# DRAFT

# Blue2thprinting: {blue-[tooth}-printing]

## Answering the question "WTF am I even looking at?!"

Xeno Kovah
OpenSecurityTraining2 (ost2.fyi)
& Dark Mentor LLC (darkmentor.com)

# About Me

- 75% of my time is spent making free (as in beer), open access, and *open source* (CreativeCommons licensed) classes for a non-profit I started, **OpenSecurityTraining2 (ost2.fyi)**

# About Me

- <u>75% of my time</u> is spent making free (as in beer), open access, and *open source* (CreativeCommons licensed) classes for a non-profit I started, **OpenSecurityTraining2 (ost2.fyi)**

- <u>25% of my time</u> doing consulting and research for **Dark Mentor LLC**
  - The research is for fun, but is *also a trojan horse* to get me into conferences to tell you about OST2 ;)

# OST2 Crew

In order of appearance

Xeno Kovah

Arch1001 ~28 hours, Arch2001 ~27 hours
Arch4001 ~14 hours, HW1101 ~6 hours
Vulns1001 ~15 hours, Vulns1002 ~23 hours

Piotr Król

Arch4021 ~6 hours
Arch4031 ~6 hours

Michał Żygowski

Arch4021 ~6 hours

Thaís Moreira Hamasaki

RE3201 ~6 hours

Gal Zaban

RE3011 ~6 hours

Kc Udonsi

Vulns1001 ~15 hours

Cedric Halbronn

Dbg3011 ~6 hours
Arch2821 ~5 hours
Exp4011 ~33(!) hours

Sina Karvandi

Dbg3301 ~16 hours

# What I Want To Know:

# What Bluetooth Chip Is Inside Any Device

DARK MENTOR
DST2.FYI

nest

TEXAS INSTRUMENTS

BROADCOM®

SILICON LABS®

?

# Why I Want To Know It:

# So I Know if it's Vulnerable To a Firmware-Level Exploit

DARK MENTOR
DST2.FYI

TEXAS INSTRUMENTS

SILICON LABS

BROADCOM

DARK MENTOR
DST2.FYI

ARMIS®

TEXAS INSTRUMENTS

SILICON LABS®

BROADCOM®

ARMIS®

TEXAS INSTRUMENTS

SEMO
SECURE MOBILE NETWORKING

SILICON LABS®

BROADCOM®

ARMIS®

DARK MENTOR

TEXAS INSTRUMENTS

SEMO

SECURE MOBILE NETWORKING

SILICON LABS®

BROADCOM®

ARMIS®

DARK MENTOR

DARK MENTOR

TEXAS INSTRUMENTS

SEMO

SECURE MOBILE NETWORKING

SILICON LABS®

BROADCOM®

VersionPrint

ChipPrint

***3330*** *Product Makers registered* with Bluetooth SIG as of the time of writing!

***3330*** *Product Makers registered* with Bluetooth SIG as of the time of writing!

***3330*** *Product Makers registered* with Bluetooth SIG as of the time of writing!

***3330*** *Product Makers registered* with Bluetooth SIG as of the time of writing!

***3330*** *Product Makers registered* with Bluetooth SIG as of the time of writing!

***3330*** *Product Makers registered* with Bluetooth SIG as of the time of writing!

***3330*** *Product Makers registered* with Bluetooth SIG as of the time of writing!

There is almost always a 1:1 relationship from Products to Chips

**We need to discover it**

***3330*** *Product Makers registered* with Bluetooth SIG as of the time of writing!

There is almost always a 1:1 relationship from Products to Chips

**We need to discover it**

***3330*** *Product Makers registered* with Bluetooth SIG as of the time of writing!

There is almost always a 1:1 relationship from Products to Chips

**We need to discover it**

***3330*** *Product Makers registered* with Bluetooth SIG as of the time of writing!

***3330*** *Product Makers registered* with Bluetooth SIG as of the time of writing!

Module Makers only use certain chips
Mapping that, reduces the possible Chip space

***3330*** *Product Makers registered* with Bluetooth SIG as of the time of writing!

Module Makers only use certain chips
Mapping that, reduces the possible Chip space

***3330*** *Product Makers registered* with Bluetooth SIG as of the time of writing!

Module Makers only use certain chips
Mapping that, reduces the possible Chip space

***3330*** *Product Makers registered* with Bluetooth SIG as of the time of writing!

DARK MENTOR

VersionPrint
Ver. 1
Ver. 2
Ver. 3
Ver. N
ChipPrint
Chip 1
Chip 2
Chip 3
Chip N
ChipMakerPrint
ChipMaker 1
ChipMaker 2
ChipMaker N
ModuleMakerPrint
ModuleMaker 1
ModuleMaker 2
ModuleMaker N
ProductPrint
Product 1
Product 2
Product N
ProductMakerPrint
ProductMaker 1

# WHAT I WANT!

VersionPrint
Ver. 1
Ver. 2
Ver. 3
Ver. N
ChipPrint
Chip 1
Chip 2
Chip 3
Chip N
ChipMakerPrint
ChipMaker 1
ChipMaker 2
ChipMaker N
ModuleMakerPrint
ModuleMaker 1
ModuleMaker 2
ModuleMaker N
ProductPrint
Product 1
Product 2
Product N
ProductMakerPrint
ProductMaker 1

VersionPrint
Ver. 1
Ver. 2
Ver. 3
Ver. N
WHAT I WANT!
ChipPrint
Chip 1
Chip 2
Chip 3
Chip N
WHAT I MOSTLY GET
ChipMakerPrint
ChipMaker 1
ChipMaker 2
ChipMaker N
ModuleMakerPrint
ModuleMaker 1
ModuleMaker 2
ModuleMaker N
ProductPrint
Product 1
Product 2
Product N
ProductMakerPrint
ProductMaker 1

# My Terminology

- BTC = Bluetooth Classic
- BLE = Bluetooth Low Energy
- BDADDR = Bluetooth Device Address (like MAC address)
  - If(BTC), first 3 bytes are IEEE-assigned organizationally unique identifier (OUI), like you're probably familiar with from Ethernet MAC addresses
  - else, see next slide

# Background - BT Device Address (BDADDR)

Image & slide-in text from
https://novelbits.io/bluetooth-address-privacy-ble/

# Background - BT Device Address (BDADDR)

Image & slide-in text from
https://novelbits.io/bluetooth-address-privacy-ble/

# Background - BT Device Address (BDADDR)

Image & slide-in text from
https://novelbits.io/bluetooth-address-privacy-ble/

# Background - BT Device Address (BDADDR)

Image & slide-in text from
https://novelbits.io/bluetooth-address-privacy-ble/

# Background - BT Device Address (BDADDR)

Image & slide-in text from
https://novelbits.io/bluetooth-address-privacy-ble/

# Background - BT De

Random Static Addresses can be used in one of two ways:

- It can be assigned and fixed for the lifetime of the device
- It can be changed at bootup

However, it **cannot** be changed during runtime.

The format of Random Static Addresses looks like this:

Image & slide-in text from
https://novelbits.io/bluetooth-address-privacy-ble/

# Background - BT Device Address (BDADDR)

Image & slide-in text from
https://novelbits.io/bluetooth-address-privacy-ble/

# Background - BT Device Address (BDADDR)

Minimally Trackable *(in principle)*
(Possibly trackable using device-specific data (e.g. mDNS))

Image & slide-in text from
https://novelbits.io/bluetooth-address-privacy-ble/

# Background - B

A Resolvable Random Private address is made up of the following fields:

*Random Private Resolvable Address format (little-endian format)*

- 0 and 1 are fixed in the most significant bits (MSB)
- The next 22 bits are randomly generated
- The **prand** constitutes of these most significant 24 bits
- The lower 24 bits represent a hash value which is generated using the **prand** and the **IRK**

Image & slide-in text from
https://novelbits.io/bluetooth-address-privacy-ble/

# Background - BT Device Address (BDADDR)

Minimally Trackable *(in principle)*
(Possibly trackable using device-specific data (e.g. mDNS))

Image & slide-in text from
https://novelbits.io/bluetooth-address-privacy-ble/

# Background - BT Device Address (BDADDR)

Image & slide-in text from
https://novelbits.io/bluetooth-address-privacy-ble/

# Background - BT Device Address (BDADDR)

Image & slide-in text from
https://novelbits.io/bluetooth-address-privacy-ble/

# Background - BT Device Address (BDADDR)

Image & slide-in text from
https://novelbits.io/bluetooth-address-privacy-ble/

# Overall BDADDR Data

## My data as of 2023-10-26

- 73,047 *unique* BT Classic BDADDRs
- 8,392,322 *unique* BLE BDADDRs
  - 194,151 "public" BDADDRs
  - 2,731,623 "random static" BDADDRs
  - 4,950,427 "random resolvable" BDADDRs
  - 299,244 "random non-resolvable" BDADDRs

# So you've got a couple million datapoints...

## Is any of it useful?

- I started doing basic naïve BT data collection as a hobby during the pandemic as a way to get out of the house and drive around
  - See "It was harder to sniff Bluetooth through my mask during the Pandemic" talk from summer 2023
- The majority of my data is Linux "HCI" logs
  - HCI is the Host-Controller-Interface, between Linux, and the BT chip on my logging platform (mostly Raspberry Pi Zeros or 4Bs)
- I started work on customized 2thprinting in May 2023, once I had a chance to start looking at what the HCI logs could, and couldn't, provide for 2thprinting

# What I Want To Know:

# What Bluetooth Chip Is Inside Any Device

# Why I Want To Know It:

# So I Know if it's Vulnerable To a Firmware-Level Exploit

🧘 Passive

# 2thprinting Approaches

🚶 Mostly-Passive

🤾 Active

# Gathering: Passive vs. Mostly-Passive vs. Active

- 🧘Passive:
  - Run a sniffer, never send *any* packets
  - **BLE:** Sniffle, Ice9 Sniffer, **BTC:** Ubertooth
  - BLE ADV_IND, NONCONN_ADV_IND

# Gathering: Passive vs. Mostly-Passive vs. Active

- 🚶 *Mostly*-Passive:
  - Ask your OS "What Bluetooth devices are currently visible?"
  - It will scan for any advertisements, and *potentially,* autonomously, query some limited information about devices it finds, such as device name, class of device, etc (depends on OS & Bluetooth stack version)
    - This is most of my data

# Gathering: Passive vs. Mostly-Passive vs. Active

- 🤾‍♂️Active:
  - Send custom BLE LL / BTC LMP packets (requires custom controller firmware)
  - Connect to all connectable interfaces, query all queryable information

# 2thprint by Name

🧘‍♂️ or 🚶‍♂️

# 2thprint by Name

## aka "*NamePrint*"

- Let's start with the easy situation, where a device more or less literally tells you what it is, based on its name
  - "Ember Ceramic Mug"
  - "Nest Cam"
  - "Versa 4"
  - "User’s MacBook Pro"

# Background

## Passive Scanning 🧘

## "Mostly-passive" 🚶 Scanning

Images from https://microchipdeveloper.com/wireless:ble-link-layer-discovery

# Background

Passive Scanning 🧘

Sometimes the name will be here 🧘

"Mostly-passive" 🚶 Scanning

Images from https://microchipdeveloper.com/wireless:ble-link-layer-discovery

# Background

Passive Scanning 🧘

Sometimes the name will be here 🧘

"Mostly-passive" 🚶 Scanning

Other times the OS will ask for it 🚶 , and it comes back in a SCAN_RSP

Images from https://microchipdeveloper.com/wireless:ble-link-layer-discovery

# BT Name Behavior: One-to-One

**One device == one *unique* name && one *unique* BDADDR**

# BT Name Behavior: One-to-One

## One device == one *unique* name && one *unique* BDADDR

***Complete*** **BDADDR inclusion:**

`oura_A038F8445ED7` → `a0:38:f8:44:5e:d7` (public address)

`NOKE3K_C8D8DCF760F6` → `c8:d8:dc:f7:60:f6` (random static)

# BT Name Behavior: One-to-One

## One device == one *unique* name && one *unique* BDADDR

***Complete*** **BDADDR inclusion:**

`oura_A038F8445ED7` → `a0:38:f8:44:5e:d7` (public address)

`NOKE3K_C8D8DCF760F6` → `c8:d8:dc:f7:60:f6` (random static)

***Partial*** **BDADDR inclusion:**

`Galaxy Fit2 (987C)` → `10:39:17:36:98:7c` (public address)

`Xiaomi Smart Band 7 34C1` → `c6:05:ea:95:34:c1` (random static)

# BT Name Behavior: One-to-One

## One device == one *unique* name && one *unique* BDADDR

***Complete*** **BDADDR inclusion:**

`oura_A038F8445ED7` → `a0:38:f8:44:5e:d7` (public address)

`NOKE3K_C8D8DCF760F6` → `c8:d8:dc:f7:60:f6` (random static)

***Partial*** **BDADDR inclusion:**

`Galaxy Fit2 (987C)` → `10:39:17:36:98:7c` (public address)

`Xiaomi Smart Band 7 34C1` → `c6:05:ea:95:34:c1` (random static)

**(Presumed) serial number inclusion**

`TW370_TIA00414` → `4c:36:4e:4c:57:2a` (public address)

`CATBTNT-04 DKS02390` → `00:81:f9:7e:37:a6` (public address)

# BT Name Behavior: One-to-One

**One device == one *unique* name && one *unique* BDADDR**

# BT Name Behavior: One-to-One

## One device == one *unique* name && one *unique* BDADDR

***Possible*** **partial BDADDR disclosure?:**

```
Galaxy Watch Active2(C898) LE→c0:b9:b5:02:10:31 (random static)
```

# BT Name Behavior: One-to-Many

**One device == one *unique* name && many BDADDRs (exclusive to BLE random addresses)**

```
                                        (random resolvable)
                                        49:58:0f:57:4d:51
                                        53:84:f8:b4:3b:10
LE-sams bose speaker                    54:f3:ba:38:47:f3
                                        6d:4b:98:7a:6b:80
                                        74:21:7c:d5:c0:e0
```

# BT Name Behavior: Many-to-Many v1

## Many devices, one *shared* name, one *unique* BDADDR per device

Example: Tesla Keyfob

# BT Name Behavior: Many-to-Many v2

## Many devices, one *shared* name, *many* BDADDR per device

Example: Nvidia SHIELD

# NamePrint Data

## BTC as of 2023-10-26

- **I wrote 1,447 regex "NamePrints"**
  - 557 NamePrints matched on BTC data
- 65,451 *unique* BT Classic BDADDRs with a name
  - 65,744 *unique* {name:BDADDR} pairs
    - 23,803 NamePrint matches
      - 23,803 / 65,744 = 36% of all BTC data
  - 33,368 *unique names* in BTC data

How can there be more bdaddr:name pairs than bdaddrs? Multiple names per bdaddr! (Often due to receiving corrupt name data.)
How can there be more names than matches? Because NamePrints are regexes, one NamePrint matches multiple names

# Top 10 BTC Matches

# Summary of BTC NamePrint % applicability

- 36% of 90% of 1% (.324%) of all my data is BTC with a name that matches a NamePrint

# NamePrint Data

## BLE as of 2023-10-26

- **Made 1447 regex "NamePrints"**
  - 912 NamePrints matched on BLE data
- 188,059 *unique* BLE BDADDRs with a name
  - 195,954 *unique* name:BDADDR pairs
    - 123,865 matches
      - 39% of named data, 1.7% of all BTC data
  - 18,959 *unique names*
    - So 540 regexes match 18959/37939 ~ 50% of the names

How can there be more bdaddr:name pairs than bdaddrs? Multiple names per bdaddr! (Often due to receiving corrupt name data.)
How can there be more names than matches? Because NamePrints are regexes, one NamePrint matches multiple names

# Top 10 BLE Matches

# Summary of BTC NamePrint % applicability

- 39% of 2% of 99% (7.128%) of all my data is BLE with a name that matches a NamePrint

# What I Want

Individual Device → Device Type → Model

Firmware Version ← BT Chip ← Manufacturer

# What I Want

```
Name = "Ace2"
```

Individual Device → Device Type → Model

Firmware Version ← BT Chip ← Manufacturer

# What I Want

NamePrint
DATABASE LOOKUP
Name = "Ace2"
Individual Device
Device Type
Model
Manufacturer
BT Chip
Firmware Version

# What I Want

NamePrint
DATABASE LOOKUP
Name = "Ace2"
Ace2
Individual Device
Device Type
Model
Manufacturer
BT Chip
Firmware Version

# What I Want

*ASSUMPTION*:
Identifying a Model also allows
the identification of the Manufacturer & Device Type

# What I Want

*ASSUMPTION*:
Identifying a Model also allows
the identification of the Manufacturer & Device Type

# What I Want

*ASSUMPTION*:
Identifying a Model also allows
the identification of the Manufacturer & Device Type

# Occasionally NamePrint -> ChipPrint

- *Very rarely (as in, I've only seen this once so far ;))*, the vendor will make a big deal out of what chip it uses, and then you can quickly get a ChipPrint directly from the NamePrint!
- NamePrint="^FiiO BTR5$",ChipPrint="CSR8675"

# BTR5

## Flagship Portable High-Fidelity Bluetooth Amplifier

High Performance DAC ES9218P*2
Flagship Bluetooth chip CSR8675
FPGA clock management，dual independent crystal oscillators
Bluetooth 5.0 with full format support
Independent control chip XMOS XUF208
USB DAC supporting up to 384kHz/DSD256 native
Double-sided 2.5D glass with OLED display
3.5mm+2.5mm headphone outputs
Intelligent control with FiiO Music app
One-touch NFC pairing

VGP 2021 受賞 | VGP 2020 SUMMER 受賞

XMOS | DAC | 8675

# BTR5

## Flagship Portable High-Fidelity Bluetooth Amplifier

High Performance DAC ES9218P*2
Flagship Bluetooth chip CSR8675
FPGA clock management，dual independent crystal oscillators
Bluetooth 5.0 with full format support
Independent control chip XMOS XUF208
USB DAC supporting up to 384kHz/DSD256 native
Double-sided 2.5D glass with OLED display
3.5mm+2.5mm headphone outputs
Intelligent control with FiiO Music app
One-touch NFC pairing

VGP 2021 受賞 | VGP 2020 SUMMER 受賞

XMOS | DAC | 8675

# 2thprint by BDADDR OUI

🧘‍♀️

# 2thprint by BDADDR

## aka "OUIPrint" - https://standards-oui.ieee.org/oui/oui.txt

- BTC BDADDRs, and BLE "public" BDADDRs behave like MAC addresses, in that their first 24 bits *should be* set to an IEEE-assigned organizationally unique identifier (OUI) value, that corresponds to a particular company
  - 80:A9:97 = Apple, Inc.
  - BC:DF:58 = Google, Inc.
  - F0:39:65  = Samsung Electronics Co.,Ltd
  - F4:02:28  = SAMSUNG ELECTRO-MECHANICS(THAILAND)
- It's very clear that some vendors are using OUIs other than their own...

