user defined period (default - 30 seconds).
- When the Configuration Service period ends, the Eddystone Service broadcasts advertisement packets.

Running the application

Requirements

You should install the "Physical Web" application on your phone:

- Android version
- iOS version

Note: It is also possible to use a regular scanner to interact with your Eddystone beacon but it requires knowledge about BLE and Eddystone beacon specification out of the scope of this document.

Hardware requirements are in the main readme.

Building instructions

Source: ARM

Select your Board

The screenshot shows the mbed Compiler interface with the following details:

- Program Workspace:** Shows a tree view of files and projects, including "My Programs" and "mbed-os-example-ble-EddystoneService".
- Current Project:** Program: /mbed-os-example-ble-EddystoneService
- Platform Selection Dialog:** A modal window titled "Nordic nRF51-DK" is open, displaying information about the board and its memory usage.
- Description:** The dialog includes a description of the nRF51 Development Kit, stating it's a single-board development kit for Bluetooth Smart, ANT and 2.4GHz proprietary applications using the nRF51 Series SoC. It supports both development for nRF51822 and nRF51422 SoCs. The kit is compatible with the Arduino Uno Rev3 standard and supports 3rd-party shields.
- Memory Usage:** A table shows memory usage for the nRF51-DK:

Type	Size	Max
Flash	61.1 kB	270.0 kB
SRAM	5.9 kB	n/a
RAM	3.1 kB	256.0 kB
FLASH	0.9 kB	24.0 kB
SRAM	7.3 kB	24.0 kB
RAM	1.8 kB	n/a
FLASH	65.1 kB	n/a
SRAM	64.9 kB	256.0 kB
RAM	8.8 kB	24.0 kB
- Your registered platforms:** A grid of icons for various boards, including u-Blox C027, NUCLEO-F103RB, NUCLEO-L152RE, NUCLEO-L053RBT, NUCLEO-F411RE, NUCLEO-F401RE, Nordic nRF51-DK, Nordic nRF52-DK, and an "Add Platform" button.
- Bottom Navigation:** Buttons for "Compile Output", "Find Results", and "Notifications".

Change UUID

ARM mbed Compiler /mbed-os X https://developer.mbed.org/compiler/#nav:/mbed-os-example-ble-EddystoneService/source/main.cpp; mbed /mbed-os-example-ble-EddystoneService/source/main.cpp New Import Save Save All Compile Commit Revision Find Notifications Help Nordic nRF51-DK Default Program Workspace

My Programs
F103RB_rsvp_dev
F103RB_rtos_test
L152RE_rtos_test
mbed-os-example-ble-EddystoneService
Classes
Structs
img
mbed-events
shields
source
PersistentStorageHelper
EddystoneService.cpp
EddystoneService.h
EddystoneTypes.h
main.cpp
TLMFrame.cpp
TLMFrame.h
UIDFrame.cpp
UIDFrame.h
URLFrame.cpp
URLFrame.h
config.json
mbed_app.json
module.json
readme.md
mbed-os
mbed_EddystoneURL_Beacon
mbed_rtos_test
Nucleo_blink_led
Nucleo_display_time
Nucleo_firstRun_test
Nucleo_i2c_master
Nucleo_printf
Nucleo_read_analog_value
Nucleo_read_button_interrupt
Nucleo_rtos_demo_thread

main.cpp EddystoneService.h EddystoneTypes.h config.json

```
20 #include "ble/BLE.h"
21 #include "EddystoneService.h"
22
23 #include "PersistentStorageHelper/ConfigParamsPersistence.h"
24
25 EddystoneService *eddyServicePtr;
26
27 /* Duration after power-on that config service is available. */
28 static const int CONFIG_ADVERTISEMENT_TIMEOUT_SECONDS = 30;
29
30 /* Default UID frame data */
31 /* SHA1 of http://patternagents.com */
32 static const UIDNamespaceID_t uidNamespaceID = {0xC4, 0x5B, 0xBE, 0x11, 0x45, 0x75, 0xFF, 0xAE, 0x54, 0x89};
33 static const UIDInstanceID_t uidInstanceId = {0x00, 0x00, 0x00, 0x00, 0x00, 0x01};
34
35 /* Default version in TLM frame */
36 static const uint8_t tlmVersion = 0x00;
37
38 /* Values for ADV packets related to firmware levels, calibrated based on measured values at 1m */
39 static const PowerLevels_t defaultAdvPowerLevels = {-47, -33, -21, -13};
40 /* Values for radio power levels, provided by manufacturer. */
41 static const PowerLevels_t radioPowerLevels = {-30, -16, -4, 4};
42
43 static EventQueue eventQueue(
44     /* event count */ 16 /* event size */ 32
```