# Overall BDADDR Data

## IEEE OUI Applicability

- **73,047 *unique* BT Classic BDADDRs** 👍
- 8,392,322 *unique* BLE BDADDRs
  - **194,151 "public" BDADDRs** 👍
  - ~~2,731,623 "random static" BDADDRs~~
  - ~~4,950,427 "random resolvable" BDADDRs~~
  - ~~299,244 "random non-resolvable" BDADDRs~~
- So OUIPrints applicable to only ~3.2% of the BDADDRs (73,047+194,151)/(73,047+8,392,322)

📱 = phone-related, 🚗 = car-related, 🎶 = audio-related, 🍪 = chip-related, 🧩 = module-related

# BTC Data (top 20 of 593 companies seen)

| | company_by_bdaddr | | found |
|---|---|---|---|
| GPS/Watches/etc | Garmin International | | 10423 |
| 📱📺 | Samsung Electronics Co.,Ltd | *Qualcomm, Broadcom, MediaTek, Samsung* 🍪 | 6359 |
| 📱 | OnePlus Technology (Shenzhen) Co., Ltd | | 3682 |
| 🍪 | Actions Semiconductor Co.,Ltd.(Cayman Islands) | | 3036 |
| 📱 | Apple, Inc. | *Mostly Broadcom, Sometimes Apple (e.g. AirPods)* 🍪 | 2604 |
| 🎶🚗🍪🧩 | Panasonic Automotive Systems Co.,Ltd | | 2491 |
| 🎶🚗🍪🧩 | Laird Connectivity | | 2238 |
| 🎶🚗🍪🧩 | PIONEER CORPORATION | | 1977 |
| 🍪 | Intel Corporate | | 1960 |
| 🚗 | ALPSALPINE CO,.LTD | | 1919 |
| 📱 | GUANGDONG OPPO MOBILE TELECOMMUNICATIONS CORP.,LTD | | 1648 |
| 🍪🧩 | AzureWave Technology Inc. | | 1369 |
| 🍪 | Texas Instruments | | 974 |
| 📱 | Wistron Neweb Corporation | | 920 |
| 🍪🧩 | silex technology, Inc. | *Qualcomm* 🍪 | 890 |
| 🎶🚗🍪🧩 | Shinwa Industries(China) Ltd. | *Qualcomm/CSR, TI, Cypress, Sunplus* 🍪 | 858 |
| 🎶🚗🍪🧩 | MITSUMI ELECTRIC CO.,LTD. | *CSR->Qualcomm* 🍪 | 845 |
| 🎶🚗🍪🧩 | PARROT SA | *TI* 🍪 | 741 |
| 📱 integrator? | Wingtech Mobile Communications Co., Ltd. | *MediaTek?* 🍪 | 690 |
| 🎶🚗🍪 | Sunplus Technology Co., Ltd. | | 606 |

📱 = phone-related, 🚗 = car-related, 🎶 = audio-related, 🍪 = chip-related, 🧩 = module-related

# BTC Data (top 20 of 593 companies seen)

| | company_by_bdaddr | | found |
|---|---|---|---|
| GPS/Watches/etc | Garmin International | | 10423 |
| 📱📺 | Samsung Electronics Co.,Ltd | *Qualcomm, Broadcom, MediaTek, Samsung* 🍪 | 6359 |
| 📱 | OnePlus Technology (Shenzhen) Co., Ltd | | 3682 |
| 🍪 | Actions Semiconductor Co.,Ltd.(Cayman Islands) | | 3036 |
| 📱 | Apple, Inc. | *Mostly Broadcom, Sometimes Apple (e.g. AirPods)* 🍪 | 2604 |
| 🎶🚗🍪🧩 | Panasonic Automotive Systems Co.,Ltd | | 2491 |
| 🎶🚗🍪🧩 | Laird Connectivity | | 2238 |
| 🎶🚗🍪🧩 | PIONEER CORPORATION | | |
| 🍪 | Intel Corporate | | |
| 🚗 | ALPSALPINE CO,.LTD | | 1919 |
| 📱 | GUANGDONG OPPO MOBILE TELECOMMUNICATIONS CORP.,LTD | | 1648 |
| 🍪🧩 | AzureWave Technology Inc. | | 1369 |
| 🍪 | Texas Instruments | | 974 |
| 📱 | Wistron Neweb Corporation | | 920 |
| 🍪🧩 | silex technology, Inc. | *Qualcomm* 🍪 | 890 |
| 🎶🚗🍪🧩 | Shinwa Industries(China) Ltd. | *Qualcomm/CSR, TI, Cypress, Sunplus* 🍪 | 858 |
| 🎶🚗🍪🧩 | MITSUMI ELECTRIC CO.,LTD. | *CSR->Qualcomm* 🍪 | 845 |
| 🎶🚗🍪🧩 | PARROT SA | *TI* 🍪 | 741 |
| 📱 integrator? | Wingtech Mobile Communications Co., Ltd. | *MediaTek?* 🍪 | 690 |
| 🎶🚗🍪 | Sunplus Technology Co., Ltd. | | 606 |

Note! My data is skewed towards vehicles, because I like to put my sniffers over freeways!

# BLE Data (top 20 of 608 companies seen)

| | company_by_bdaddr | | found | |
|---|---|---|---|---|
| 🍪 | Texas Instruments | | 31799 | |
| 🎶 | VXi Corporation | | 25190 | |
| 📱📺 | Samsung Electronics Co.,Ltd | *Qualcomm, Broadcom, MediaTek, Samsung* 🍪 | 12340 | |
| 🎶 | Bose Corporation | | 8475 | |
| 🐀⌨️ | Logitech, Inc | | 4980 | |
| 🎶🚗🍪🧩 | Cambridge Mobile Telematics, Inc. | *CSR->Qualcomm* 🍪 | 4958 | |
| 📱 | Apple, Inc. | *Mostly Broadcom, Sometimes Apple (e.g. AirPods)* 🍪 | 4821 | |
| 🍪 | Silicon Laboratories | | 3802 | |
| 🍪 | Espressif Inc. | | 2975 | |
| BLE beacons | Shenzhen Minew Technologies Co., Ltd. | | 2462 | |
| 🍪🧩 | Murata Manufacturing Co., Ltd. | *Nordic, Cypress, CSR, Onsemi, Dialog* 🍪 | 2276 | |
| 🍪 | Telink Semiconductor (Taipei) Co. Ltd. | | 1943 | |
| 🍪🧩 + 🍪? | Shenzhen Jingxun Software Telecommunication Technology Co.,Ltd | | 1879 | *RealTek, Airoha* 🍪 |
| GPS/Watches/etc | Garmin International | | 1651 | |
| 🍪🧩 | Sunitec Enterprise Co.,Ltd | *Broadcom* 🍪 | 1426 | |
| BLE beacons | Aruba, a Hewlett Packard Enterprise Company | | 1388 | |
| LED lights & sensors | Shenzhen Intellirocks Tech co.,ltd | *Telink* 🍪 | 1139 | |
| Teslas | Shanghai Rui Rui Communication Technology Co.Ltd. | | 1070 | (TI bought OUI?) |
| 🎶🚗🍪🧩 | ALPSALPINE CO,.LTD | | 969 | |
| 🎶 Hearing aids | Starkey Labs Inc. | | 934 | |

# Of what I want to know

- If the BDADDR maps to a silicon vendor, it suggests the product manufacturer either a) didn't have their own IEEE OUI registration, or b) didn't care enough about the product to use it. And thus we have an idea of what chip maker is being used. E.g.

| | | |
|---|---|---|
| "Texas Instruments" | "Silicon Laboratories" | "Intel Corporate" |
| "Telink Semiconductor (Taipei) Co. Ltd." | "Cambridge Silicon Radio" | "Espressif Inc." |
| "Nordic Semiconductor ASA" | "NXP Semiconductors" | "NXP France Semiconductors France" |
| "NXP Semiconductor (Tianjin) LTD." | "NXP (China) Management Ltd." | "Microchip Technology Inc." |
| "Broadcom" | "Qualcomm Technologies International, Ltd. (QTIL)" | "REALTEK SEMICONDUCTOR CORP." |

- For some other names (like module-makers), we might know that there's only 1 or 2 chips they ever use, though of course that can change
  - Need to collect that data over time

# The First Traces

## Of what I want to know

- If the BDADDR maps to a silicon vendor, it suggests the product manufacturer either a) didn't have their own IEEE OUI registration, or b) didn't care enough about the product to use it. And thus we have an idea of what chip maker is being used. E.g.

| | | |
|---|---|---|
| "Texas Instruments" | "Silicon Laboratories" | "Intel Corporate" |
| "Telink Semiconductor (Taipei) Co. Ltd." | "Cambridge Silicon Radio" | "Espressif Inc." |
| "Nordic Semiconductor ASA" | "NXP Semiconductors" | "NXP France Semiconductors France" |
| "NXP Semiconductor (Tianjin) LTD." | "NXP (China) Management Ltd." | "Microchip Technology Inc." |
| "Broadcom" | "Qualcomm Technologies International, Ltd. (QTIL)" | "REALTEK SEMICONDUCTOR CORP." |

- For some other names (like module-makers), we might know that there's only 1 or 2 chips they ever use, though of course that can change
  - Need to collect that data over time

# What I Want

Individual Device → Device Type → Model

Firmware Version ← BT Chip ← Manufacturer

# What I Want

BDADDR = **FC:A8:9B:**11:22:33

Individual Device → Device Type → Model

Firmware Version ← BT Chip ← Manufacturer

# What I Want

IEEE OUI
DATABASE LOOKUP

BDADDR = **FC:A8:9B:**11:22:33

Individual Device → Device Type → Model

Firmware Version ← BT Chip ← Manufacturer

# What I Want

IEEE OUI
DATABASE LOOKUP

BDADDR = **FC:A8:9B:**11:22:33

Individual Device → Device Type → Model

Firmware Version ← BT Chip ← Manufacturer

"Texas Instruments"

# What I Want

IEEE OUI
DATABASE LOOKUP

BDADDR = **FC:A8:9B:**11:22:33

Individual Device → Device Type → Model

Firmware Version ← BT Chip ← Manufacturer

"Texas Instruments"

# What I Want

IEEE OUI
DATABASE LOOKUP

BDADDR = **FC:A8:9B:**11:22:33

Individual Device → Device Type → Model

Firmware Version ← BT Chip ← Manufacturer

"Texas Instruments"

*ASSUMPTION*:

OUIPrint == ChipPrint, for OUI == {Silicon Vendor OUIs}

# What I Want

*ASSUMPTION*:

OUIPrint == ChipPrint, for OUI == {Silicon Vendor OUIs}

# What I Want

*ASSUMPTION*:

OUIPrint == ChipPrint, for OUI == {Silicon Vendor OUIs}

# 2thprint by Link Layer Version Info

🚶‍♂️ or 🤾‍♂️

# 2thprint by Link Layer Version Information

## LL_VERSION_IND (BLE), LMP_version_res (BTC)

- BTC has Link Management Protocol (LMP) and BLE has Link-Layer (LL) Control packets, that can be sent to request some chip and firmware version information
- These can be sent/received *without* any sort of BT pairing/bonding!

# 2thprint by

## LMP_version_res (BTC)

| M/O | PDU | Contents |
|---|---|---|
| M | LMP_version_req | VersNr<br>CompId<br>SubVersNr |
| M | LMP_version_res | VersNr<br>CompId<br>SubVersNr |

*Table 4.26: PDUs used for LMP version request*

*Sequence 77: Request for LMP version*

From BT
Core Spec v4.2

# 2thprint by
## LL_VERSION_IND (BLE)

### *2.4.2.13 LL_VERSION_IND*

The format of the CtrData field is shown in Figure 2.23.

| CtrData | | |
|---|---|---|
| VersNr<br>(1 octet) | CompId<br>(2 octets) | SubVersNr<br>(2 octets) |

*Figure 2.23: CtrData field of the LL_VERSION_IND PDU*

The LL_VERSION_IND CtrData consists of three fields:

- VersNr field shall contain the version of the Bluetooth Controller specification (see Bluetooth Assigned Numbers).
- CompId field shall contain the company identifier of the manufacturer of the Bluetooth Controller (see Bluetooth Assigned Numbers).
- SubVersNr field shall contain a unique value for each implementation or revision of an implementation of the Bluetooth Controller.

From BT
Core Spec v4.2

# Prior Work

## "ESPwn32: Hacking with ESP32 System-on-Chips"

- Oh no! I got scooped! (Or did I?)
- [1] by Cayre et al. from May 2023 recognized that LL_VERSION_IND packets contains useful information
- But it subsequently *assumes* this information is *sufficient* for vulnerability applicability assessment, without offering any proof / data analysis

**Remote BLE stack fingerprinting !**

[1] https://wootconference.org/papers/woot23-paper22.pdf

# Prior Work

## "ESPwn32: Hacking with ESP32 System-on-Chips"

- Oh no! I got scooped! (Or did I?)
- [1] by Cayre et al. from May 2023 recognized that LL_VERSION_IND packets contains useful information
- But it subsequently *assumes* this information is *sufficient* for vulnerability applicability assessment, without offering any proof / data analysis

**Remote BLE stack fingerprinting !**

[1] https://wootconference.org/papers/woot23-paper22.pdf

# LMP/LL 2thprints useful, but not definitive

- It turns out that for MediaTek (CID = 70), the most common value for the sub version is 0

| device_BT_CID | lmp_sub_version | frequency |
|---|---|---|
| 70 | 0 | 774 |
| 70 | 288 | 189 |
| 70 | 4648 | 86 |
| 70 | 4101 | 62 |
| 70 | 2051 | 15 |
| 70 | 1571 | 14 |
| 70 | 613 | 11 |
| 70 | 533 | 10 |
| 70 | 2344 | 10 |
| 70 | 1030 | 9 |
| 70 | 4391 | 8 |
| 70 | 1797 | 7 |
| 70 | 4135 | 5 |
| 70 | 304 | 4 |
| 70 | 726 | 4 |
| 70 | 791 | 4 |
| 70 | 1560 | 4 |
| 70 | 1817 | 4 |

...

| device_BT_CID | ll_sub_version | frequency |
|---|---|---|
| 70 | 0 | 36 |
| 70 | 534 | 8 |
| 70 | 4101 | 8 |
| 70 | 4648 | 6 |
| 70 | 2051 | 5 |
| 70 | 4355 | 3 |
| 70 | 4373 | 2 |
| 70 | 288 | 1 |
| 70 | 304 | 1 |
| 70 | 546 | 1 |
| 70 | 776 | 1 |
| 70 | 791 | 1 |
| 70 | 1033 | 1 |
| 70 | 1042 | 1 |
| 70 | 1045 | 1 |
| 70 | 1568 | 1 |
| 70 | 4097 | 1 |
| 70 | 4116 | 1 |

# LMP/LL 2thprints useful, but not definitive

- It turns out that for MediaTek (CID = 70), the most common value for the sub version is 0

| device_BT_CID | lmp_sub_version | frequency |
|---|---|---|
| 70 | 0 | 774 |
| 70 | 288 | 189 |
| 70 | 4648 | 86 |
| 70 | 4101 | 62 |
| 70 | 2051 | 15 |
| 70 | 1571 | 14 |
| 70 | 613 | 11 |
| 70 | 533 | 10 |
| 70 | 2344 | 10 |
| 70 | 1030 | 9 |
| 70 | 4391 | 8 |
| 70 | 1797 | 7 |
| 70 | 4135 | 5 |
| 70 | 304 | 4 |
| 70 | 726 | 4 |
| 70 | 791 | 4 |
| 70 | 1560 | 4 |
| 70 | 1817 | 4 |

...

| device_BT_CID | ll_sub_version | frequency |
|---|---|---|
| 70 | 0 | 36 |
| 70 | 534 | 8 |
| 70 | 4101 | 8 |
| 70 | 4648 | 6 |
| 70 | 2051 | 5 |
| 70 | 4355 | 3 |
| 70 | 4373 | 2 |
| 70 | 288 | 1 |
| 70 | 304 | 1 |
| 70 | 546 | 1 |
| 70 | 776 | 1 |
| 70 | 791 | 1 |
| 70 | 1033 | 1 |
| 70 | 1042 | 1 |
| 70 | 1045 | 1 |
| 70 | 1568 | 1 |
| 70 | 4097 | 1 |
| 70 | 4116 | 1 |

# LMP/LL 2thprints useful, but not definitive

- It turns out that for MediaTek (CID = 70), the most common value for the sub version is 0

| device_BT_CID | lmp_sub_version | frequency |
|---|---|---|
| 70 | 0 | 774 |
| 70 | 288 | 189 |
| 70 | 4648 | 86 |
| 70 | 4101 | 62 |
| 70 | 2051 | 15 |
| 70 | 1571 | 14 |
| 70 | 613 | 11 |
| 70 | 533 | 10 |
| 70 | 2344 | 10 |
| 70 | 1030 | 9 |
| 70 | 4391 | 8 |
| 70 | 1797 | 7 |
| 70 | 4135 | 5 |
| 70 | 304 | 4 |
| 70 | 726 | 4 |
| 70 | 791 | 4 |
| 70 | 1560 | 4 |
| 70 | 1817 | 4 |

...

| device_BT_CID | ll_sub_version | frequency |
|---|---|---|
| 70 | 0 | 36 |
| 70 | 534 | 8 |
| 70 | 4101 | 8 |
| 70 | 4648 | 6 |
| 70 | 2051 | 5 |
| 70 | 4355 | 3 |
| 70 | 4373 | 2 |
| 70 | 288 | 1 |
| 70 | 304 | 1 |
| 70 | 546 | 1 |
| 70 | 776 | 1 |
| 70 | 791 | 1 |
| 70 | 1033 | 1 |
| 70 | 1042 | 1 |
| 70 | 1045 | 1 |
| 70 | 1568 | 1 |
| 70 | 4097 | 1 |
| 70 | 4116 | 1 |

# LMP/LL 2thprints useful, but not definitive

- It turns out that for MediaTek (CID = 70), the most common value for the sub version is 0

| device_BT_CID | lmp_sub_version | frequency |
|---|---|---|
| 70 | 0 | 774 |
| 70 | 288 | 189 |
| 70 | 4648 | 86 |
| 70 | | |
| 70 | | |
| 70 | |

# OnePlus Pad

Halo Green | 8 GB RAM + 128 GB Storage

CA$649.99

Color: Halo Green

ROM

8 GB RAM + 128 GB Storage

# ut not definitive

most common value for the sub

| | | |
|---|---|---|
| 70 | 4101 | 62 |
| 70 | 2051 | 15 |
| 70 | 1571 | 14 |
| 70 | 613 | 11 |
| 70 | 533 | 10 |
| 70 | 2344 | 10 |
| 70 | 1030 | 9 |
| 70 | 4391 | 8 |
| 70 | 1797 | 7 |
| 70 | 4135 | 5 |
| 70 | 304 | 4 |
| 70 | 726 | 4 |
| 70 | 791 | 4 |
| 70 | 1560 | 4 |
| 70 | 1817 | 4 |

...

| device_BT_CID | ll_sub_version | frequency |
|---|---|---|
| 70 | 0 | 36 |
| 70 | 534 | 8 |
| 70 | 4101 | 8 |
| 70 | 4648 | 6 |
| 70 | 2051 | 5 |
| 70 | 4355 | 3 |
| 70 | 4373 | 2 |
| 70 | 288 | 1 |
| 70 | 304 | 1 |
| 70 | 546 | 1 |
| 70 | 776 | 1 |
| 70 | 791 | 1 |
| 70 | 1033 | 1 |
| 70 | 1042 | 1 |
| 70 | 1045 | 1 |
| 70 | 1568 | 1 |
| 70 | 4097 | 1 |
| 70 | 4116 | 1 |

# OnePlus Pad

Halo Green | 8 GB RAM + 128 GB Storage

CA$649.99

Color: Halo Green

ROM

8 GB RAM + 128 GB Storage

OnePlus Trade-In Open for Every

| | | |
|---|---|---|
| 70 | 4101 | 62 |
| 70 | 2051 | 15 |
| 70 | 1571 | 14 |
| 70 | 613 | 11 |
| 70 | 533 | 10 |
| 70 | 2344 | 10 |
| 70 | 1030 | 9 |
| 70 | 4391 | 8 |
| 70 | 1797 | 7 |
| 70 | 4135 | 5 |
| 70 | 304 | 4 |
| 70 | 726 | 4 |
| 70 | 791 | 4 |
| 70 | 1560 | 4 |
| 70 | 1817 | 4 |

...