Find results for "UID" in "/mbed-os-example-ble-EddystoneService" Occurrences: 500 Files: 16

Resource	Resource	In Folder	Location
/* Default UID frame data */	main.cpp	/mbed-os-example-ble-EddystoneSe	Line: 30, Col: 12
static const UIDNamespaceID_t uidNamespaceID = {0xC4, 0x5B, 0xBE, 0x11, 0x45	main.cpp	/mbed-os-example-ble-EddystoneSe	Line: 32, Col: 14
static const UIDNamespaceID_t uidNamespaceID = {0xC4, 0x5B, 0xBE, 0x11, 0x45, 0:	main.cpp	/mbed-os-example-ble-EddystoneSe	Line: 32, Col: 31
static const UIDInstanceID_t uidInstanceId = {0x00, 0x00, 0x00, 0x00, 0x00	main.cpp	/mbed-os-example-ble-EddystoneSe	Line: 33, Col: 14
static const UIDInstanceID_t uidInstanceId = {0x00, 0x00, 0x00, 0x00, 0x01};	main.cpp	/mbed-os-example-ble-EddystoneSe	Line: 33, Col: 31
/* Set default URL, UID and TLM frame data if not initialized through the config se	main.cpp	/mbed-os-example-ble-EddystoneSe	Line: 91, Col: 25

Compile Output Find Results Notifications 1

Change URL

The screenshot shows the mbed Compiler IDE interface. The title bar reads "mbed Compiler /mbed-os X". The browser tab shows the URL <https://developer.mbed.org/compiler/#nav:/mbed-os-example-ble-EddystoneService/source/EddystoneTypes.h>. The main window has tabs for "main.cpp", "EddystoneService.h", "EddystoneTypes.h" (which is the active tab), and "config.json".

The "Program Workspace" sidebar lists several projects and their files. The "mbed-os-example-ble-EddystoneService" project is expanded, showing files like "PersistentStorageHelper", "EddystoneService.cpp", "EddystoneService.h", "EddystoneTypes.h", "main.cpp", "TLMFrame.cpp", "TLMFrame.h", "UIDFrame.cpp", "UIDFrame.h", "URLFrame.cpp", "URLFrame.h", "config.json", "mbed_app.json", "module.json", "readme.md", and "mbed-os".