# OnePlus Pad

Halo Green | 8 GB RAM + 128 GB Storage

CA$649.99

Color: Halo Green

ROM

8 GB RAM + 128 GB Storage

OnePlus Trade-In Open for Every

62
15
14
11
10
10
9
8
7
5
4
4
4
4
4

# OnePlus Pad

Halo Green | 8 GB RAM + 128 GB Storage

CA$649.99

Color: Halo Green

ROM

8 GB RAM + 128 GB Storage

# How to Send Packets?

- Sweyntooth[1] (2020, BLE-only) & Braktooth[2] (2022, BTC-only)
  - Provide a way in Python and C respectively to create & send arbitrary link-layer packets in an arbitrary order

[1] https://asset-group.github.io/disclosures/sweyntooth/
[2] https://asset-group.github.io/disclosures/braktooth/

# (Link Layer) Version Number

## VersNr (1 byte)

| CtrData | | |
|---|---|---|
| VersNr (1 octet) | CompId (2 octets) | SubVersNr (2 octets) |

*Figure 2.23: CtrData field of the LL_VERSION_IND PDU*

Last Modified: 2023-02-21

| Core Specification Name | Version |
|---|---|
| Bluetooth® Core Specification 1.0b (Withdrawn) | 0x00 |
| Bluetooth® Core Specification 1.1 (Withdrawn) | 0x01 |
| Bluetooth® Core Specification 1.2 (Withdrawn) | 0x02 |
| Bluetooth® Core Specification 2.0+EDR (Withdrawn) | 0x03 |
| Bluetooth® Core Specification 2.1+EDR (Withdrawn) | 0x04 |
| Bluetooth® Core Specification 3.0+HS (Withdrawn) | 0x05 |
| Bluetooth® Core Specification 4.0 (Withdrawn) | 0x06 |
| Bluetooth® Core Specification 4.1 (Deprecated) | 0x07 |
| Bluetooth® Core Specification 4.2 | 0x08 |
| Bluetooth® Core Specification 5.0 | 0x09 |
| Bluetooth® Core Specification 5.1 | 0x0A |
| Bluetooth® Core Specification 5.2 | 0x0B |
| Bluetooth® Core Specification 5.3 | 0x0C |
| Bluetooth® Core Specification 5.4 | 0x0D |

From BT
Assigned Numbers document

# (Link Layer) Company Identifier

## CompId (2 bytes) - 3330 defined at time of writing

```yaml
company_identifiers:

  - value: 0x0CA5
    name: 'Princess Cruise Lines, Ltd.'

  - value: 0x0CA4
    name: 'MITSUBISHI ELECTRIC LIGHTING CO, LTD'

  - value: 0x0CA3
    name: 'MAQUET GmbH'

  - value: 0x0CA2
    name: 'XSENSE LTD'

  - value: 0x0CA1
    name: 'YAMAHA MOTOR CO.,LTD.'

  - value: 0x0CA0
    name: 'BIGBEN'

  - value: 0x0C9F
    name: 'Dragonfly Energy Corp.'

  - value: 0x0C9E
```

From htt y!

```yaml
  name: 'H+B Hightech GmbH'

- value: 0x0C95
  name: 'Gemstone Lights Canada Ltd.'

- value: 0x0C94
  name: 'Baxter Healthcare Corporation'

- value: 0x0C93
  name: 'Movesense Oy'

- value: 0x0C92
  name: 'Kesseböhmer Ergonomietechnik GmbH'

- value: 0x0C91
  name: 'Yashu Systems'

- value: 0x0C90
  name: 'WESCO AG'

- value: 0x0C8F
  name: 'Radar Automobile Sales(Shandong)Co.,Ltd.'

- value: 0x0C8E
  name: 'Technocon Engineering Ltd.'

- value: 0x0C8D
  name: 'tonies GmbH'

- value: 0x0C8C
  name: 'T-Mobile USA'
```

From https://bitbucket.org/bluetooth-SIG/public/src/main/company_identifiers/company_identifiers.yaml, updated very frequently!

# (Link Layer) Company Identifier

## CompId (2 bytes) - 3330 defined at time of writing

```yaml
company_identifiers:

  - value: 0x0CA5
    name: 'Princess Cruise Lines, Ltd.'

  - value: 0x0CA4
    name: 'MITSUBISHI ELECTRIC LIGHTING CO, LTD'

  - value: 0x0CA3
    name: 'MAQUET GmbH'

  - value: 0x0CA2
    name: 'XSENSE LTD'

  - value: 0x0CA1
    name: 'YAMAHA MOTOR CO.,LTD.'

  - value: 0x0CA0
    name: 'BIGBEN'

  - value: 0x0C9F
    name: 'Dragonfly Energy Corp.'

  - value: 0x0C9E
```

From htt y!

# (Link Layer) Company Identifier

## CompId (2 bytes) - 3330 defined at time of writing

```
company_identifiers:

  - value: 0x0CA5
    name: 'Princess Cruise Lines, Ltd.'

  - value: 0x0CA4
    name: 'MITSUBISHI ELECTRIC LIGHTING CO, LTD'

  - value: 0x0CA3
    name: 'MAQUET GmbH'

  - value: 0x0CA2
    name: 'XSENSE LTD'

  - value: 0x0CA1
    name: 'YAMAHA MOTOR CO.,LTD.'

  - value: 0x0CA0
    name: 'BIGBEN'

  - value: 0x0C9F
    name: 'Dragonfly Energy Corp.'

  - value: 0x0C9E
```

From htt y!

# (Link Layer) Compar r

## CompId (2 bytes) - 3330 define iting

```
company_identifiers:

  - value: 0x0CA5
    name: 'Princess Cruise Lines, Ltd.'

  - value: 0x0CA4
    name: 'MITSUBISHI ELECTRIC LIGHTING CO, LTD'

  - value: 0x0CA3
    name: 'MAQUET GmbH'

  - value: 0x0CA2
    name: 'XSENSE LTD'

  - value: 0x0CA1
    name: 'YAMAHA MOTOR CO.,LTD.'

  - value: 0x0CA0
    name: 'BIGBEN'

  - value: 0x0C9F
    name: 'Dragonfly Energy Corp.'

  - value: 0x0C9E
```

From htt y!

# (Link Layer) Compar r

## CompId (2 bytes) - 3330 define iting

Top Side of PCB View

```
company_identifiers:

  - value: 0x0CA5
    name: 'Princess Cruise Lines, Ltd.'

  - value: 0x0CA4
    name: 'MITSUBISHI ELECTRIC LIGHTING CO, LTD'

  - value: 0x0CA3
    name: 'MAQUET GmbH'

  - value: 0x0CA2
    name: 'XSENSE LTD'

  - value: 0x0CA1
    name: 'YAMAHA MOTOR CO.,LTD.'

  - value: 0x0CA0
    name: 'BIGBEN'

  - value: 0x0C9F
    name: 'Dragonfly Energy Corp.'

  - value: 0x0C9E
```

From htt y!

# (Link Layer) Compar r

## CompId (2 bytes) - 3330 define iting

company_identifiers:

## Hands Full? No Problem

You walk along the corridor and voila! Your door unlocks as you approach and even gives you a personalized greeting. You'll enjoy keyless entry with automatic door locks every time you enter your stateroom.

# (Link Layer) Compar r

## CompId (2 bytes) - 3330 define iting

company_identifiers:

## Hands Full? No Problem

You walk along the corridor and voila! Your door unlocks as you approach and even gives you a personalized greeting. You'll enjoy keyless entry with automatic door locks every time you enter your stateroom.

https://hardwear.io/netherlands-2022/speakers/sultan-khan.php

# (Link Layer) Compar r

## CompId (2 bytes) - 3330 define iting

company_identifiers:

## Hands Full? No Problem

You walk along the corridor and voila! Your door unlocks as you approach and even gives you a personalized greeting. You'll enjoy keyless entry with automatic door locks every time you enter your stateroom.

https://hardwear.io/netherlands-2022/speakers/sultan-khan.php

# Vendor Prevalence

## BLE - 2023-10-26

- In general though this seems to have a higher prevalence of silicon makers than other BT company/vendor ID fields, so a better S/N ratio for what I want to know
- I have only seen ~ 11466 / 92854 (12.3%) success in my LL 2thprint log
  - Future work will take RSSI into account

| comp_by_CompId | count |
|---|---|
| Broadcom Corporation | 8108 |
| Apple, Inc. | 1466 |
| Nordic Semiconductor ASA | 163 |
| Qualcomm Technologies International, Ltd. (QTIL) | 121 |
| MediaTek, Inc. | 119 |
| Qualcomm | 114 |
| Cypress Semiconductor | 85 |
| Texas Instruments Inc. | 83 |
| Samsung Electronics Co. Ltd. | 60 |
| Dialog Semiconductor B.V. | 41 |
| RivieraWaves S.A.S | 32 |
| Telink Semiconductor Co. Ltd | 29 |
| Bestechnic(Shanghai),Ltd | 28 |
| Realtek Semiconductor Corporation | 28 |
| ST Microelectronics | 26 |
| Marvell Technology Group Ltd. | 19 |
| Casambi Technologies Oy | 15 |
| Zhuhai Jieli technology Co.,Ltd | 15 |
| Airoha Technology Corp. | 14 |
| Silicon Laboratories | 13 |
| Ambiq | 11 |
| Atheros Communications, Inc. | 10 |
| Universal Electronics, Inc. | 8 |
| Shenzhen Goodix Technology Co., Ltd | 6 |

# Vendor Prevalence

## BLE - 2023-10-26

- In general though this seems to have a higher prevalence of silicon makers than other BT company/vendor ID fields, so a better S/N ratio for what I want to know
- I have only seen ~ 11466 / 92854 (12.3%) success in my LL 2thprint log
    - Future work will take RSSI into account

| comp_by_CompId | count |
|---|---|
| Broadcom Corporation | 8108 |
| Apple, Inc. | 1466 |
| Nordic Semiconductor ASA | 163 |
| Qualcomm Technologies International, Ltd. (QTIL) | 121 |
| MediaTek, Inc. | 119 |
| Qualcomm | 114 |
| Cypress Semiconductor | 85 |
| Texas Instruments Inc. | 83 |
| Samsung Electronics Co. Ltd. | 60 |
| Dialog Semiconductor B.V. | 41 |
| RivieraWaves S.A.S | 32 |
| Telink Semiconductor Co. Ltd | 29 |
| Bestechnic(Shanghai),Ltd | 28 |
| Realtek Semiconductor Corporation | 28 |
| ST Microelectronics | 26 |
| Marvell Technology Group Ltd. | 19 |
| Casambi Technologies Oy | 15 |
| Zhuhai Jieli technology Co.,Ltd | 15 |
| Airoha Technology Corp. | 14 |
| Silicon Laboratories | 13 |
| Ambiq | 11 |
| Atheros Communications, Inc. | 10 |
| Universal Electronics, Inc. | 8 |
| Shenzhen Goodix Technology Co., Ltd | 6 |
| Yichip Microelectronics (Hangzhou) Co.,Ltd. | 5 |
| UNKNOWN_COMP_BY_BT_CID | 5 |
| beken | 5 |
| Actions (Zhuhai) Technology Co., Limited | 5 |
| PHYPLUS Inc | 4 |
| Intel Corp. | 4 |
| Toshiba Corp. | 3 |
| Shanghai wuqi microelectronics Co.,Ltd | 2 |
| Barrot Technology Co.,Ltd. | 2 |
| Hong Kong HunterSun Electronic Limited | 2 |
| WuXi Vimicro | 2 |
| NXP B.V. | 2 |
| Ingchips Technology Co., Ltd. | 2 |
| LAPIS Semiconductor Co.,Ltd | 2 |
| Bluetrum Technology Co.,Ltd | 2 |
| ON Semiconductor | 1 |
| MindTree Ltd. | 1 |

# Vendor Prevalence

## BTC - 2023-10-26

- In general though this seems to have a higher prevalence of silicon makers than other BT company/vendor ID fields, so a better S/N ratio for what I want to know
- I have only seen ~ 7916 / 25125 (31.5%) success in my LMP 2thprint log
  - Future work will take RSSI into account

| comp_by_CompId | count |
|---|---|
| Qualcomm | 35 |
| MediaTek, Inc. | 26 |
| Broadcom Corporation | 25 |
| Intel Corp. | 8 |
| Qualcomm Technologies International, Ltd. (QTIL) | 7 |
| Zhuhai Jieli technology Co.,Ltd | 7 |
| Realtek Semiconductor Corporation | 6 |
| Cypress Semiconductor | 3 |
| Marvell Technology Group Ltd. | 2 |
| Samsung Electronics Co. Ltd. | 2 |
| RivieraWaves S.A.S | 2 |
| Bluegiga | 1 |
| Equinux AG | 1 |
| G-wearables inc. | 1 |
| Anova Applied Electronics | 1 |
| Toshiba Corp. | 1 |
| Texas Instruments Inc. | 1 |
| Lumens For Less, Inc | 1 |
| Nokia Mobile Phones | 1 |
| Actions (Zhuhai) Technology Co., Limited | 1 |
| Shenzhen Feasycom Technology Co., Ltd. | 1 |

# Vendor Prevalence

## BTC - 2023-10-26

- In general though this seems to have a higher prevalence of silicon makers than other BT company/vendor ID fields, so a better S/N ratio for what I want to know

- I have only seen ~ 7916 / 25125 (31.5%) success in my LMP 2thprint log

  - Future work will take RSSI into account

Bitflip!

| comp_by_CompId | count |
|---|---|
| Qualcomm | 35 |
| MediaTek, Inc. | 26 |
| Broadcom Corporation | 25 |
| Intel Corp. | 8 |
| Qualcomm Technologies International, Ltd. (QTIL) | 7 |
| Zhuhai Jieli technology Co.,Ltd | 7 |
| Realtek Semiconductor Corporation | 6 |
| Cypress Semiconductor | 3 |
| Marvell Technology Group Ltd. | 2 |
| Samsung Electronics Co. Ltd. | 2 |
| RivieraWaves S.A.S | 2 |
| Bluegiga | 1 |
| Equinux AG | 1 |
| G-wearables inc. | 1 |
| Anova Applied Electronics | 1 |
| Toshiba Corp. | 1 |
| Texas Instruments Inc. | 1 |
| Lumens For Less, Inc | 1 |
| Nokia Mobile Phones | 1 |
| Actions (Zhuhai) Technology Co., Limited | 1 |
| Shenzhen Feasycom Technology Co., Ltd. | 1 |

# (Link Layer) Sub-Version Number

## SubVersNr (2 bytes)

- The vendor gets to make up whatever value they want here!

From BT
Assigned Numbers document

# Does SubVersNr Imply OS/Firmware Version?

- **Hypothesis:** SubVersNr will increment per OS/firmware update, and thus can be used to infer the OS/firmware version
  - Conclusion:
    - **Rejected** for Broadcom

# Does SubVersNr Imply OS/Firmware Version?

- From BlueZ 5.66 monitor/packet.c

```
} broadcom_usb_subversion_table[] = {
  { 0x210b, "BCM43142A0" },  /* 001.001.011 */
  { 0x2112, "BCM4314A0"},  /* 001.001.018 */
  { 0x2118, "BCM20702A0" },  /* 001.001.024 */
  { 0x2126, "BCM4335A0"},  /* 001.001.038 */
  { 0x220e, "BCM20702A1" },  /* 001.002.014 */
  { 0x230f, "BCM4354A2"},  /* 001.003.015 */
  { 0x4106, "BCM4335B0"},  /* 002.001.006 */
  { 0x410e, "BCM20702B0" },  /* 002.001.014 */
  { 0x6109, "BCM4335C0"},  /* 003.001.009 */
  { 0x610c, "BCM4354"},  /* 003.001.012 */
```

# Oh right...I remember something now...

- InternalBlue has firmware files per Broadcom chip, and they're ordered by numbers that look like those SubVersions!

C

•

- fw_0x2033.py
- fw_0x203a.py
- fw_0x2056.py
- fw_0x21a9.py
- fw_0x21d0.py
- fw_0x2209.py
- fw_0x220b.py
- fw_0x220c.py
- fw_0x220e.py
- fw_0x2230.py
- fw_0x240f.py
- fw_0x3032.py
- fw_0x4196.py
- fw_0x4208.py
- fw_0x420e.py
- fw_0x420e_iphone.py
- fw_0x4228.py
- fw_0x422a.py
- fw_0x6103.py
- fw_0x6109.py
- fw_0x6119.py

C

•

- fw_0x2033.py
- fw_0x203a.py
- fw_0x2056.py
- fw_0x21a9.py
- fw_0x21d0.py
- fw_0x2209.py

```
} broadcom_usb_subversion_table[] = {
  { 0x210b, "BCM43142A0" }, /* 001.001.011 */
  { 0x2112, "BCM4314A0"  }, /* 001.001.018 */
  { 0x2118, "BCM20702A0" }, /* 001.001.024 */
  { 0x2126, "BCM4335A0"  }, /* 001.001.038 */
  { 0x220e, "BCM20702A1" }, /* 001.002.014 */
  { 0x230f, "BCM4354A2"  }, /* 001.003.015 */
  { 0x4106, "BCM4335B0"  }, /* 002.001.006 */
  { 0x410e, "BCM20702B0" }, /* 002.001.014 */
  { 0x6109, "BCM4335C0"  }, /* 003.001.009 */
  { 0x610c, "BCM4354" }, /* 003.001.012 */
```

- fw_0x4196.py
- fw_0x4208.py
- fw_0x420e.py
- fw_0x420e_iphone.py
- fw_0x4228.py
- fw_0x422a.py
- fw_0x6103.py
- fw_0x6109.py
- fw_0x6119.py

# Broadcom

- So at least for Broadcom, the SubVersion is used to store a silicon chip *(& stepping revision)* information!
- ***The ideal ChipPrint!*** 🥳
- (I would have preferred a VersionPrint, but oh well...)

# Does SubVersNr Imply OS/Firmware Version?

- **Hypothesis:** SubVersNr will increment per OS/firmware update, and thus can be used to infer the OS/firmware version
  - Conclusion:
    - Weak reject for MediaTek, which seems to default to SubVersNr == 0, but where there are a range of versions seen, which probably then depends on the device maker's behavior
    - *Requires more investigation*

# Does SubVersNr Imply OS/Firmware Version?

- **Hypothesis:** SubVersNr will increment per OS/firmware update, and thus can be used to infer the OS/firmware version
  - Overall Conclusion:
    - Requires more investigation!
      - Fundamentally requires hands on with a device + manually updating firmware and observing how the sub-version-number changes

# 2thprint by Link Layer Packet Combos

# Link Layer Version Information is not the only game in town

- If we want to be more nmap-OS-fingerprint-like, it makes sense to hit the target with multiple packet types, in different orders, and see how it responds
- This was the original idea for 2thprinting...and yet...I haven't actually analyzed this data yet ¯_(ツ)_/¯ (mostly because it feels like I haven't found the right balance yet between speed and useful signal)
- LMP packet types: lmp_features_req, LMP_features_req_ext, LMP_version_req, LMP_name_req, LMP_switch_req, LMP_ping_req, LMP_encryption_key_size_req, *malformed* LMP_features_req, *malformed* LMP_features_req_ext
- BLE LL packet types: LL_VERSION_IND, LL_LENGTH_REQ, LL_PING_REQ, LL_FEATURE_REQ

# 2thprint by Device ID Profile

# BTC EIR Device ID

- Some devices seem to include Device ID information as shown here in BTC Extended Inquiry Response packets
  - Despite the fact that I can't find anything in the spec saying that's supposed to be a thing? (The "Supplement to the Bluetooth Core Specification" doesn't list it as valid...)
- If VendorIDSource = 1, the VendorID is looked up in the Bluetooth assigned company IDs. If it's 2, it's looked up in the USB assigned company IDs

### 5.1.10 Device Identification Profile (DID)

Applicable to Service Class UUIDs:

- PnPInformation: `0x1200`

Last Modified: 2023-02-02

| Attribute ID | Attribute Name |
|---|---|
| 0x0200 | SpecificationID |
| 0x0201 | VendorID |
| 0x0202 | ProductID |
| 0x0203 | Version |
| 0x0204 | PrimaryRecord |
| 0x0205 | VendorIDSource |

First screenshot from BT Assigned Numbers document, 2nd from Wireshark
3rd from https://www.t-mobile.com/support/public-files/attachments/syncup-drive/SyncUP%20DRIVE%207000T1%20User%20Manual_%20EN.pdf

# BTC EIR Device ID

- Some devices seem to include Device ID information as shown here in BTC Extended Inquiry Response packets
  - Despite the fact that I can't find anything in the spec saying that's supposed to be a thing? (The "Supplement to the Bluetooth Core Specification" doesn't list it as valid...)
- If VendorIDSource = 1, the VendorID is looked up in the Bluetooth assigned company IDs. If it's 2, it's looked up in the USB assigned company IDs

```
Extended Inquiry Response Data
  Device Name: SD-7000T_96
    Length: 12
    Type: Device Name (0x09)
    Device Name: SD-7000T_96
  Tx Power Level
    Length: 2
    Type: Tx Power Level (0x0a)
    Power Level (dBm): 6
  Device ID / Security Manager TK Value
    Length: 9
    Type: Device ID / Security Manager TK Value (0x10)
    Vendor ID Source: USB Implementer's Forum (0x0002)
    Vendor ID: Linux Foundation (0x1d6b)
    Product ID: 0x0246 (Unknown)
    Version: 0x053c
```

Last Modified: 2023-02-02

| Attribute ID | Attribute Name |
|---|---|
| 0x0200 | SpecificationID |
| 0x0201 | VendorID |
| 0x0202 | ProductID |
| 0x0203 | Version |
| 0x0204 | PrimaryRecord |
| 0x0205 | VendorIDSource |

First screenshot from BT Assigned Numbers document, 2nd from Wireshark
3rd from https://www.t-mobile.com/support/public-files/attachments/syncup-drive/SyncUP%20DRIVE%207000T1%20User%20Manual_%20EN.pdf

# BTC EIR Device ID

SyncUP DRIVE connected by T-Mobile

Product: SD-7000T1

User Manual

```
Extended Inquiry Response Data
  Device Name: SD-7000T_96
    Length: 12
    Type: Device Name (0x09)
    Device Name: SD-7000T_96
  Tx Power Level
    Length: 2
    Type: Tx Power Level (0x0a)
    Power Level (dBm): 6
  Device ID / Security Manager TK Value
    Length: 9
    Type: Device ID / Security Manager TK Value (0x10)
    Vendor ID Source: USB Implementer's Forum (0x0002)
    Vendor ID: Linux Foundation (0x1d6b)
    Product ID: 0x0246 (Unknown)
    Version: 0x053c
```

- Some devices seem to include Device ID information as shown here in BTC Extended Inquiry Response packets
  - Despite the fact that I can't find anything in the spec saying that's supposed to be a thing? (The "Supplement to the Bluetooth Core Specification" doesn't list it as valid...)
- If VendorIDSource = 1, the VendorID is looked up in the Bluetooth assigned company IDs. If it's 2, it's looked up in the USB assigned company IDs

Last Modified: 2023-02-02

| Attribute ID | Attribute Name |
|---|---|
| 0x0200 | SpecificationID |
| 0x0201 | VendorID |
| 0x0202 | ProductID |
| 0x0203 | Version |
| 0x0204 | PrimaryRecord |
| 0x0205 | VendorIDSource |

First screenshot from BT Assigned Numbers document, 2nd from Wireshark
3rd from https://www.t-mobile.com/support/public-files/attachments/syncup-drive/SyncUP%20DRIVE%207000T1%20User%20Manual_%20EN.pdf

# Top 20 Vendors for BTC Device ID data

## BTC - 2023-10-26

| vendor_source | vendor_id_hex | company_name | frequency | |
|---|---|---|---|---|
| Bluetooth | 0x4E8 | STABILO International | 4462 | <- actually Samsung |
| USB | 0x1D6B | Linux Foundation | 433 | |
| Bluetooth | 0x0 | Ericsson AB | 95 | |
| USB | 0x44E | Alps Electric Co., Ltd | 85 | |
| Bluetooth | 0xA | Qualcomm Technologies International, Ltd. (QTIL) | 62 | |
| Bluetooth | 0x75 | Samsung Electronics Co. Ltd. | 60 | |
| USB | 0xA12 | Cambridge Silicon Radio, Ltd | 58 | |
| Bluetooth | 0x4C | Apple, Inc. | 23 | |
| Bluetooth | 0x3E0 | Actions (Zhuhai) Technology Co., Limited | 21 | |
| USB | 0x108C | Robert Bosch GmbH | 16 | |
| 992 | 0xFFFF | NULL | 15 | |
| Bluetooth | 0x9E | Bose Corporation | 12 | |
| Bluetooth | 0x474C | NULL | 12 | |
| USB | 0x54C | Sony Corp. | 5 | |
| Bluetooth | 0x850 | Yealink (Xiamen) Network Technology Co.,LTD | 5 | |
| 256 | 0x4E8 | NULL | 5 | |
| USB | 0xA9 | NULL | 4 | |
| Bluetooth | 0x418 | Reserved | 4 | |
| USB | 0xA | NULL | 4 | |
| Bluetooth | 0x103 | Bang & Olufsen A/S | 4 | |

# Top 20 Vendors for BTC Device ID data

## BTC - 2023-10-26

Samsung's setting their vendor source to Bluetooth but then using their USB ID

| vendor_source | vendor_id_hex | company_name | frequency | |
|---|---|---|---|---|
| Bluetooth | 0x4E8 | STABILO International | 4462 | <- actually Samsung |
| USB | 0x1D6B | Linux Foundation | 433 | |
| Bluetooth | 0x0 | Ericsson AB | 95 | |
| USB | 0x44E | Alps Electric Co., Ltd | 85 | |
| Bluetooth | 0xA | Qualcomm Technologies International, Ltd. (QTIL) | 62 | |
| Bluetooth | 0x75 | Samsung Electronics Co. Ltd. | 60 | |
| USB | 0xA12 | Cambridge Silicon Radio, Ltd | 58 | |
| Bluetooth | 0x4C | Apple, Inc. | 23 | |
| Bluetooth | 0x3E0 | Actions (Zhuhai) Technology Co., Limited | 21 | |
| USB | 0x108C | Robert Bosch GmbH | 16 | |
| 992 | 0xFFFF | NULL | 15 | |
| Bluetooth | 0x9E | Bose Corporation | 12 | |
| Bluetooth | 0x474C | NULL | 12 | |
| USB | 0x54C | Sony Corp. | 5 | |
| Bluetooth | 0x850 | Yealink (Xiamen) Network Technology Co.,LTD | 5 | |
| 256 | 0x4E8 | NULL | 5 | |
| USB | 0xA9 | NULL | 4 | |
| Bluetooth | 0x418 | Reserved | 4 | |
| USB | 0xA | NULL | 4 | |
| Bluetooth | 0x103 | Bang & Olufsen A/S | 4 | |

# 2thprint by Manufacturer-Specific Data

🧘‍♂️ or 🚶‍♂️

# 2thprint by Manufacturer-Specific Data (MSD):

## Company ID (CID)

- BLE Advertisements and BTC Extended Inquiry Response packets can include MSD data where the manufacturer can *mostly* put whatever they want



### 1 DATA TYPES DEFINITIONS AND FORMATS

This part defines the basic data types used for Extended Inquiry Response (EIR), Advertising Data (AD), Scan Response Data (SRD), Additional Controller Advertising Data (ACAD), and OOB data blocks. Additional data types may be defined in profile specifications.

Each data type shall only be used in accordance with the requirements specified in Table 1.1.

| | Context | | | | |
|---|---|---|---|---|---|
| **Data type** | **EIR** | **AD** | **SRD** | **ACAD** | **OOB** |
| Service UUID | O | O | O | O | O |
| Local Name | C.1 | C.1 | C.1 | X | C.1 |
| Flags | C.1 | C.1 | X | X | C.1 |
| Manufacturer Specific Data | O | O | O | O | O |

# 2thprint by Manufacturer-Specific Data (MSD):

## Company ID (CID)

- BLE Advertisements and BTC Extended Inquiry Response packets can include MSD data where the manufacturer can *mostly* put whatever they want



*Data Types Specification*

Bluetooth®

## 1 DATA TYPES DEFINITIONS AND FORMATS

This part defines the basic data types used for Extended Inquiry Response (EIR), Advertising Data (AD), Scan Response Data (SRD), Additional Controller Advertising Data (ACAD), and OOB data blocks. Additional data types may be defined in profile specifications.

Each data type shall only be used in accordance with the requirements specified in Table 1.1.

| | Context | | | | |
|---|---|---|---|---|---|
| **Data type** | **EIR** | **AD** | **SRD** | **ACAD** | **OOB** |
| Service UUID | O | O | O | O | O |
| Local Name | C.1 | C.1 | C.1 | X | C.1 |
| Flags | C.1 | C.1 | X | X | C.1 |
| Manufacturer Specific Data | O | O | O | O | O |

# 2thprint by Manufacturer-Specific Data (MSD):

## Company ID (CID)

- They's *supposed* to put their company's assigned number in the first 2 bytes, but not everyone does...
- Also, some put the company ID as little-endian, and some as big-endian, and some use both!



### 1.4 MANUFACTURER SPECIFIC DATA

#### 1.4.1 Description

The Manufacturer Specific data type is used for manufacturer specific data. The first two data octets shall contain a company identifier from Assigned Numbers. The interpretation of any other octets within the data shall be defined by the manufacturer specified by the company identifier.

#### 1.4.2 Format

| Data Type | Description |
|---|---|
| «Manufacturer Specific Data» *uint16*, which may be followed by *struct* | The first value contains the Company Identifier Code. Any remainder contains manufacturer specific data. |

*Table 1.5: Manufacturer Specific data type*

# 2thprint by Manufacturer-Specific Data (MSD):

## Company ID (CID)

- They's *supposed* to put their company's assigned number in the first 2 bytes, but not everyone does...
- Also, some put the company ID as little-endian, and some as big-endian, and some use both!



*Data Types Specification*

Bluetooth

### 1.4 MANUFACTURER SPECIFIC DATA

#### 1.4.1 Description

The Manufacturer Specific data type is used for manufacturer specific data. The first two data octets shall contain a company identifier from Ass Numbers. The interpretation of any other octets within the data shall ed by the manufacturer specified by the company identifier.

It doesn't say what endianness it should use!

#### 1.4.2 Format

| Data Type | Description |
|---|---|
| «Manufacturer Specific Data» *uint16*, which may be followed by *struct* | The first value contains the Company Identifier Code. Any remainder contains manufacturer specific data. |

*Table 1.5: Manufacturer Specific data type*

# Endianness info from BT Spec 4.0!

**Lost in BT Spec 4.2 when they created the Core Specification Supplement doc?**

## 8.1 EIR DATA TYPE DEFINITIONS

This section defines the basic EIR data types. Additional EIR data types may be defined in profile specifications.

All EIR data type values are listed in the Bluetooth Assigned Numbers document.

All numerical multi-byte entities and values associated with the following data types shall use little-endian byte order.

# Top 20 Vendors for BTC MSD data

## BTC - 2023-10-26

| device_BT_CID | company_name | frequency | |
|---|---|---|---|
| 0x8700 | Garmin International (wrong-endian) | 5942 | ❌ |
| 0x4C00 | Apple, Inc. (wrong-endian) | 2487 | 🍪 |
| 0x1D | Qualcomm | 2437 | 🍪 |
| 0xFF19 | Samsung Electronics Co. Ltd. (just wacky) | 1938 | 🍪 |
| 0xF | Broadcom Corporation | 950 | 🍪 |
| 0x75 | Samsung Electronics Co. Ltd. | 879 | 🍪 |
| 0x7500 | Samsung Electronics Co.,Ltd (wrong-endian) | 278 | 🍪 |
| 0xD906 | Shanghai Mountain View Silicon Co.,Ltd. (wrong-endian) | 274 | 🍪 |
| 0x3E0 | Actions (Zhuhai) Technology Co., Limited | 97 | 🍪 |
| 0x4C | Apple, Inc. | 45 | 🍪 |
| 0x27D | HUAWEI Technologies Co., Ltd. | 43 | ❌ |
| 0xA | Qualcomm Technologies International, Ltd. (QTIL) | 32 | 🍪 |
| 0xA02 | Ayxon-Dynamics GmbH | 14 | ❌ |
| 0x200 | Verifone Systems Pte Ltd. Taiwan Branch | 7 | ❌ |
| 0x0 | Ericsson AB | 7 | ❌ |
| 0xA00 | Ampler Bikes OU | 6 | ❌ |
| 0x5F0 | beken | 6 | 🍪 |
| 0x850 | Yealink (Xiamen) Network Technology Co.,LTD | 5 | ❌ |
| 0x18C | Wilo SE | 3 | ❌ |
| 0x67 | GN Audio A/S | 3 | ❌ |

...

# Top 20 Vendors for BLE MSD data

## BLE - 2023-10-26

| device_BT_CID | company_name | | frequency |
|---|---|---|---|
| 0x4C | Apple, Inc. | 🍪 | 9503999 |
| 0x6 | Microsoft | ❌ | 94967 |
| 0x75 | Samsung Electronics Co. Ltd. | 🍪 | 83182 |
| 0x11B | Hewlett Packard Enterprise | ❌ | 18899 |
| 0x183 | Walt Disney | ❌ | 14073 |
| 0x3 | IBM Corp. | ❌ | 11937 |
| 0x87 | Garmin International, Inc. | ❌ | 10150 |
| 0x131 | Cypress Semiconductor | 🍪 | 7872 |
| 0x0 | Ericsson AB | ❌ | 7547 |
| 0x171 | Amazon.com Services LLC | ❌ | 6926 |
| 0x57 | Harman International Industries, Inc. | ❌ | 6424 |
| 0x87F | Phillips Connect Technologies LLC | ❌ | 5735 |
| 0x12D | Sony Corporation | ❌ | 5265 |
| 0xE0 | Google | ❌ | 4370 |
| 0x310 | SGL Italia S.r.l. | ❌ | 3730 |
| 0xB01 | RESIDEO TECHNOLOGIES, INC. | ❌ | 3608 |
| 0xA01 | Cleveron AS | ❌ | 3139 |
| 0x157 | Anhui Huami Information Technology Co., Ltd. | ❌ | 2657 |
| 0x65 | HP, Inc. | ❌ | 2412 |
| 0x4C00 | Apple, Inc. (wrong-endian) | 🍪 | 1946 |

...

# 2thprint by UUID16

🧘‍♂️ or 🚶‍♂️

# 2thprint by UUID16

## UUID16Print

- 16-bit UUIDs can be optionally included in BTC EIR packets or BLE Advertisements
- They are frequently used to advertise company-specific services (usually served over GATT)
- Silicon vendors are present, but not prevalent

```
./TellMeEverything.py --UUID16stats=quiet
----= BLUETOOTH CLASSIC RESULTS =----
count   uuid16  company
❌ 254   0xfe4c  Volkswagen AG
❌ 57   0xfe31  Volkswagen AG
❌ 31   0xfe35  HUAWEI Technologies Co., Ltd
❌ 24   0xfdd1  Huawei Technologies Co., Ltd
🍪 12   0xfd69  Samsung Electronics Co., Ltd
❌ 1    0xfeea  Swirl Networks, Inc.

----= BLUETOOTH LOW ENERGY RESULTS =----
count   uuid16  company
❌ 109258   0xfef3  Google LLC
🍪 108953   0xfd6f  Apple, Inc.
🍪 44895   0xfd69  Samsung Electronics Co., Ltd
❌ 29877   0xfeed  Tile, Inc.
❌ 29059   0xfebe  Bose Corporation
❌ 13392   0xfeaa  Google LLC
❌ 13184   0xfe9f  Google LLC
❌ 12533   0xfe03  Amazon.com Services, Inc.
❌ 9450    0xfd5a  Samsung Electronics Co., Ltd.
❌ 8922    0xfe50  Google LLC
❌ 6642    0xfe1f  Garmin International, Inc.
❌ 5107    0xfe2c  Google LLC
❌ 4686    0xfe61  Logitech International SA
❌ 3903    0xfe4c  Volkswagen AG
❌ 3801    0xfe26  Google LLC
❌ 2663    0xfea0  Google LLC
❌ 2451    0xfd82  Sony Corporation
❌ 2420    0xfe78  Hewlett-Packard Company
❌ 2353    0xfee7  Tencent Holdings Limited.
❌ 2018    0xfe48  General Motors
...
```

# 2thprint by UUID16

## UUID16Print

- 16-bit UUIDs can be optionally included in BTC EIR packets or BLE Advertisements
- They are frequently used to advertise company-specific services (usually served over GATT)
- Silicon vendors are present, but not prevalent

```
./TellMeEverything.py --UUID16stats=quiet
----= BLUETOOTH CLASSIC RESULTS =----
count   uuid16   company
❌ 254   0xfe4c   Volkswagen AG
❌ 57   0xfe31   Volkswagen AG
❌ 31   0xfe35   HUAWEI Technologies Co., Ltd
❌ 24   0xfdd1   Huawei Technologies Co., Ltd
🍪 12   0xfd69   Samsung Electronics Co., Ltd
❌ 1    0xfeea   Swirl Networks, Inc.

----= BLUETOOTH LOW ENERGY RESULTS =----
count   uuid16   company
❌ 109258   0xfef3   Google LLC
🍪 108953   0xfd6f   Apple, Inc.
🍪 44895   0xfd69   Samsung Electronics Co., Ltd
❌ 29877   0xfeed   Tile, Inc.
❌ 29059   0xfebe   Bose Corporation
❌ 13392   0xfeaa   Google LLC
❌ 13184   0xfe9f   Google LLC
❌ 12533   0xfe03   Amazon.com Services, Inc.
❌ 9450    0xfd5a   Samsung Electronics Co., Ltd.
❌ 8922    0xfe50   Google LLC
❌ 6642    0xfe1f   Garmin International, Inc.
❌ 5107    0xfe2c   Google LLC
❌ 4686    0xfe61   Logitech International SA
           0xfe4c   Volkswagen AG
           0xfe26   Google LLC
           0xfea0   Google LLC
           0xfd82   Sony Corporation
           0xfe78   Hewlett-Packard Company
           0xfee7   Tencent Holdings Limited.
           0xfe48   General Motors
...
```

```
1740   0xfee0 🍪 Anhui Huami Information Technology Co., Ltd.
1558   0xfd92 🍪 Qualcomm Technologies International, Ltd. (QTIL)
749    0xfef5 🍪 Dialog Semiconductor GmbH
216    0xfe8f 🍪 CSR
189    0xfefb 🍪 Telit Wireless Solutions (Formerly Stollmann E+V GmbH)
136    0xfef1 🍪 CSR
74     0xfe59 🍪 Nordic Semiconductor ASA
```

# What I Want

Individual Device → Device Type → Model

Firmware Version ← BT Chip ← Manufacturer

# What I Want

Individual Device → Device Type → Model

Firmware Version ← BT Chip ← Manufacturer

UUID16: **0xfef1**

# What I Want

Individual Device
Device Type
Model
Firmware Version
BT Chip
Manufacturer
CSR
UUID16Print
DATABASE LOOKUP
UUID16:
0xfef1

# What I Want

Individual Device
Device Type
Model
Firmware Version
BT Chip
Manufacturer
CSR
CSR
UUID16Print
DATABASE LOOKUP
UUID16:
0xfef1