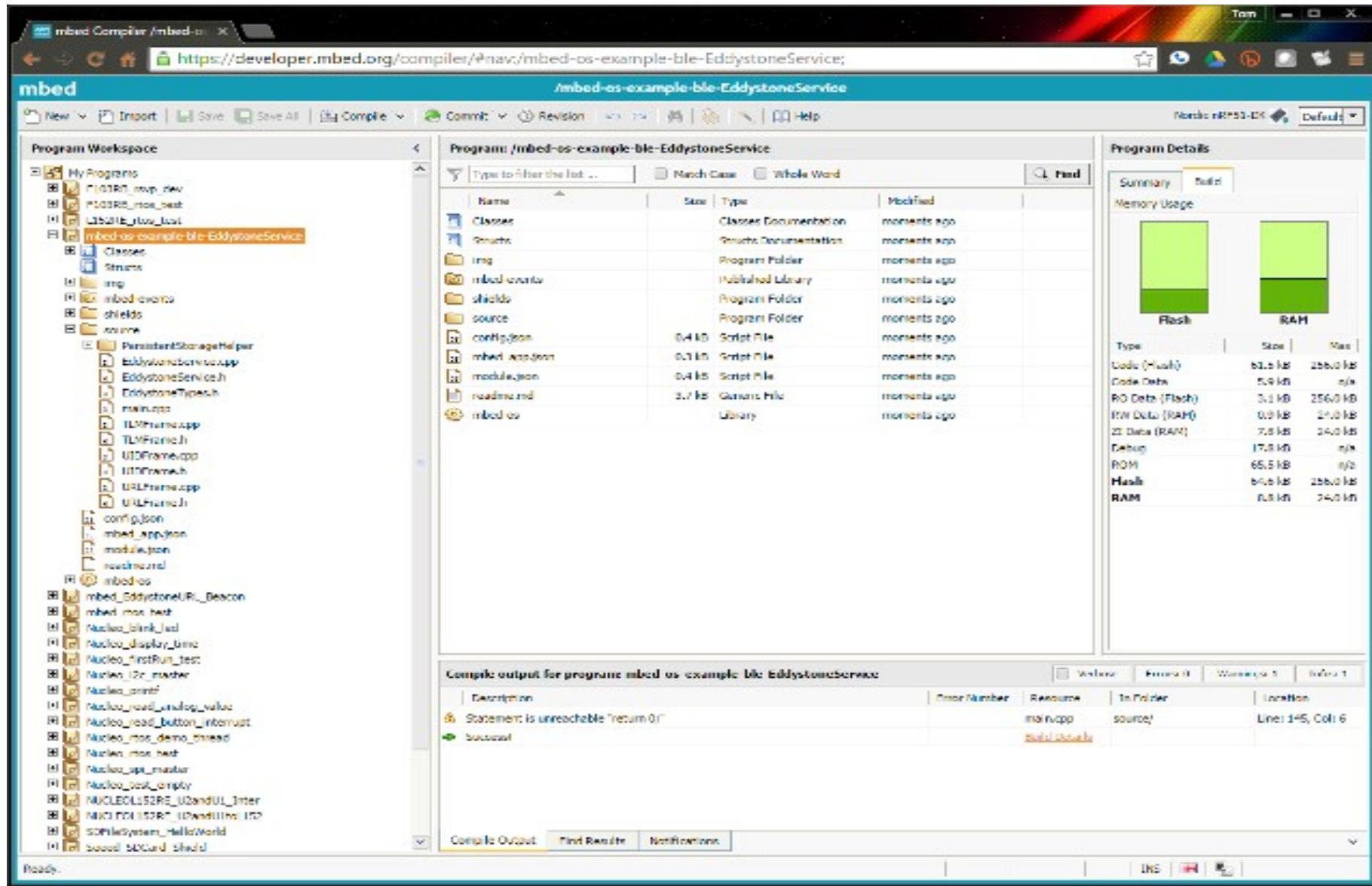
The "EddystoneTypes.h" code editor contains the following code:

```
16 #ifndef __EDDYSTONETYPES_H__
17 #define __EDDYSTONETYPES_H__
18
19
20 #include <stdint.h>
21 #include <stddef.h>
22
23 #ifndef YOTTA_CFG_EDDYSTONE_DEFAULT_DEVICE_NAME
24     #define YOTTA_CFG_EDDYSTONE_DEFAULT_DEVICE_NAME "PatternAgent"
25 #endif
26
27 #ifndef YOTTA_CFG_EDDYSTONE_DEFAULT_URL
28     #define YOTTA_CFG_EDDYSTONE_DEFAULT_URL "https://bit.ly/2b4ed2k"
29 #endif
30 /**
31 * Macro to expand a 16-bit Eddystone UUID to 128-bit UUID.
32 */
33
34 #define UUID_URL_BEACON(FIRST, SECOND) {
35     0xee, 0x0c, FIRST, SECOND, 0x87, 0x86, 0x40, 0xba,
36     0xab, 0x96, 0x99, 0xb9, 0x1a, 0xc9, 0x81, 0xd8,
37 }
38
39 /**
40 * Eddystone 16-bit UUID.
```