# What I Want

*ASSUMPTION*:

UUID16Prints are just the assigned values from the BT public/assigned_numbers/uuids/member_uuids.yaml

# 2thprint by UUID128

🧘‍♂️ or 🚶‍♂️

# 2thprint by UUID128

## UUID128Print

- Detective Work🕵️: UUID128Print -> NamePrint

# Vendor-specific 128-bit UUIDs

- abbaff00-e56a-484c-b832-8b17cf6cbfe8
  - Versa (|2|Lite), Ionic
- adab*fb00*-6e7d-4601-bda2-bffaa68956ba
  - Inspire HR, Flex 2
- adab*0d57*-6e7d-4601-bda2-bffaa68956ba
- adab*6552*-6e7d-4601-bda2-bffaa68956ba
  - One
- adab*5b8c*-6e7d-4601-bda2-bffaa68956ba
  - Flex

# Vendor-specific 128-bit UUIDs

- abbaff00-e56a-484c-b832-8b17cf6cbfe8
  - Versa (|2|Lite), Ionic
- adab*fb00*-6e7d-4601-bda2-bffaa68956ba
  - Inspire HR, Flex 2
- adab*0d57*-6e7d-4601-bda2-bffaa68956ba
- adab*6552*-6e7d-4601-bda2-bffaa68956ba
  - One
- adab*5b8c*-6e7d-4601-bda2-bffaa68956ba
  - Flex

```
> HCI Event: LE Meta Event (0x3e) plen 42
    LE Advertising Report (0x02)
      Num reports: 1
      Event type: Connectable undirected - ADV_IND (0x00)
      Address type: Random (0x01)
      Address: F5:6E:B2:C3:73:D2 (Static)
      Data length: 30
      Flags: 0x06
        LE General Discoverable Mode
        BR/EDR Not Supported
      128-bit Service UUIDs (partial): 1 entry
        Vendor specific (abbaff00-e56a-484c-b832-8b17cf6cbfe8)
      Service Data (UUID 0x180a): 2604329303
      RSSI: -93 dBm (0xa3)
```

# Vendor-specific 128-bit UUIDs

- abbaff00-e56a-484c-b832-8b17cf6cbfe8
  - Versa (|2|Lite), Ionic
- adab*fb00*-6e7d-4601-bda2-bffaa68956ba
  - Inspire HR, Flex 2
- adab*0d57*-6e7d-4601-bda2-bffaa68956ba
- adab*6552*-6e7d-4601-bda2-bffaa68956ba
  - One
- adab*5b8c*-6e7d-4601-bda2-bffaa68956ba
  - Flex

```
> HCI Event: LE Meta Event (0x3e) plen 42
    LE Advertising Report (0x02)
      Num reports: 1
      Event type: Connectable undirected - ADV_IND (0x00)
      Address type: Random (0x01)
      Address: F5:6E:B2:C3:73:D2 (Static)
      Data length: 30
      Flags: 0x06
        LE General Discoverable Mode
        BR/EDR Not Supported
      128-bit Service UUIDs (partial): 1 entry
        Vendor specific (abbaff00-e56a-484c-b832-8b17cf6cbfe8)
      Service Data (UUID 0x180a): 2604329303
      RSSI: -93 dBm (0xa3)
```

# What I Want

Individual Device → Device Type → Model

Firmware Version ← BT Chip ← Manufacturer

# What I Want

Individual Device → Device Type → Model

Firmware Version ← BT Chip ← Manufacturer

UUID128: **adab**fb00**-6e7d-4601-bda2-bffaa68956ba**

# What I Want

Individual Device → Device Type → Model

Firmware Version ← BT Chip ← Manufacturer

*Fitbit*

UUID128Print
DATABASE LOOKUP

UUID128: **adab**fb00**-6e7d-4601-bda2-bffaa68956ba**

# What I Want

NamePrint
DATABASE LOOKUP
Flex 2
Inspire HR
Activity
Tracker
Individual Device
Device Type
Model
Firmware Version
BT Chip
Manufacturer
Fitbit
UUID128Print
DATABASE LOOKUP
UUID128: adabfb00-6e7d-4601-bda2-bffaa68956ba

# What I Want

*ASSUMPTION*:
UUID128Prints can be reused *within* Manufacturers,
but not reused *between* Manufacturers

# Vendor-specific 128-bit UUIDs

## 🍪Silicon-specific examples: Texas Instruments

- f000ffc0-0451-4000-b000-000000000000 - Texas Instruments OTA Firmware update GATT Service
  - f000ffc1-0451-4000-b000-000000000000, f000ffc2-0451-4000-b000-000000000000, f000ffc3-0451-4000-b000-000000000000, f000ffc4-0451-4000-b000-000000000000 - Associated GATT Characteristics
- BleedingBit exploited an Aruba-customized TI OTA firmware update service (that just added some security-by-obscurity magic unlock code)

https://software-dl.ti.com/lprf/simplelink_cc2640r2_sdk/1.35.00.33/exports/docs/ble5stack/ble_user_guide/html/oad/oad_concepts.html#oad-service-0xffc0

# Vendor-specific 128-bit UUIDs

## 🍪Silicon-specific examples: Nordic

- 6e400001-b5a3-f393-e0a9-e50e24dcca9e - UART GATT Service (advertised in SCAN_RSP as well as GATT)
  - 6e400002-b5a3-f393-e0a9-e50e24dcca9e - UART RX GATT Characteristic
  - 6e400003-b5a3-f393-e0a9-e50e24dcca9e - UART TX GATT Characteristic
- 00001530-1212-efde-1523-785feabcd123 - "Legacy" (Insecure) Device Firmware Update (DFU) GATT Service
  - 00001531-1212-efde-1523-785feabcd123, 00001532-1212-efde-1523-785feabcd123, 000015**3**-1212-efde-1523-785feabcd123 - Associated GATT Characteristics
- 0000fe59-0000-1000-8000-00805f9b34fb - Secure Device Firmware Update (DFU) GATT Service
  - 8EC90001-F315-4F60-9FB8-838830DAEA50, 8EC90002-F315-4F60-9FB8-838830DAEA50 - Associated GATT Characteristics

https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/libraries/bluetooth_services/services/nus.html
https://nordicsemiconductor.github.io/IOSnRFConnect/assets/files/UserManual.pdf
https://github.com/adafruit/Bluefruit_LE_Connect_Android/blob/master/app/src/main/java/com/adafruit/bluefruit/le/connect/ble/KnownUUIDs.java

# Vendor-specific 128-bit UUIDs

🍪**Silicon-specific examples: Cambridge Silicon Radio (bought by Qualcomm)**

- 0000110**0**-d102-11e1-9b23-00025b00a5a5 - CSR GAIA (Over The Air update protocol) GATT service
  - 0000110**1**-d102-11e1-9b23-00025b00a5a5, 0000110**2**-d102-11e1-9b23-00025b00a5a5, 0000110**3**-d102-11e1-9b23-00025b00a5a5 - Associated GATT characteristics

https://developer.qualcomm.com/qfile/34081/csr102x_otau_overview.pdf

# Vendor-specific 128-bit UUIDs

## 🍪Silicon-specific examples: Silicon Labs

- 331a36f5-2459-45ea-9d95-6142f0c4b307 - BGX Xpress Streaming (arbitrary data) GATT Service
  - a9da6040-0823-4995-94ec-9ce41ca28833, a73e9a10-628f-4494-a099-12efaf72258f, 75a9f022-af03-4e41-b4bc-9de90a47d50b - Associated GATT characteristics

https://docs.silabs.com/gecko-os/1/bgx/latest/ble-services

# Vendor-specific 128-bit UUIDs

## 🧩Module-specific examples: Laird

- 569a1101-b87f-490c-92cb-11ba5ea5167c - Virtual Serial Port Service (GATT & ADV_IND)
  - 569a2000-b87f-490c-92cb-11ba5ea5167c - TX GATT Characteristic
  - 569a2001-b87f-490c-92cb-11ba5ea5167c - RX GATT Characteristic

https://www.mouser.com/pdfDocs/ApplicationNoteLairdBLESerialPortServicev1_0_0.pdf

# What I Want

Individual Device → Device Type → Model

Firmware Version ← BT Chip ← Manufacturer

# What I Want

Individual Device → Device Type → Model

Firmware Version ← BT Chip ← Manufacturer

UUID128: 6e400001-b5a3-f393-e0a9-e50e24dcca9e

# What I Want

Individual Device → Device Type → Model

Firmware Version ← BT Chip ← Manufacturer

*Nordic*

UUID128Print
DATABASE LOOKUP

UUID128: 6e400001-b5a3-f393-e0a9-e50e24dcca9e

# What I Want

Individual Device → Device Type → Model

Firmware Version ← BT Chip ← Manufacturer

*Nordic*

UUID128Print
DATABASE LOOKUP

`UUID128: 6e400001-b5a3-f393-e0a9-e50e24dcca9e`

*ASSUMPTION*:
UUID128Prints can be reused *within* Manufacturers,
but not reused *between* Manufacturers

# What I Want

*ASSUMPTION*:
UUID128Prints can be reused *within* Manufacturers,
but not reused *between* Manufacturers

# iBeacon: "The other UUID128!"

## UUID128Print

- One of the most common types of "Manufacturer-Specific Data" (MSD) is the Apple-specified iBeacon (https://developer.apple.com/ibeacon/)
- It contains a UUID128 that beacon-deployers are supposed to associate with themselves, a Major ID and Minor ID that they're supposed to associate with individual beacons (e.g. Major ID for a country/store, and minor ID for an individual beacon)

```
Device Company ID: 0x004c (Apple, Inc.) - take with a grain of salt, not all companies populate this accurately!
         Endianness-flipped device company ID (in case the vendor used the wrong endianness): 0x4c00 (No Match)
Raw Data: 0215f34ebac47cd74027878e55f4e71d03090000000088
Apple iBeacon:
        UUID128: f34ebac4-7cd7-4027-878e-55f4e71d0309
        Major ID: 0000
        Minor ID: 0000
        RSSI at 1 meter: -120dBm
        In BT LE Data (LE_bdaddr_to_mf_specific), bdaddr_random = 0 (Public)
        This was found in an event of type 3 which corresponds to Non-Connectable Undirected Advertising (ADV_NONCONN_IND)
```

# iBeacon: "The other UUID128!"

## UUID128Print

- Detective Work🕵️: GATT -> Google Search

```
        GATT Characteristic: 00002a29-0000-1000-8000-00805f9b34fb (Manufacturer Name String), Properties: 2 ('Readable' )
        GATT Characteristic value read as b'MILWAUKEE TOOL'
GATT Descriptor: 00002803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 9
GATT Descriptor: 00002a24-0000-1000-8000-00805f9b34fb, Descriptor Handle: 10
        GATT Characteristic: 00002a24-0000-1000-8000-00805f9b34fb (Model Number String), Properties: 2 ('Readable' )
        GATT Characteristic value read as b'BLE112'
GATT Descriptor: 00002803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 11
GATT Descriptor: 00002a25-0000-1000-8000-00805f9b34fb, Descriptor Handle: 12
        GATT Characteristic: 00002a25-0000-1000-8000-00805f9b34fb (Serial Number String), Properties: 2 ('Readable' )
        GATT Characteristic value read as b'123456789'
```

# iBeacon: "The other UUID128!"

## UUID128Print

- Detective Work🕵️: GATT -> Google Search

```
istic: 00002a29-0000-1000-8000-00805f9b34fb (Manufacturer
istic value read as b'MILWAUKEE TOOL'
803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 9
a24-0000-1000-8000-00805f9b34fb, Descriptor Handle: 10
istic: 00002a24-0000-1000-8000-00805f9b34fb (Model Number
istic value read as b'BLE112'
803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 11
a25-0000-1000-8000-00805f9b34fb, Descriptor Handle: 12
istic: 00002a25-0000-1000-8000-00805f9b34fb (Serial Number
istic value read as b'123456789'
```

# iBeacon: "The other UUID128!"

## UUID128Print

- Detective Work🕵️: GATT -> Google Search

```
istic: 00002a29-0000-1000-8000-00805f9b34fb (Manufacturer
istic value read as b MILWAUKEE TOOL
803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 9
a24-0000-1000-8000-00805f9b34fb, Descriptor Handle: 10
istic: 00002a24-0000-1000-8000-00805f9b34fb (Model Number
istic value read as b'BLE112'
803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 11
a25-0000-1000-8000-00805f9b34fb, Descriptor Handle: 12
istic: 00002a25-0000-1000-8000-00805f9b34fb (Serial Number
istic value read as b'123456789'
```

# iBeacon: "The other UUID128

## UUID128Print

- Detective Work🕵️: GATT -> Google Search

```
istic: 00002a29-0000-1000-8000-00805f9b34fb (Manufacturer
istic value read as b MILWAUKEE TOOL
803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 9
a24-0000-1000-8000-00805f9b34fb, Descriptor Handle: 10
istic: 00002a24-0000-1000-8000-00805f9b34fb (Model Number
istic value read as b'BLE112'
803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 11
a25-0000-1000-8000-00805f9b34fb, Descriptor Handle: 12
istic: 00002a25-0000-1000-8000-00805f9b34fb (Serial Number
istic value read as b'123456789'
```

# iBeacon: "The other UUID128!"

## UUID128Print

- Detective Work🕵️: GATT -> Google Search

```
istic: 00002a29-0000-1000-8000-00805f9b34fb (Manufacturer
istic value read as b MILWAUKEE TOOL
803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 9
a24-0000-1000-8000-00805f9b34fb, Descriptor Handle: 10
istic: 00002a24-0000-1000-8000-00805f9b34fb (Model Number
istic value read as b'BLE112'
803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 11
a25-0000-1000-8000-00805f9b34fb, Descriptor Handle: 12
istic: 00002a25-0000-1000-8000-00805f9b34fb (Serial Number
istic value read as b'123456789'
```

# iBeacon: "The other UUID128!"

## UUID128Print

- Detective Work🕵️: GATT -> Google Search

```
istic: 00002a29-0000-1000-8000-00805f9b34fb (Manufacturer
istic value read as b MILWAUKEE TOOL
803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 9
a24-0000-1000-8000-00805f9b34fb, Descriptor Handle: 10
istic: 00002a24-0000-1000-8000-00805f9b34fb (Model Number
istic value read as b BLE112
803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 11
a25-0000-1000-8000-00805f9b34fb, Descriptor Handle: 12
istic: 00002a25-0000-1000-8000-00805f9b34fb (Serial Number
istic value read as b'123456789'
```

Milwaukee tool BLE112 bluetooth

Shopping | Images | Videos | App | News | Maps | Books | Flights | Finance

Milwaukee Tool
https://www.milwaukeetool.com › Products

## TICK Tool and Equipment Tracker

**Bluetooth** tracker for **tools** and **equipment** offers multiple attachment options and a low profile design - users can glue, screw, rivet or strap the TICK™…

★★★★½ Rating: 4.7 · 901 reviews

Missing: ~~BLE112~~ | Show results with: BLE112

Milwaukee Tool
https://www.milwaukeetool.com › Products

Milwaukee tool BLE112 bluetooth

Shopping | Images | Videos | App | News | Maps | Books | Flights | Finance

**They make an AirTag "for tools"!**

***Where's your anti-stalking defense now?***

and a low profile design - users can glue, screw, rivet or strap the TICK™

★★★★½ Rating: 4.7 · 901 reviews

Missing: ~~BLE112~~ | Show results with: BLE112

Milwaukee Tool

https://www.milwaukeetool.com › Products

Milwaukee tool BLE112 bluetooth

Shopping | Images | Videos | App | News | Maps | Books | Flights | Finance

Milwaukee Tool
https://www.milwaukeetool.com › Products

## TICK Tool and Equipment Tracker

**Bluetooth** tracker for **tools** and **equipment** offers multiple attachment options and a low profile design - users can glue, screw, rivet or strap the TICK™…

★★★★½ Rating: 4.7 · 901 reviews

Missing: ~~BLE112~~ | Show results with: BLE112

Milwaukee Tool
https://www.milwaukeetool.com › Products

Can Milwaukee tools be traced?

Does Milwaukee have Bluetooth?

Feedback

Silicon Labs
https://www.silabs.com › bluetooth › device.bled112

## BLED112

The **BLED112 Bluetooth** Low Energy Dongle integrates all **Bluetooth** LE features. The USB dongle has a virtual COM port that enables seamless ho…

BLE112 bluetooth -"BLED112"

Images Videos Manual Shopping News Maps Books Flights Finan

**Sponsored**

Mouser Electronics
https://www.mouser.com

## BLED112 Bluetooth Smart Dongle - Silicon Labs | Mouser

Mouser is an Authorized Silicon Labs Distributor with Inventory, Prices & Datasheets. Huge Selection of Silicon Labs In Stock with No Minimum Orders & Fast Delivery! Fast Delivery. AS6496 Certified. Authorized Distributor. ISO 9001:2015. Order by 8PM CST.

Get phone number

Selection of Silicon Labs In Stock with No Minimum Orders & Fast Delivery! Fast Delivery. AS6496 Certified. Authorized Distributor. ISO 9001:2015. Order by 8PM CST.

Get phone number

Silicon Labs
https://www.silabs.com › public › data-sheets

## BLE112 Datasheet

**BLE112** offers all **Bluetooth** Low Energy features: radio, stack, profiles and application space for customer applications, so no external processor is needed. The ...

36 pages

Digikey
https://forum.digikey.com › getting-started-with-the-bl...

## Getting Started with the BlueGiga BLE112 Bluetooth Low ...

Mar 9, 2021 — Purpose. This page explains how to set up the Silicon Labs **BLE112** **Bluetooth** Low Energy module to communicate with a microcontroller via UART.

https://www.silabs.com/documents/public/data-sheets/BLE112-DataSheet.pdf

Not Recommended for New Designs

# BLE112

DATA SHEET

Wednesday, 02 December 2020

Version 1.8

SILICON LABS

Selection of Silicon Labs In Stock with No Minimum Orders & Fast Delivery! Fast Delivery. AS6496 Certified. Authorized Distributor. ISO 9001:2015. Order by 8PM CST.

Get phone number

Silicon Labs
https://www.silabs.com › public › data-sheets

## BLE112 Datasheet

**BLE112** offers all **Bluetooth** Low Energy features: radio, stack, profiles and application space for customer applications, so no external processor is needed. The ...

36 pages

Digikey
https://forum.digikey.com › getting-started-with-the-bl...

## Getting Started with the BlueGiga BLE112 Bluetooth Low ...

Mar 9, 2021 — Purpose. This page explains how to set up the Silicon Labs **BLE112** **Bluetooth** Low Energy module to communicate with a microcontroller via UART.

Selection of Silicon Labs In Stock with No Minimum Orders & Fast Delivery! Fast Delivery. AS6496 Certified. Authorized Distributor. ISO 9001:2015. Order by 8PM CST.

Get phone number

Silicon Labs
https://www.silabs.com › public › data-sheets

## BLE112 Datasheet

**BLE112** offers all **Bluetooth** Low Energy features: radio, stack, profiles and application space for customer applications, so no external processor is needed. The ...

36 pages

Digikey
https://forum.digikey.com › getting-started-with-the-bl...

## Getting Started with the BlueGiga BLE112 Bluetooth Low ...

Mar 9, 2021 — Purpose. This page explains how to set up the Silicon Labs **BLE112 Bluetooth** Low Energy module to communicate with a microcontroller via UART

# iBeacon: "The other UUID128!"

## UUID128Print

- Detective Work🕵️: GATT -> Google Search -> **BLE112 == Bluegiga module**

```
n:
rvice: Begin Handle: 1   End Handle: 5           UUID128: 00001800-0000-1000-8000-00805f9b34fb (Generic Access)
 GATT Descriptor: 00002800-0000-1000-8000-00805f9b34fb, Descriptor Handle: 1
 GATT Descriptor: 00002803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 2
 GATT Descriptor: 00002a00-0000-1000-8000-00805f9b34fb, Descriptor Handle: 3
         GATT Characteristic: 00002a00-0000-1000-8000-00805f9b34fb (Device Name), Properties: 10 ('Readable' 'Writable' )
         GATT Characteristic value read as b'000C007771MKE  '
 GATT Descriptor: 00002803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 4
 GATT Descriptor: 00002a01-0000-1000-8000-00805f9b34fb, Descriptor Handle: 5
         GATT Characteristic: 00002a01-0000-1000-8000-00805f9b34fb (Appearance), Properties: 2 ('Readable' )
         GATT Characteristic value read as b'\x00\x01'
                 Appearance decodes as: Category (8): Tag, Sub-Category (0): Generic
rvice: Begin Handle: 6   End Handle: 12          UUID128: 0000180a-0000-1000-8000-00805f9b34fb (Device Information)
 GATT Descriptor: 00002800-0000-1000-8000-00805f9b34fb, Descriptor Handle: 6
 GATT Descriptor: 00002803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 7
 GATT Descriptor: 00002a29-0000-1000-8000-00805f9b34fb, Descriptor Handle: 8
         GATT Characteristic: 00002a29-0000-1000-8000-00805f9b34fb (Manufacturer Name String), Properties: 2 ('Readable' )
         GATT Characteristic value read as b'MILWAUKEE TOOL'
 GATT Descriptor: 00002803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 9
 GATT Descriptor: 00002a24-0000-1000-8000-00805f9b34fb, Descriptor Handle: 10
         GATT Characteristic: 00002a24-0000-1000-8000-00805f9b34fb (Model Number String), Properties: 2 ('Readable' )
         GATT Characteristic value read as b'BLE112'
 GATT Descriptor: 00002803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 11
 GATT Descriptor: 00002a25-0000-1000-8000-00805f9b34fb, Descriptor Handle: 12
         GATT Characteristic: 00002a25-0000-1000-8000-00805f9b34fb (Serial Number String), Properties: 2 ('Readable' )
         GATT Characteristic value read as b'123456789'
```

```
p:c4:7a:
 IEEE OUI (00:07:80): Bluegiga Technologies OY

 Inquiry Result Device info.



d.

er found.

ata found.

ecific Data:
Company ID: 0x6501 (No Match) - take with a grain of salt, not all companies pop
  Endianness-flipped device company ID (in case the vendor used the wrong endian
a: 000c00777100
 In BT LE Data (LE_bdaddr_to_mf_specific), bdaddr_random = 0 (Public)
 This was found in an event of type 0 which corresponds to Connectable Undirecte

ice Data found.

n:
rvice: Begin Handle: 1  End Handle: 5          UUID128: 00001800-0000-1000-800
```

```
IEEE OUI (00:07:80): Bluegiga Technologies OY
Inquiry Result Device info.


d.

er found.

ata found.

ecific Data:
Company ID: 0x6501 (No Match) - take with a grain of salt, not all companies pop
 Endianness-flipped device company ID (in case the vendor used the wrong endian
a: 000c00777100
 In BT LE Data (LE_bdaddr_to_mf_specific), bdaddr_random = 0 (Public)
 This was found in an event of type 0 which corresponds to Connectable Undirecte

ice Data found.

n:
```

bluegiga silicon labs

Images | Bluetooth | App | Videos | BLED112 | BLE GUI tool | UART demo | Shopp

Silicon Labs
https://www.silabs.com › developers › bluegiga-blueto...

## Bluegiga Bluetooth Smart Software Development Kit (SDK)

**Bluegiga** Bluetooth Smart Software is a complete Bluetooth Smart software stack for **Bluegiga** Legacy Bluetooth Smart products, such as BLE112, BLE113, ...

## People also ask

Is Silicon Labs safe?

## Bluegiga Legacy Modules (BLE)

The **Bluegiga** Legacy Bluetooth 4.0 Low Energy (BLE) Modules were launched in 2011. For new Bluetooth 5 designs, we recommend our latest Bluetooth modules.

Silicon Labs Newsroom
https://news.silabs.com › 2015-02-03-Silicon-Labs-Ac...

## Silicon Labs Acquires Bluegiga, a Leader in Bluetooth and ...

Feb 3, 2015 — **Silicon Labs** completed the acquisition of **Bluegiga** Technologies Oy on January 30, 2015. Under the agreement, **Bluegiga** investors received ...

## Bluegiga Legacy Modules (BLE)

The **Bluegiga** Legacy Bluetooth 4.0 Low Energy (BLE) Modules were launched in 2011. For new Bluetooth 5 designs, we recommend our latest Bluetooth modules.

Silicon Labs Newsroom

https://news.silabs.com › 2015-02-03-Silicon-Labs-Ac

## Silicon Labs Acquires Bluegiga, a Leader in Bluetooth and ...

Feb 3, 2015 — **Silicon Labs** completed the acquisition of **Bluegiga** Technologies Oy on January 30, 2015. Under the agreement, **Bluegiga** investors received ...

# iBeacon: "The other UUID128!"

## UUID128Print

- Detective Work🕵️: GATT -> Google Search -> BLE112 == Bluegiga module -> Google Search **-> Bluegiga == Silicon Labs**

# iBeacon: "The other UUID128!"

## UUID128Print

- Detective Work🕵️: GATT -> Google Search -> BLE112 == Bluegiga module -> Google Search -> Bluegiga == Silicon Labs

```
istic: 00002a29-0000-1000-8000-00805f9b34fb (Manufacturer
istic value read as b'MILWAUKEE TOOL'
803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 9
a24-0000-1000-8000-00805f9b34fb, Descriptor Handle: 10
istic: 00002a24-0000-1000-8000-00805f9b34fb (Model Number
istic value read as b BLE112
803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 11
a25-0000-1000-8000-00805f9b34fb, Descriptor Handle: 12
istic: 00002a25-0000-1000-8000-00805f9b34fb Serial Number
istic value read as b 123456789
```

# iBeacon: "The other UUID128!"

## UUID128Print

- Detective Work🕵️: GATT -> Google Search -> BLE112 == Bluegiga module -> Google Search -> Bluegiga == Silicon Labs

```
istic: 00002a29-0000-1000-8000-00805f9b34fb (Manufacturer
istic value read as b'MILWAUKEE TOOL'
803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 9
a24-0000-1000-8000-00805f9b34fb, Descriptor Handle: 10
istic: 00002a24-0000-1000-8000-00805f9b34fb (Model Number
istic value read as b BLE112
000-1000-8000-00805f9b34fb, Descriptor Handle: 11
805f9b34fb, Descriptor Handle: 12
1000-8000-00805f9b34fb Serial Number
123456789
```

Kinda seems *uninitialized*

```
n:
rvice: Begin Handle: 1   End Handle: 5           UUID128: 00001800-0000-1000-8000-00805f9b34fb (Generic Access)
 GATT Descriptor: 00002800-0000-1000-8000-00805f9b34fb, Descriptor Handle: 1
 GATT Descriptor: 00002803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 2
 GATT Descriptor: 00002a00-0000-1000-8000-00805f9b34fb, Descriptor Handle: 3
         GATT Characteristic: 00002a00-0000-1000-8000-00805f9b34fb (Device Name), Properties: 10 ('Readable' 'Writable' )
         GATT Characteristic value read as b'000C007771MKE  '
 GATT Descriptor: 00002803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 4
 GATT Descriptor: 00002a01-0000-1000-8000-00805f9b34fb, Descriptor Handle: 5
         GATT Characteristic: 00002a01-0000-1000-8000-00805f9b34fb (Appearance), Properties: 2 ('Readable' )
         GATT Characteristic value read as b'\x00\x01'
                 Appearance decodes as: Category (8): Tag, Sub-Category (0): Generic
rvice: Begin Handle: 6   End Handle: 12          UUID128: 0000180a-0000-1000-8000-00805f9b34fb (Device Information)
 GATT Descriptor: 00002800-0000-1000-8000-00805f9b34fb, Descriptor Handle: 6
 GATT Descriptor: 00002803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 7
 GATT Descriptor: 00002a29-0000-1000-8000-00805f9b34fb, Descriptor Handle: 8
         GATT Characteristic: 00002a29-0000-1000-8000-00805f9b34fb (Manufacturer Name String), Properties: 2 ('Readable' )
         GATT Characteristic value read as b'MILWAUKEE TOOL'
 GATT Descriptor: 00002803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 9
 GATT Descriptor: 00002a24-0000-1000-8000-00805f9b34fb, Descriptor Handle: 10
         GATT Characteristic: 00002a24-0000-1000-8000-00805f9b34fb (Model Number String), Properties: 2 ('Readable' )
         GATT Characteristic value read as b'BLE112'
 GATT Descriptor: 00002803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 11
 GATT Descriptor: 00002a25-0000-1000-8000-00805f9b34fb, Descriptor Handle: 12
         GATT Characteristic: 00002a25-0000-1000-8000-00805f9b34fb (Serial Number String), Properties: 2 ('Readable' )
         GATT Characteristic value read as b'123456789'
```

# iBeacon: "The other UUID128!"

## UUID128Print

```
turer-specific Data:
 Device Company ID: 0x6501 (No Match) - take with a grain of salt, not all companies populate
          Endianness-flipped device company ID (in case the vendor used the wrong endianness): 0x0165 (Milwaukee Electric
 Raw Data: 000c00777100
         In BT LE Data (LE_bdaddr_to_mf_specific), bdaddr_random = 0 (Public)
         This was found in an event of type 0 which corresponds to Connectable Undirected Advertising (ADV_IND)

s of Device Data found.

formation:
 GATT Service: Begin Handle: 1   End Handle: 5           UUID128: 00001800-0000-1000-8000-00805f9b34fb (Generic Access)
         GATT Descriptor: 00002800-0000-1000-8000-00805f9b34fb, Descriptor Handle: 1
         GATT Descriptor: 00002803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 2
         GATT Descriptor: 00002a00-0000-1000-8000-00805f9b34fb, Descriptor Handle: 3
                 GATT Characteristic: 00002a00-0000-1000-8000-00805f9b34fb (Device Name), Properties: 10 ('Readable' 'Writa
                 GATT Characteristic value read as b'000C007771MKE  '
         GATT Descriptor: 00002803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 4
         GATT Descriptor: 00002a01-0000-1000-8000-00805f9b34fb, Descriptor Handle: 5
                 GATT Characteristic: 00002a01-0000-1000-8000-00805f9b34fb (Appearance), Properties: 2 ('Readable' )
                 GATT Characteristic value read as b'\x00\x01'
                         Appearance decodes as: Category (8): Tag, Sub-Category (0): Generic
 GATT Service: Begin Handle: 6   End Handle: 12          UUID128: 0000180a-0000-1000-8000-00805f9b34fb (Device Information)
         GATT Descriptor: 00002800-0000-1000-8000-00805f9b34fb, Descriptor Handle: 6
         GATT Descriptor: 00002803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 7
         GATT Descriptor: 00002a29-0000-1000-8000-00805f9b34fb, Descriptor Handle: 8
                 GATT Characteristic: 00002a29-0000-1000-8000-00805f9b34fb (Manufacturer Name String), Properties: 2 ('Reac
                 GATT Characteristic value read as b'MILWAUKEE TOOL'
         GATT Descriptor: 00002803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 9
```

# iBeacon: "The other UUID128!"

## UUID128Print

```
turer-specific Data:
 Device Company ID: 0x6501 (No Match) - take with a grain of salt, not all companies populate
          Endianness-flipped device company ID (in case the vendor used the wrong endianness): 0x0165 (Milwaukee Electric
 Raw Data: 000c00777100
         In BT LE Data (LE_bdaddr_to_mf_specific), h        ndom = 0 (Public)
         This was found in an event of type 0 which        ds to Connectable Undirected Advertising (ADV_IND)

s of Device Data found.

formation:
 GATT Service: Begin Handle: 1   End Handle: 5          UID128:
         GATT Descriptor: 00002800-0000-1000-800        34fb, Desc
         GATT Descriptor: 00002803-0000-1000-80         fb, Desc
         GATT Descriptor: 00002a00-0000-1000-8          , Desc
                 GATT Characteristic: 00002a00          00805f
                 GATT Characteristic value read as b'000C00777 1MKE   '
         GATT Descriptor: 00002803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 4
         GATT Descriptor: 00002a01-0000-1000-8000-00805f9b34fb, Descriptor Handle: 5
                 GATT Characteristic: 00002a01-0000-1000-8000-00805f9b34fb (Appearance), Properties: 2 ('Readable' )
                 GATT Characteristic value read as b'\x00\x01'
                         Appearance decodes as: Category (8): Tag, Sub-Category (0): Generic
 GATT Service: Begin Handle: 6   End Handle: 12          UUID128: 0000180a-0000-1000-8000-00805f9b34fb (Device Information)
         GATT Descriptor: 00002800-0000-1000-8000-00805f9b34fb, Descriptor Handle: 6
         GATT Descriptor: 00002803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 7
         GATT Descriptor: 00002a29-0000-1000-8000-00805f9b34fb, Descriptor Handle: 8
                 GATT Characteristic: 00002a29-0000-1000-8000-00805f9b34fb (Manufacturer Name String), Properties: 2 ('Reac
                 GATT Characteristic value read as b'MILWAUKEE TOOL'
         GATT Descriptor: 00002803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 9
```

They seem to be using a big-endian BT CID, rather than the more common little-endian

# iBeacon: "The other UUID128!"

## UUID128Print

- Detective Work🕵️: GATT -> Google Search -> BLE112 == Bluegiga module -> Google Search -> Bluegiga == Silicon Labs -> iBeacons

```
turer-specific Data:
 Device Company ID: 0x6501 (No Match) - take with a grain of salt, not all companies populate this accurately!
          Endianness-flipped device company ID (in case the vendor used the wrong endianness): 0x0165 (Milwaukee Electric T
 Raw Data: 000c03249000
         In BT LE Data (LE_bdaddr_to_mf_specific), bdaddr_random = 0 (Public)
         This was found in an event of type 0 which corresponds to Connectable Undirected Advertising (ADV_IND)
 Device Company ID: 0x004c (Apple, Inc.) - take with a grain of salt, not all companies populate this accurately!
          Endianness-flipped device company ID (in case the vendor used the wrong endianness): 0x4c00 (No Match)
 Raw Data: 0215f34ebac47cd74027878e55f4e71d03090000000088
 Apple iBeacon:
         UUID128: f34ebac4-7cd7-4027-878e-55f4e71d0309
         Major ID: 0000
         Minor ID: 0000
         RSSI at 1 meter: -120dBm
         In BT LE Data (LE_bdaddr_to_mf_specific), bdaddr_random = 0 (Public)
         This was found in an event of type 3 which corresponds to Non-Connectable Undirected Advertising (ADV_NONCONN_IND)
```

# iBeacon: "The other UUID128!"

## UUID128Print

- Detective Work🕵️: GATT -> Google Search -> BLE112 == Bluegiga module -> Google Search -> Bluegiga == Silicon Labs -> iBeacons

```
turer-specific Data:
 Device Company ID: 0x6501 (No Match) - take with a grain of salt, not all companies populate this accurately!
          Endianness-flipped device company ID (in case the vendor used the wrong endianness): 0x0165 (Milwaukee Electric T
 Raw Data: 000c03249000
         In BT LE Data (LE_bdaddr_to_mf_specific), bdaddr_random = 0 (Public)
         This was found in an event of type 0 which corresponds to Connectable Undirected Advertising (ADV_IND)
 Device Company ID: 0x004c (Apple, Inc.) - take with a grain of salt, not all companies populate this accurately!
          Endianness-flipped device company ID (in case the vendor used the wrong endianness): 0x4c00 (No Match)
 Raw Data: 0215f34ebac47cd74027878e55f4e71d03090000000088
 Apple iBeacon:
         4-7cd7-4027-878e-55f4e71d0309
         Major ID: 0000
         Minor ID: 0000
         -120dBm
         In BT LE Data (LE_bdaddr_to_mf_specific), bdaddr_random = 0 (Public)
         This was found in an event of type 3 which corresponds to Non-Connectable Undirected Advertising (ADV_NONCONN_IND)
```

# iBeacon: "The other UUID128!"

## UUID128Print

- Detective Work🕵️: GATT -> Google Search -> BLE112 == Bluegiga module -> Google Search -> Bluegiga == Silicon Labs -> iBeacons

```
turer-specific Data:
 Device Company ID: 0x6501 (No Match) - take with a grain of salt, not all companies populate this accurately!
          Endianness-flipped device company ID (in case the vendor used the wrong endianness): 0x0165 (Milwaukee Electric T
 Raw Data: 000c03249000
         In BT LE Data (LE_bdaddr_to_mf_specific), bdaddr_random = 0 (Public)
         This was found in an event of type 0 which corresponds to Connectable Undirected Advertising (ADV_IND)
 Device Company ID: 0x004c (Apple, Inc.) - take with a grain of salt, not all companies populate this accurately!
          Endianness-flipped device company ID (in case the vendor used the wrong endianness): 0x4c00 (No Match)
 Raw Data: 0215f34ebac47cd74027878e55f4e71d03090000000088
 Apple iBeacon:
         4-7cd7-4027-878e-55f4e71d0309
         Major ID: 0000
         Minor ID: 0000
         -120dBm
         In BT LE Data (L                bdaddr_                (Public)
         This was found i                corre                n-Connectable Undirected Advertising (ADV_NONCONN_IND)
```

Kinda seems *uninitialized*

# iBeacon: "The other UUID128!"

## UUID128Print

- Detective Work🕵️: GATT -> Google Search -> BLE112 == Bluegiga module -> Google Search -> Bluegiga == Silicon Labs -> iBeacons -> UUID128

```
turer-specific Data:
 Device Company ID: 0x6501 (No Match) - take with a grain of salt, not all companies populate this accurately!
          Endianness-flipped device company ID (in case the vendor used the wrong endianness): 0x0165 (Milwaukee Electric T
 Raw Data: 000c03249000
         In BT LE Data (LE_bdaddr_to_mf_specific), bdaddr_random = 0 (Public)
         This was found in an event of type 0 which corresponds to Connectable Undirected Advertising (ADV_IND)
 Device Company ID: 0x004c (Apple, Inc.) - take with a grain of salt, not all companies populate this accurately!
          Endianness-flipped device company ID (in case the vendor used the wrong endianness): 0x4c00 (No Match)
 Raw Data: 0215f34ebac47cd74027878e55f4e71d03090000000088
 Apple iBeacon:
         UUID128: f34ebac4-7cd7-4027-878e-55f4e71d0309
         Major ID: 0000
         Minor ID: 0000
         RSSI at 1 meter: -120dBm
         In BT LE Data (LE_bdaddr_to_mf_specific), bdaddr_                    (Public)
         This was found in an event of type 3 which corre                    n-Connectable Undirected Advertising (ADV_NONCONN_IND)
```

# iBeacon: "The other UUID128!"

## UUID128Print

- Detective Work🕵️: GATT -> Google Search -> BLE112 == Bluegiga module -> Google Search -> Bluegiga == Silicon Labs -> iBeacons -> UUID128

```
turer-specific Data:
 Device Company ID: 0x6501 (No Match) - take with a grain of salt, not all companies populate this accurately!
          Endianness-flipped device company ID (in case the vendor used the wrong endianness): 0x0165 (Milwaukee Electric T
 Raw Data: 000c03249000
         In BT LE Data (LE_bdaddr_to_mf_specific), bdaddr_random = 0 (Public)
         This was found in an event of type 0 which corresponds to Connectable Undirected Advertising (ADV_IND)
 Device Company ID: 0x004c (Apple, Inc.) - take with a grain of salt, not all companies populate this accurately!
          Endianness-flipped device company ID (in case the vendor used the wrong endianness): 0x4c00 (No Match)
 Raw Data: 0215f34ebac47cd74027878e55f4e71d03090000000088
 Apple i
         UUID128: f34ebac4-7cd7-4027-878e-55f4e71d0309

         Minor ID: 0000
         RSSI at 1 meter: -120dBm
         In BT LE D                                                    (Public)
         This was f                                   n-Connectable Undirected Advertising (ADV_NONCONN_IND)
```

Where else does this UUID128 appear?