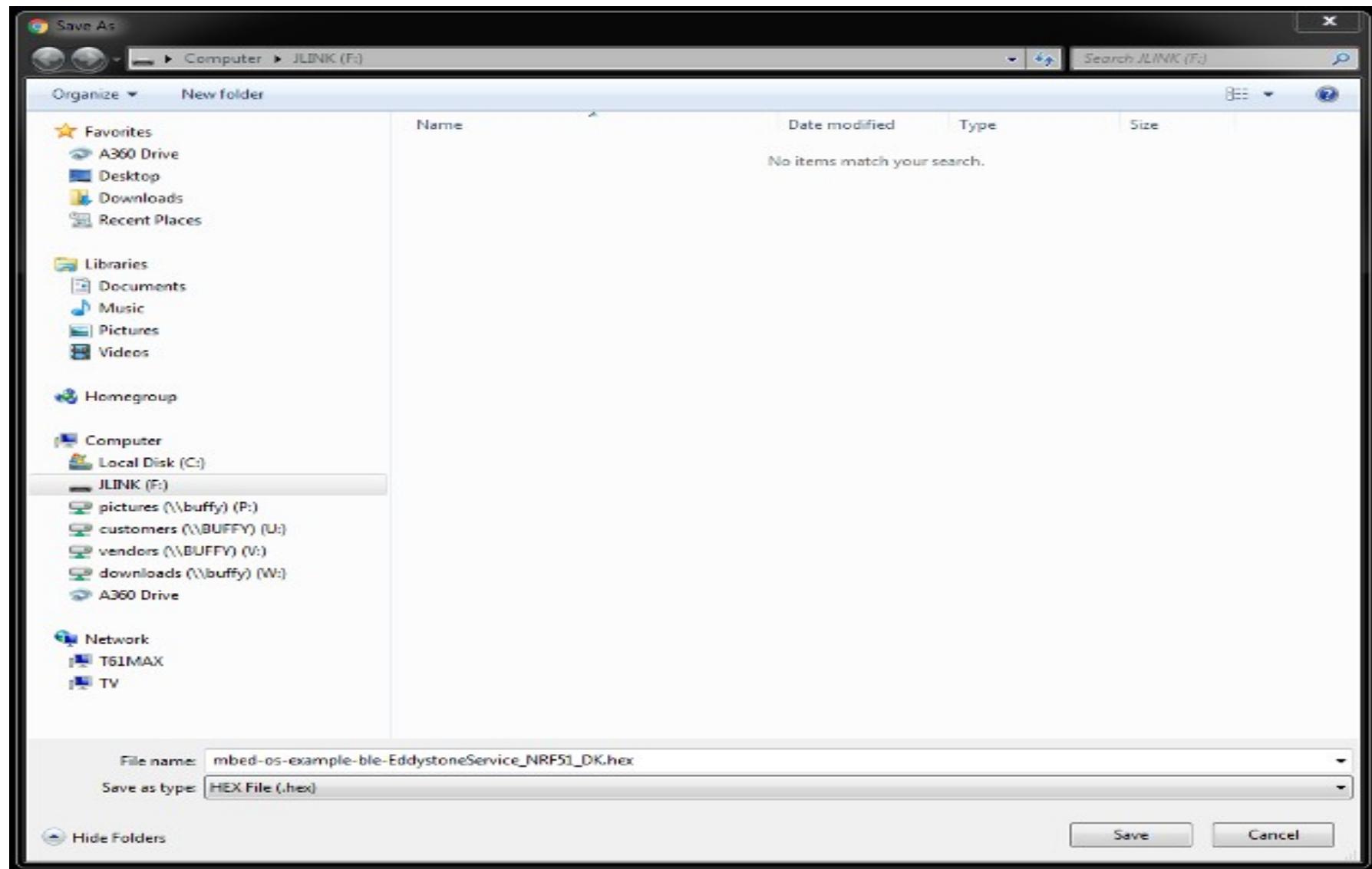
The "Find Results" panel at the bottom shows occurrences of "EDDYSTONE_DEFAULT_URL" across three files: main.cpp, EddystoneTypes.h, and EddystoneService.h. The results are:

Resource	Resource	In Folder	Location
const char* url = YOTTA_CFG_EDDYSTONE_DEFAULT_URL;	main.cpp	/mbed-os-example-ble-EddystoneSe	Line: 92, Col: 33
#ifndef YOTTA_CFG_EDDYSTONE_DEFAULT_URL	EddystoneTypes.h	/mbed-os-example-ble-EddystoneSe	Line: 27, Col: 19
#define YOTTA_CFG_EDDYSTONE_DEFAULT_URL "https://bit.ly/2b4ed2k"	EddystoneTypes.h	/mbed-os-example-ble-EddystoneSe	Line: 28, Col: 23
const char DEFAULT_URL[] = YOTTA_CFG_EDDYSTONE_DEFAULT_URL;	EddystoneTypes.h	/mbed-os-example-ble-EddystoneSe	Line: 102, Col: 38
#ifndef YOTTA_CFG_EDDYSTONE_DEFAULT_URL_FRAME_INTERVAL	EddystoneService.h	/mbed-os-example-ble-EddystoneSe	Line: 35, Col: 19
#define YOTTA_CFG_EDDYSTONE_DEFAULT_URL_FRAME_INTERVAL 700	EddystoneService.h	/mbed-os-example-ble-EddystoneSe	Line: 36, Col: 23