# iBeacon: "The other UUID128!"

## UUID128Print

- Detective Work🕵️: GATT -> Google Search -> BLE112 == Bluegiga module -> Google Search -> Bluegiga == Silicon Labs -> iBeacons -> UUID128

```
turer-specific Data:
 Device Company ID: 0x6501 (No Match) - take with a grain of salt, not all companies populate this accurately!
          Endianness-flipped device company ID (in case the vendor used the wrong endianness): 0x0165 (Milwaukee Electric T
 Raw Data: 000c03249000
         In BT LE Data (LE_bdaddr_to_mf_specific), bdaddr_random = 0 (Public)
         This was found in an event of type 0 which corresponds to Connectable Undirected Advertising (ADV_IND)
 Device Company ID: 0x004c (Apple, Inc.) - take with a grain of salt, not all companies populate this accurately!
          Endianness-flipped device company ID (in case the vendor used the wrong endianness): 0x4c00 (No Match)
 Raw Data: 0215f34ebac47cd74027878e55f4e71d03090000000088
 Apple i
         UUID128: f34ebac4-7cd7-4027-878e-55f4e71d0309
         Minor ID: 0000
         RSSI at 1 meter: -120dBm
         In BT LE Data (LE_bdaddr_to_mf_specific), bdaddr_                    (Public)
         This was found in an event of type 3 which corre                    n-Connectable Undirected Advertising (ADV_NONCONN_IND)
```

# iBeacon: "The other UUID128!"

## UUID128Print

- Detective Work🕵️: GATT -> Google Search -> BLE112 == Bluegiga module -> Google Search -> Bluegiga == Silicon Labs -> iBeacons -> UUID128

```
turer-specific Data:
 Device Company ID: 0x6501 (No Match) - take with a grain of salt, not all companies populate this accurately!
          Endianness-flipped device company ID (in case the vendor used the wrong endianness): 0x0165 (Milwaukee Electric T
 Raw Data: 000c03249000
         In BT LE Data (LE_bdaddr_to_mf_specific), bdaddr_random = 0 (Public)
         This was found in an event of type 0 which corresponds to Connectable Undirected Advertising (ADV_IND)
 Device Company ID: 0x004c (Apple, Inc.) - take with a grain of salt, not all companies populate this accurately!
          Endianness-flipped device company ID (in case the vendor used the wrong endianness): 0x4c00 (No Match)
 Raw Data: 0215f34ebac47cd74027878e55f4e71d03090000000088
 Apple i
         UUID128: f34ebac4-7cd7-4027-878e-55f4e71d0309
         Minor ID: 0000
         RSSI
         In B                                   bdaddr                    (Public)
         This                                    corre                    n-Connectable Undirected Advertising (ADV_NONCONN_IND)
```

Let's search for this UUID128, but remove anything associated with Milwaukee...

```
================================================================================
For bdaddr = 84:71:27:69:c3:dc:
        Company Name by IEEE OUI (84:71:27): Silicon Laboratories

        No BTC Extended Inquiry Result Device info.

        No Names found.

        No UUID16s found.

        No transmit power found.

        No Appearance data found.

        Manufacturer-specific Data:
                Device Company ID: 0x004c (Apple, Inc.) - take with a grain of salt, not all companies populate this accurately!
                         Endianness-flipped device company ID (in case the vendor used the wrong endianness): 0x4c00 (No Match)
                Raw Data: 0215f34ebac47cd74027878e55f4e71d03090000000088
                Apple iBeacon:
                        UUID128: f34ebac4-7cd7-4027-878e-55f4e71d0309
                        Major ID: 0000
                        Minor ID: 0000
                        RSSI at 1 meter: -120dBm
                        In BT LE Data (LE_bdaddr_to_mf_specific), bdaddr_random = 0 (Public)
                        This was found in an event of type 3 which corresponds to Non-Connectable Undirected Advertising (ADV_NONCONN_IND)

        No Class of Device Data found.

        No GATT Information found.

        No BLE 2thprint Info found.

        No BTC 2thprint Info found.

================================================================================
For bdaddr = 84:fd:27:34:d2:62:
        Company Name by IEEE OUI (84:fd:27): Silicon Laboratories

        No BTC Extended Inquiry Result Device info.

        No Names found.
```

```
================================================================================
For bdaddr = 84:71:27:69:c3:dc:
        Company Name by IEEE OUI (84:71:27): Silicon Laboratories

        No BTC Extended Inquiry Result Device info.

        No Names found.

        No UUID16s found.

        No transmit power found.

        No Appearance data found.

        Manufacturer-specific Data:
                Device Company ID: 0x004c (Apple, Inc.) - take with a grain of salt, not all companies populate this accurately!
                         Endianness-flipped device company ID (in case the vendor used the wrong endianness): 0x4c00 (No Match)
                Raw Data: 0215f34ebac47cd74027878e55f4e71d03090000000088
                Apple iBeacon:
                        UUID128: f34ebac4-7cd7-4027-878e-55f4e71d0309
                        Major ID: 0000
                        Minor ID: 0000
                        RSSI at 1 meter: -120dBm
                        In BT LE Data (LE_bdaddr_to_mf_specific), bdaddr_random = 0 (Public)
                        This was found in an event of type 3 which corresponds to Non-Connectable Undirected Advertising (ADV_NONCONN_IND)

        No Class of Device Data found.

        No GATT Information found.

        No BLE 2thprint Info found.

        No BTC 2thprint Info found.

================================================================================
For bdaddr = 84:fd:27:34:d2:62:
        Company Name by IEEE OUI (84:fd:27): Silicon Laboratories

        No BTC Extended Inquiry Result Device info.

        No Names found.
```

```
                        In BT LE Data (LE_bdaddr_to_mf_specific), bdaddr_random = 0 (Public)
                        This was found in an event of type 3 which corresponds to Non-Connectable Undirected Advertising (ADV_NONCONN_IND)

        No Class of Device Data found.

        No GATT Information found.

        No BLE 2thprint Info found.

================================================================================
For bdaddr = 88:6b:0f:0e:22:f0:
        Company Name by IEEE OUI (88:6b:0f) Bluegiga Technologies OY

        No BTC Extended Inquiry Result Device info.

        No Names found.

        No UUID16s found.

        No transmit power found.

        No Appearance data found.

        Manufacturer-specific Data:
                Device Company ID: 0x004c (Apple, Inc.) - take with a grain of salt, not all companies populate this accurately!
                         Endianness-flipped device company ID (in case the vendor used the wrong endianness): 0x4c00 (No Match)
                Raw Data: 0215f34ebac47cd74027878e55f4e71d03090000000088
                Apple iBeacon:
                        UUID128: f34ebac4-7cd7-4027-878e-55f4e71d0309
                        Major ID: 0000
                        Minor ID: 0000
                        RSSI at 1 meter: -120dBm
                        In BT LE Data (LE_bdaddr_to_mf_specific), bdaddr_random = 0 (Public)
                        This was found in an event of type 3 which corresponds to Non-Connectable Undirected Advertising (ADV_NONCONN_IND)

        No Class of Device Data found.

        No GATT Information found.

        No BLE 2thprint Info found.

        No BTC 2thprint Info found.
```

```
                        This was found in an event of type 3 which corresponds to Non-Connectable Undirected Advertising (ADV_NONCONN_IND)

        No Class of Device Data found.

        No GATT Information found.

        No BLE 2thprint Info found.

==============================================================================
For bdaddr = 88:6b:0f:0e:22:f0:
        Company Name by IEEE OUI (88:6b:0f) Bluegiga Technologies OY

        No BTC Extended Inquiry Result Device info.

        No Names found.

        No UUID16s found.

        No transmit power found.

        No Appearance data found.

        Manufacturer-specific Data:
                Device Company ID: 0x004c (Apple, Inc.) - take with a grain of                    accurately!
                         Endianness-flipped device company ID (in case the vend                   (No Match)
                Raw Data: 0215f34ebac47cd74027878e55f4e71d03090000000088
                Apple iBeacon:
                        UUID128: f34ebac4-7cd7-4027-878e-55f4e71d0309
                        Major ID: 0000
                        Minor ID: 0000
                        RSSI at 1 meter: -120dBm
                        In BT LE Data (LE_bdaddr_to_mf_specific), bdaddr_random
                        This was found in an event of type 3 which corresponds                    ing (A          N_IND)

        No Class of Device Data found.

        No GATT Information found.

        No BLE 2thprint Info found.

        No BTC 2thprint Info found.
```

Does this UUID128, ever appear with any device NOT associated with Silicon Labs or Bluegiga based on the OUI?

```
                        In BT LE Data (LE_bdaddr_to_mf_specific), bdaddr_random = 0 (Public)
                        This was found in an event of type 3 which corresponds to Non-Connectable Undirected Advertising (ADV_NONCONN_IND)

        No Class of Device Data found.

        No GATT Information found.

        No BLE 2thprint Info found.

================================================================================
For bdaddr = 88:6b:0f:0e:22:f0:
        Company Name by IEEE OUI (88:6b:0f): Bluegiga Technologies OY

        No BTC Extended Inquiry Result Device info.

        No Names found.

        No UUID16s found.

        No transmit power found.

        No Appearance data found.

        Manufacturer-specific Data:
                Device Company ID: 0x004c (Apple, Inc.) - take with a grain of salt, not all companies populate this accurately!
                         Endianness-flipped device company ID (in case the vendor used the wrong endianness): 0x4c00 (No Match)
                Raw Data: 0215f34ebac47cd74027878e55f4e71d03090000000088
                Apple iBeacon:
                        UUID128: f34ebac4-7cd7-4027-878e-55f4e71d0309
                        Major ID: 0000
                        Minor ID: 0000
                        RSSI at 1 meter: -120dBm
                        In BT LE Data (LE_bdaddr_to_mf_specific), bdaddr_random = 0 (Public)
                        This was found in an event of type 3 which corresponds to Non-Connectable Undirected Advertising (ADV_NONCONN_IND)

        No Class of Device Data found.

        No GATT Information found.

        No BLE 2thprint Info found.

        No BTC 2thprint Info found.
```

```
                        This was found in an event of type 3 which corresponds to Non-Connectable Undirected Advertising (ADV_NONCONN_IND)

        No Class of Device Data found.

        No GATT Information found.

        No BLE 2thprint Info found.

================================================================================
For bdaddr = 88:6b:0f:0e:22:f0:
        Company Name by IEEE OUI (88:6b:0f): Bluegiga Technologies OY

        No BTC Extended Inquiry Result Device info.

        No Names found.

        No UUID16s found.

        No transmit power found.

        No Appearance data found.

        Manufacturer-specific Data:
                Device Company ID: 0x004c (Apple, Inc.) - take with a grain o                    is accurately
                         Endianness-flipped device company ID (in case the ve                    00 (No Match)
                Raw Data: 0215f34ebac47cd74027878e55f4e71d03090000000088
                Apple iBeacon:
                        UUID128: f34ebac4-7cd7-4027-878e-55f4e71d0309
                        Major ID: 0000
                        Minor ID: 0000
                        RSSI at 1 meter: -120dBm
                        In BT LE Data (LE_bdaddr_to_mf_specific), bdaddr_ranc
                        This was found in an event of type 3 which correspond                    isin  DV  IND)

        No Class of Device Data found.

        No GATT Information found.

        No BLE 2thprint Info found.

        No BTC 2thprint Info found.
```

No! Therefore, there is a high probability that this UUID128, if present, is strongly indicative of Bluegiga/SiLabs chips!

# iBeacon: "The other UUID128!"

## UUID128Print

- Detective Work🕵️: GATT -> Google Search -> BLE112 == Bluegiga module -> Google Search -> Bluegiga == Silicon Labs -> iBeacons -> UUID128
- UUID128 **f34ebac4-7cd7-4027-878e-55f4e71d0309** == Bluegiga/ Silicon chips!
  - It doesn't hurt that the OUIPrint is corroborating ;)
- Note: I could be wrong, and it could be that this UUID128 is actually all Milwaukee equipment that is missing the other IDs like UUID16 or Milwaukee MSD or GATT info
- This is why we need more crowdsourced data!

# iBeacon: "The other UUID128!"

## UUID128Print

# iBeacon: "The other UUID128!"

## UUID128Print

- Plot Twist! The Bluegiga *module* was based on a TI *chip*!

# iBeacon: "The other UUID128!"

## UUID128Print

- Plot Twist! The Bluegiga *module* was based on a TI *chip*!

## 7 Block diagram

BLE112 is based on TI's CC2540 chip. Embedded 32 MHz and 32.678 kHz crystals are used for clock generation. Matched balun and low pass filter provide optimal radio performance with extremely low spurious emissions. Small ceramic chip antenna gives good radiation efficiency even when the module is used in layouts with very limited space. https://www.silabs.com/documents/public/data-sheets/BLE112-DataSheet.pdf

# iBeacon: "The other UUID128!"

## UUID128Print

- Plot Twist! The Bluegiga *module* was based on a TI *chip*!

## 7 Block diagram

BLE112 is based on TI's CC2540 chip Embedded 32 MHz and 32.678 kHz crystals are used for clock generation. Matched balun and low pass filter provide optimal radio performance with extremely low spurious emissions. Small ceramic chip antenna gives good radiation efficiency even when the module is used in layouts with very limited space. https://www.silabs.com/documents/public/data-sheets/BLE112-DataSheet.pdf

# But sometimes...

Individual Device → Device Type → Model

Firmware Version ← BT Chip ← Manufacturer

# But sometimes...

Individual Device → Device Type → Model

Firmware Version ← BT Chip ← Manufacturer

UUID128: **f34ebac4-7cd7-4027-878e-55f4e71d0309**

# But sometimes...

# But sometimes...

# But sometimes...

*ASSUMPTION*:
UUID128Prints can be reused *within* Manufacturers,
but not reused *between* Manufacturers

# 2thprint by SDP

# Prior Work

**"Blueprinting - Remote Device Identification based on Bluetooth Fingerprinting Techniques"**

- [1] by Herfurt & Mulliner from 2004 connected to BTC devices via Service Discovery Protocol (SDP) and created a hash from selected data within the available profiles
  - I'm simply using the deprecated "sdptool" from BlueZ with its existing XML output option. *But I haven't decided how to process the data yet!*

[1] http://www.mulliner.org/collin/publications/Blueprinting.pdf

# Prior Work

**"Blueprinting - Remote Device Identification based on Bluetooth Fingerprinting Techniques"**

- [1] by Herfurt & Mulliner from 2004 connected to BTC devices via Service Discovery Protocol (SDP) and created a hash from selected data within the available profiles
  - I'm simply using the deprecated "sdptool" from BlueZ with its existing XML output option. *But I haven't decided how to process the data yet!*

[1] http://www.mulliner.org/collin/publications/Blueprinting.pdf

# Prior Work

**"Blueprinting - Remote Device Identification based on Bluetooth Fingerprinting Techniques"**

- [1] by Herfurt & Mulliner from 2004 connected to BTC devices via Service Discovery Protocol (SDP) and created a hash from selected data within the available profiles
  - I'm simply using the deprecated "sdptool" from BlueZ with its existing XML output option. *But I haven't decided how to process the data yet!*

[1] http://www.mulliner.org/collin/publications/Blueprinting.pdf

# 2thprint by GATT mostly 🤾*

* I would tend to argue though that GATT UUID16/UUID128s included in ADV_IND or SCAN_RSP are passive and semi-passive respectively

# 2thprint by GATT

## *GATTPrint*

- **G**eneric **Att**ribute *Profile* (GATT) is a sort of weird thing that's built on top of Attribute Protocol (ATT), which is the actual protocol for sending & receiving data
- Mostly used on BLE, though it can technically be used on BTC devices too
- You can *theoretically* get ALL the types of information through GATT!

# 2thprint by GATT

## *GATTPrint*

- **G**eneric **Att**ribute Profile (GATT) is a sort of weird thing that's built on top of Attribute Protocol (ATT), which is the actual protocol for sending & receiving data
- Mostly used on BLE, though it can technically be used on BTC devices too
- You can *theoretically* get ALL the types of information through GATT!

# 2thprint by GATT

## *GATTPrint*

- **G**eneric **Att**ribute Profile (GATT) is a sort of weird thing that's built on top of Attribute Protocol (ATT), which is the actual protocol for sending & receiving data
- Mostly used on BLE, though it can technically be used on BTC devices too
- You can *theoretically* get ALL the types of information through GATT!

# GATT on Phones

- When you open a Bluetooth scanner app like LightBlue or BluefruitConnect on your phone, and they show you information, usually that is GATT information
- This is from BluefruitConnect

# GATT on Phones

- When you open a Bluetooth scanner app like LightBlue or BluefruitConnect on your phone, and they show you information, usually that is GATT information
- This is from BluefruitConnect

Individual Device → Device Type → Model → Manufacturer → BT Chip → Firmware Version

# GATT on Phones

- When you open a Bluetooth scanner app like LightBlue or BluefruitConnect on your phone, and they show you information, usually that is GATT information
- This is from BluefruitConnect

Individual Device → Device Type → Model → Manufacturer → BT Chip → Firmware Version

# GATT on Phones

- When you open a Bluetooth scanner app like LightBlue or BluefruitConnect on your phone, and they show you information, usually that is GATT information
- This is from BluefruitConnect

Individual Device → Device Type → Model → Manufacturer → BT Chip → Firmware Version

# GATT on Phones

- When you open a Bluetooth scanner app like LightBlue or BluefruitConnect on your phone, and they show you information, usually that is GATT information
- This is from BluefruitConnect

Individual Device → Device Type → Model → Manufacturer → BT Chip → Firmware Version

# GATT on Phones

- When you open a Bluetooth scanner app like LightBlue or BluefruitConnect on your phone, and they show you information, usually that is GATT information
- This is from BluefruitConnect

# GATT on Phones

- When you open a Bluetooth scanner app like LightBlue or BluefruitConnect on your phone, and they show you information, usually that is GATT information
- This is from BluefruitConnect

# GATT on Phones

- When you open a Bluetooth scanner app like LightBlue or BluefruitConnect on your phone, and they show you information, usually that is GATT information
- This is from BluefruitConnect

# GATT in General

- We may instead get a whole bunch of *nothing useful* from GATT
- Devices may not respond to GATT requests, and when they do, we're in no way guaranteed to get "characteristics" which contain the kind of information we want
  - GATT inquiries can take a few seconds. For moving targets, there is a low probability of success to perform a full information collection without a strong transmitter and good antenna

# GATT in General

- We may instead get a whole bunch of *nothing useful* from GATT
- Devices may not respond to GATT requests, and when they do, we're in no way guaranteed to get "characteristics" which contain the kind of information we want
  - GATT inquiries can take a few seconds. For moving targets, there is a low probability of success to perform a full information collection without a strong transmitter and good antenna

# Prior Work

**"Fingerprinting Bluetooth-Low-Energy Devices Based on the Generic Attribute Profile"**

- [1] by Celosia & Cunche from 2019 connected to BLE devices via GATT and just called all the returned information the fingerprint
  - Noteworthy, in my mind, for the different threat model (user privacy), that called out basically everything *except* firmware revision string as interesting

> We identified that information found in GATT profiles can be used to infer the following information: **device type**, **device model**, **device manufacturer** and **user's name**. All this information can threaten the privacy of the device owner.