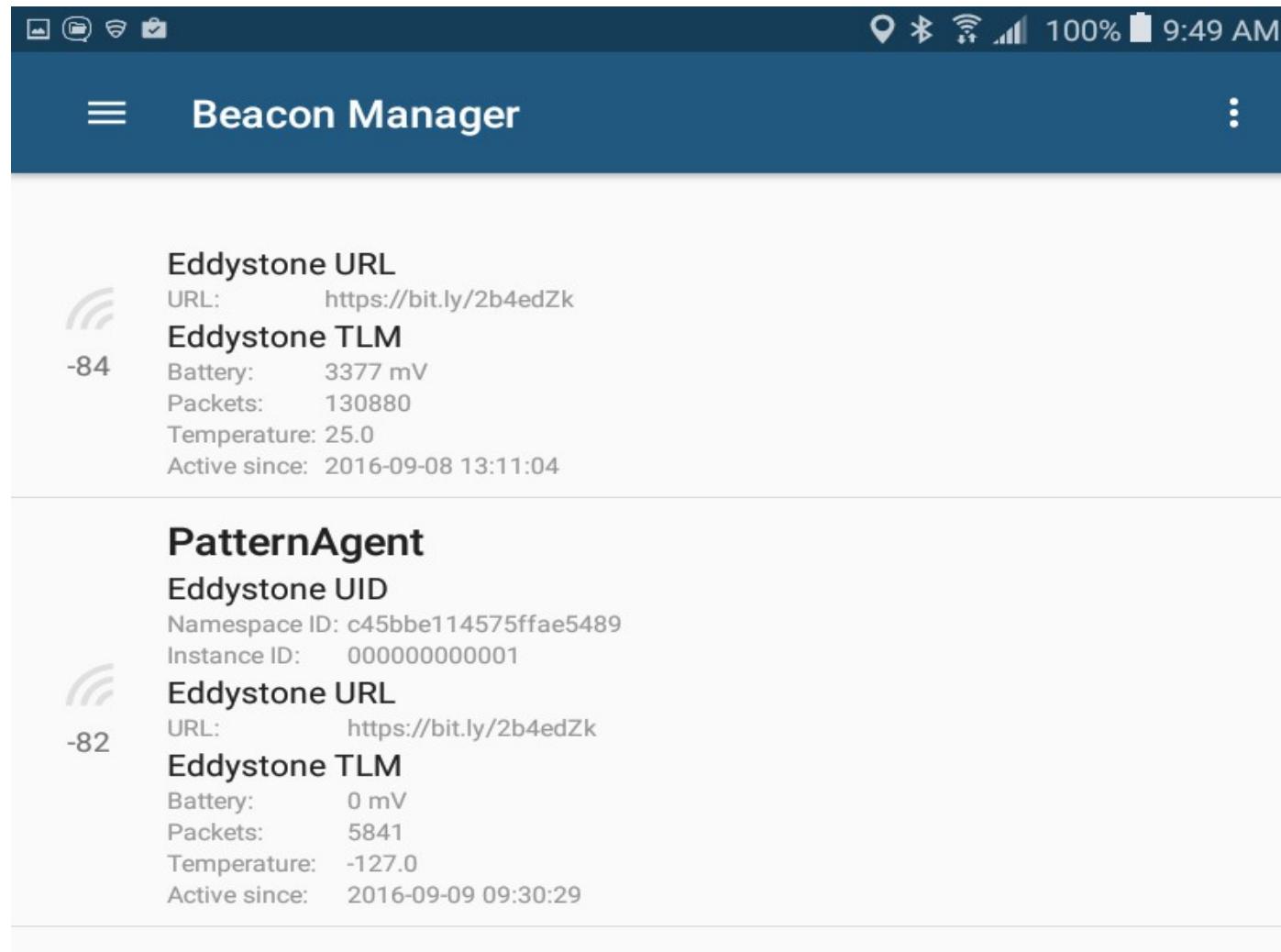
Compile Eddystone Project



Download and Run



Test the Beacon Firmware



Source: @PatternAgents

Creekside Five Beehive

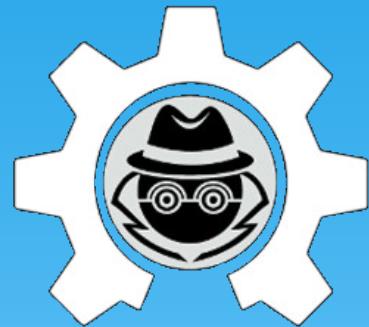


- * The Beacon is set to describe the Beehive just outside
- * It's a simple page, just a demo
- * We'll be adding metrics to it as time allows...

Summary

- * Beacons are just starting to take off
- * IMHO, Eddystone will win in 2-3 years time
- * Decentralized, open standard
- * Low barriers to entry
- * Proximity Beacon API is awesome, well designed
- * Attachments provide more data to apps than the beacon is capable of in hardware, a nice idea!
- * Attachments can be JSON packets or whatever...
- * “One beacon, Many apps!”
- * It's easy, go ahead and deploy your own beacons!

Questions?



* Thank You!

Questions

- * Are World Bee Populations actually increasing?

Personally, I don't think there is enough data on wild hives to make that assertion; however, (perhaps in response to increased CCD awareness) the number of new, "managed" hives introduced in the US is currently exceeding the CCD die-off rate of older and existing "managed" hives.

- * Many more people are introducing backyard and urban hives in the U.S. and elsewhere; and are volunteering to share data on their hives, so – more hives...

Questions ?

- * Are RF emissions safe for Bees and/or People?

Clearly, exposure to any *high power* RF source can be a problem for living things, but are the 2.4Ghz signals of Bluetooth and Wi-Fi radios high enough power to be a problem for Bees? In my experience, there has been no discernible difference between adjacent hives, one in close proximity to the transmitter, and the other 6-8 feet away.
(Hardly a study... just anecdotal...)

However, the transmitter works best “outside” the hive, so an external antennae placement is preferred.

Beacons for the Bees

For PDXMakerWeek, we will be talking about working with the new Google EddyStone Beacon technology and other physical web applications, like Beacons for Beehives. Google Beacons create a simple way to "tag" the physical location of objects, such as wild and domestic Beehives. We'll demonstrate ways to monitor and Geo-tag Beehives.

To participate in the presentation, you can:

- bring a Bluetooth Enabled Smart Phone
- install the latest version of Google Chrome
- enable Physical Web Notifications in Google Chrome
(see: support.google.com/chrome/answer/6239299)

RSVP REQUIRED:

Wednesday, September 14, 2016
6:00 PM to 8:00 PM



@PatternAgents
Open Source Design Patterns