[1] https://inria.hal.science/hal-02359914/file/paper.pdf

# Prior Work

**"Fingerprinting Bluetooth-Low-Energy Devices Based on the Generic Attribute Prof**

You know what's really a threat to user privacy? Getting pwned over-the-air ;)

- [1] by Celosia & Cunche from 2019 c ... 5 an just called all the returned informatic
  - Noteworthy, in my mind, for the different threat model (user privacy), that called out basically everything *except* firmware revision string as interesting

> We identified that information found in GATT profiles can be used to infer the following information: **device type**, **device model**, **device manufacturer** and **user's name**. All this information can threaten the privacy of the device owner.

[1] https://inria.hal.science/hal-02359914/file/paper.pdf

**Table 4: Average time to collect a GATT profile among different devices.**

| Device type | Device | Time (sec) |
|---|---|---|
| Lightbulb | Osram Smart+ | 6.531 |
| Motion sensor | Eve Motion | 6.468 |
| Socket outlet | Eve Energy | 5.919 |
| Smartphone | Apple iPhone 8 | 4.354 |
| Smartphone | Apple iPhone 6 | 4.259 |
| Keyring | Nut | 4.148 |
| TV dongle | Google Chromecast | 3.660 |
| Fitness wristband | Fitbit Inspire | 3.231 |
| Presentation remote | Logitech Spotlight | 2.860 |
| Smartwatch | Apple Watch Series 3 | 2.853 |
| Heart rate monitor | Polar H7 | 2.751 |
| Fitness wristband | Fitbit Flex | 2.552 |
| Headset | Bose SoundLink Around-Ear II | 2.181 |
| Speaker | Divacore Ktulu2+ | 1.742 |
| Keyring | Chipolo | 1.426 |
| | Average | 3.662 |

[1] https://inria.hal.science/hal-02359914/file/paper.pdf

**Table 4: Average time to collect a GATT profile among different devices.**

| Device type | Device | Time (sec) |
|---|---|---|
| Lightbulb | Osram Smart+ | 6.531 |
| Motion sensor | Eve Motion | 6.468 |
| Socket outlet | Eve Energy | 5.919 |
| Smartphone | Apple iPhone 8 | 4.354 |
| Smartphone | Apple iPhone 6 | 4.259 |
| Keyring | Nut | 4.148 |
| TV dongle | Google Chromecast | 3.660 |
| Fitness wristband | Fitbit Inspire | 3.231 |
| Presentation remote | Logitech Spotlight | 2.860 |
| Smartwatch | Apple Watch Series 3 | 2.853 |
| Heart rate monitor | Polar H7 | 2.751 |
| Fitness wristband | Fitbit Flex | 2.552 |
| | ound-Ear II | 2.181 |
| | | 1.742 |
| | | 1.426 |
| | | 3.662 |

I've seen things take up to 24 seconds to reply to GATTprinting...

[1] https://inria.hal.science/hal-02359914/file/paper.pdf

**Table 4: Average time to collect a GATT profile among different devices.**

| Device type | Device | Time (sec) |
|---|---|---|
| Lightbulb | Osram Smart+ | 6.531 |
| Motion sensor | Eve Motion | 6.468 |
| Socket outlet | Eve Energy | 5.919 |
| Smartphone | Apple iPhone 8 | 4.354 |
| Smartphone | Apple iPhone 6 | 4.259 |
| Keyring | Nut | 4.148 |
| TV dongle | Google Chromecast | 3.660 |
| Fitness wristband | Fitbit Inspire | 3.231 |
| Presentation remote | Logitech Spotlight | 2.860 |
| Smartwatch | Apple Watch Series 3 | 2.853 |
| Heart rate monitor | Polar H7 | 2.751 |
| Fitness wristband | Fitbit Flex | 2.552 |
| Headset | Bose SoundLink Around-Ear II | 2.181 |
| Speaker | Divacore Ktulu2+ | 1.742 |
| Keyring | Chipolo | 1.426 |
| | Average | 3.662 |

[1] https://inria.hal.science/hal-02359914/file/paper.pdf

**Table 4: Average time to collect a GATT profile among different devices.**

| Device type | Device | Time (sec) |
|---|---|---|
| Lightbulb | Osram Smart+ | 6.531 |
| Motion sensor | Eve Motion | 6.468 |
| Socket outlet | Eve Energy | 5.919 |
| Smartphone | Apple iPhone 8 | 4.354 |
| Smartphone | Apple iPhone 6 | 4.259 |
| Keyring | Nut | 4.148 |
| TV dongle | Google Chromecast | 3.660 |
| Fitness wristband | Fitbit Inspire | 3.231 |
| Presentation remote | Logitech Spotlight | 2.860 |
| Smartwatch | Apple Watch Series 3 | 2.853 |
| Heart rate monitor | Polar H7 | 2.751 |
| Fitness wristband | Fitbit Flex | 2.552 |
| Headset | Bose SoundLink Around-Ear II | 2.181 |
| Speaker | Divacore Ktulu2+ | 1.742 |
| Keyring | Chipolo | 1.426 |
| | Average | 3.662 |

[1] https://inria.hal.science/hal-02359914/file/paper.pdf

| | All | |
|---|---|---|
| | % | # |
| Device Name | 99.49 | 13182 |
| Appearance | 99.48 | 13082 |
| Service Changed | 0.02 | 2 |
| Manufacturer Name String | 99.48 | 9177 |
| Model Number String | 99.36 | 9158 |
| Battery Level | 2.45 | 191 |
| Current Time | 0.41 | 31 |
| Peripheral Preferred Connection Parameters | 99.90 | 1051 |
| Software Revision String | 97.04 | 1017 |
| Hardware Revision String | 95.95 | 996 |
| Serial Number String | 97.12 | 979 |
| Firmware Revision String | 94.99 | 835 |

[1] https://inria.hal.science/hal-02359914/file/paper.pdf

| | All | |
|---|---|---|
| | % | # |
| Device Name | 99.49 | 13182 |
| Appearance | 99.48 | 13082 |
| Service Changed | 0.02 | 2 |
| Manufacturer Name String | 99.48 | 9177 |
| Model Number String | 99.36 | 9158 |
| Battery Level | 2.45 | 191 |
| Current Time | 0.41 | 31 |
| Peripheral Preferred Connection Parameters | 99.90 | 1051 |
| Software Revision String | 97.04 | 1017 |
| Hardware Revision String | 95.95 | 996 |
| Serial Number String | 97.12 | 979 |
| Firmware Revision String | 94.99 | 835 |

At least 13182 devices reported a Name (and it was readable 99.49% of the time)

[1] https://inria.hal.science/hal-02359914/file/paper.pdf

| | All | |
|---|---|---|
| | % | # |
| Device Name | 99.49 | 13182 |
| Appearance | 99.48 | 13082 |
| Service Changed | 0.02 | 2 |
| Manufacturer Name String | 99.48 | 9177 |
| Model Number String | 99.36 | 9158 |
| Battery Level | 2.45 | 191 |
| Current Time | 0.41 | 31 |
| Peripheral Preferred Connection Parameters | 99.90 | 1051 |
| Software Revision String | 97.04 | 1017 |
| Hardware Revision String | 95.95 | 996 |
| Serial Number String | 97.12 | 979 |
| Firmware Revision String | 94.99 | 835 |

At least 13182 devices reported a Name (and it was readable 99.49% of the time)

***Only 835 devices reported a Firmware Revision String!***
(and it was readable 94.49% of the time)

[1] https://inria.hal.science/hal-02359914/file/paper.pdf

| | All | |
|---|---|---|
| | % | # |
| Device Name | 99.49 | 13182 |
| Appearance | 99.48 | 13082 |
| Service Changed | 0.02 | 2 |
| Manufacturer Name String | 99.48 | 9177 |
| Model Number String | 99.36 | 9158 |
| Battery Level | 2.45 | 191 |
| Current Time | 0.41 | 31 |
| Peripheral Preferred Connection Parameters | 99.90 | 1051 |
| Software Revision String | 97.04 | 1017 |
| Hardware Revision String | 95.95 | 996 |
| Serial Number String | 97.12 | 979 |
| Firmware Revision String | 94.99 | 835 |

At least 13182 devices reported a Name (and it was readable 99.49% of the time)

***Only 835 devices reported a Firmware Revision String!***
(and it was readable 94.49% of the time)

*Note: Their dataset is heavily skewed by containing at least 9924 iPhones, and iPhones don't report a Firmware Revision String*

[1] https://inria.hal.science/hal-02359914/file/paper.pdf

| | All | |
|---|---|---|
| | % | # |
| Device Name | 99.49 | 13182 |
| Appearance | 99.48 | 13082 |
| Service Changed | 0.02 | 2 |
| Manufacturer Name String | 99.48 | 9177 |
| Model Number String | 99.36 | 9158 |
| Battery Level | 2.45 | 191 |
| Current Time | 0.41 | 31 |
| Peripheral Preferred Connection Parameters | 99.90 | 1051 |
| Software Revision String | 97.04 | 1017 |
| Hardware Revision String | 95.95 | 996 |
| Serial Number String | 97.12 | 979 |
| Firmware Revision String | 94.99 | 835 |

At least 13182 devices reported a Name (and it was readable 99.49% of the time)

***Only 835 devices reported a Firmware Revision String!***
(and it was readable 94.49% of the time)

*Note: Their dataset is heavily skewed by containing at least 9924 iPhones, and iPhones don't report a Firmware Revision String*

[1] https://inria.hal.science/hal-02359914/file/paper.pdf

| | All | |
|---|---|---|
| | % | # |
| Device Name | 99.49 | 13182 |
| Appearance | 99.48 | 13082 |
| Service Changed | 0.02 | 2 |
| Manufacturer Name String | 99.48 | 9177 |
| Model Number String | 99.36 | 9158 |
| Battery Level | 2.45 | 191 |
| Current Time | 0.41 | 31 |
| Peripheral Preferred Connection Parameters | 99.90 | 1051 |
| Software Revision String | 97.04 | 1017 |
| Hardware Revision String | 95.95 | 996 |
| Serial Number String | 97.12 | 979 |
| Firmware Revision String | 94.99 | 835 |

At least 13182 devices reported a Name (and it was readable 99.49% of the time)

***Only 835 devices reported a Firmware Revision String!***
(and it was readable 94.49% of the time)

*Note: Their dataset is heavily skewed by containing at least 9924 iPhones, and iPhones don't report a Firmware Revision String*

[1] https://inria.hal.science/hal-02359914/file/paper.pdf

# Prior Work

**"Fingerprinting Bluetooth-Low-Energy Devices Based on the Generic Attribute Profile"**

- [1] by Zuo et al. from 2019 connected to BLE devices via Generic Attribute Profile (GATT) and just called all the returned information the fingerprint
  - Here it's referring to more like a "device-*model*" fingerprint, not "individual-device" fingerprint
  - Same basic idea as Celosia & Cunche 2019, but they claimed they only did passive scanning for what was advertised, and they didn't connect to devices and request the GATT information for "ethical reasons" (Which I don't buy. More like CYA.)
  - Most interestingly, they scraped a bunch of Android applications to attempt to extract GATT-related UUID128s via static analysis

[1] https://web.cse.ohio-state.edu/~lin.3021/file/CCS19a.pdf

# My GATTPrint-er

- The BlueZ *(deprecated)* "gatttool" seemed to already do a good job of collecting this information. But the output wasn't easily machine-parsable. So I modified the source to store info to a more easily machine-parsable log file
  - Pro tip: If you try to use the higher-layer BT APIs (such as those available via most Python->BT libraries), you will not be able to collect this info without first pairing. But pairing isn't actually necessary for most characteristics of most devices!
    - Some devices though may refuse read/write requests on access control grounds, due to lack of the encryption/authentication that comes as a result of pairing

# Individual

## UUID128Print

- Detective Work🕵️: GATT -> Google Search "00001100-d102-11e1-9b23-00025b00a5a5" -> Cambridge Silicon Radio (CSR) CSR102x chips' over the air update protocol!

```
./TellMeEverything.py --UUID128regex "00001100-d102-11e1-9b23-00025b00a5a5" | grep DeviceName | sort | uniq -c
   2  DeviceName: BW-FYE13
   2  DeviceName: Crusher ANC
   3  DeviceName: Crusher Evo
   2  DeviceName: HK385
   1  DeviceName: Ink'd+ Active
   7  DeviceName: Jabra Elite 3
   4  DeviceName: Jabra Elite 4 Active
   1  DeviceName: LE-Device
   1  DeviceName: LE-Headset
   1  DeviceName: LE-MINOR III
  16  DeviceName: LE-OpenRun Pro
   1  DeviceName: LE_ATH-TWX9
   7  DeviceName: LG-TONE-FP9_LE
   2  DeviceName: M50
   3  DeviceName: Shure AONIC TW2
   1  DeviceName: Tribit FlyBuds C2
   1  DeviceName: [AV] MX-T40
   2  DeviceName: al hydrajolt
   2  DeviceName: mini lifejacket
   2  DeviceName: mini lifejacket jolt
```

# Bluetooth Low Energy upgrade protocols

## Bluetooth Low Energy Characteristics

To transport packets, the following Bluetooth Low Energy service and characteristics are used.

- Service
  - Name: “CSR GAIA”
  - UUID: `00001100-d102-11e1-9b23-00025b00a5a5`
- Characteristic 1 from “CSR GAIA”
  - Name: “CSR GAIA Command Endpoint”
  - UUID: `00001101-d102-11e1-9b23-00025b00a5a5`
- Characteristic 2 from “CSR GAIA”
  - Name: “CSR GAIA Response Endpoint”
  - UUID: `00001102-d102-11e1-9b23-00025b00a5a5`



https://developer.qualcomm.com/qfile/34081/csr102x_otau_overview.pdf

# Individual

## UUID128Print

- Detective Work🕵️: GATT -> Google Search "6e400001-b5a3-f393-e0a9-e50e24dcca9e" -> Nordic chips' UART GATT Service (but advertised via SCAN_RSP)

```
For bdaddr = c5:09:25:d3:5f:29:
        Company Name by IEEE OUI (c5:09:25): No Match

        No BTC Extended Inquiry Result Device info.

        DeviceName: IoTVD35F29
                In BT LE Data (LE_bdaddr_to_name), bdaddr_random = 1 (Random Static)
                        NamePrint: match found for ^IoTV[A-F0-9]{6}$: Veoride Electric kick scooters and e-bikes (last 6 are last 6 of BDADDR)
                This was found in an event of type 19 which corresponds to (New wireshark) Connectable, Scannable

        UUID16s found:
                UUID16 0x180a (GATT Service ID: Device Information)
                         Found in BT LE data (LE_bdaddr_to_UUID16s)
                This was found in an event of type 27 which corresponds to (New wireshark) Connectable, Scannable, Scan Response

        UUID128s found:
                UUID128 6e400001-b5a3-f393-e0a9-e50e24dcca9e (Custom UUID128: Nordic UART Service)
                        Found in BT LE data (LE_bdaddr_to_UUID128s)
                        This was found in an event of type 27 which corresponds to (New wireshark) Connectable, Scannable, Scan Response
```

# Top 20 GATT Services in my data FWIW

## 2023-10-26

| UUID128 | Count | |
|---|---|---|
| 00001800-0000-1000-8000-00805f9b34fb | 21127 | -> "Generic Access" (GAP) |
| 00001801-0000-1000-8000-00805f9b34fb | 20675 | -> Generic Attribute (GATT) |
| 0000180a-0000-1000-8000-00805f9b34fb | 18044 | -> Device Information |
| 9fa480e0-4967-4542-9390-d343dc5d04ae | 15738 | -> Apple Nearby |
| d0611e78-bbb4-4591-a5f8-487910ae4366 | 15737 | -> Apple Continuity |
| 0000180f-0000-1000-8000-00805f9b34fb | 11284 | -> Battery |
| 00001805-0000-1000-8000-00805f9b34fb | 10641 | -> Current Time |
| 89d3502b-0f36-433a-8ef4-c502ad55f8dc | 10636 | -> Apple Media Service |
| 7905f431-b5ce-4e99-a40f-4b1e122d00d0 | 10636 | -> Apple Notification Center Service |
| 0000febe-0000-1000-8000-00805f9b34fb | 843 | -> Bose |
| 0000fe03-0000-1000-8000-00805f9b34fb | 598 | -> Amazon Alexa |
| 0000fe2c-0000-1000-8000-00805f9b34fb | 421 | -> Google Fast Pair |
| 9aa4730f-b25c-4cc3-b821-c931559fc196 | 359 | -> Apple Watch? (my data seems to support) |
| eedd5e73-6aa8-4673-8219-398a489da87c | 280 | -> Samsung SmartTag Authentication Service |
| 0000fd69-0000-1000-8000-00805f9b34fb | 246 | -> Samsung SmartTag Offline Finding Advertisement |
| 0000feed-0000-1000-8000-00805f9b34fb | 234 | -> Tile |
| 0000180d-0000-1000-8000-00805f9b34fb | 222 | -> Heart Rate |
| eed6d5cc-c3b2-4d7b-8c6b-7acbf7965bb6 | 204 | -> Samsung Galaxy Watch (based on my data) |
| 00001855-0000-1000-8000-00805f9b34fb | 202 | -> Telephony and Media Audio |
| 0000184c-0000-1000-8000-00805f9b34fb | 196 | -> Generic Telephone Bearer |

# Top 20 GATT Characteristics in my data FWIW

## 2023-10-26

| char_UUID128 | Count | |
|---|---|---|
| 00002a00-0000-1000-8000-00805f9b34fb | 19592 | -> Device Name |
| 00002a01-0000-1000-8000-00805f9b34fb | 19205 | -> Appearance |
| 00002a05-0000-1000-8000-00805f9b34fb | 18443 | -> Service Changed |
| 00002a29-0000-1000-8000-00805f9b34fb | 16776 | -> Manufacturer Name String |
| 00002a24-0000-1000-8000-00805f9b34fb | 16650 | -> Model Number String |
| af0badb1-5b99-43cd-917a-a77bc549e3cc | 15194 | -> Apple Nearby Characteristic |
| 8667556c-9a37-4c91-84ed-54ee27d90049 | 15193 | -> Apple Continuity Characteristic |
| 00002a19-0000-1000-8000-00805f9b34fb | 10632 | -> Battery Level |
| 00002a2b-0000-1000-8000-00805f9b34fb | 10245 | -> Current Time |
| 9fbf120d-6301-42d9-8c58-25e699a21dbd | 10194 | -> Apple Notification Center Notification Source |
| 69d1d8f3-45e1-49a8-9821-9bbdfdaad9d9 | 10194 | -> Apple Notification Center Control Point |
| 22eac6e9-24d6-4bb5-be44-b36ace7c7bfb | 10192 | -> Apple Notification Center Data Source |
| 9b3c81d8-57b1-4a8a-b8df-0e56f7ca51c2 | 10185 | -> Apple Media Center Remote Command |
| 2f7cabce-808d-411f-9a0c-bb92ba96c102 | 10183 | -> Apple Media Center Entity Update |
| c6b2f38c-23ab-46d8-a6ab-a3a870bbd5d7 | 10183 | -> Apple Media Center Entity Attribute |
| 00002a0f-0000-1000-8000-00805f9b34fb | 10176 | -> Local Time Information |
| 00002a26-0000-1000-8000-00805f9b34fb | 1682 | -> Firmware Revision String |
| 00002aa6-0000-1000-8000-00805f9b34fb | 1655 | -> Central Address Resolution |
| 00002a04-0000-1000-8000-00805f9b34fb | 1644 | -> Peripheral Preferred Connection Parameters |
| 00002a28-0000-1000-8000-00805f9b34fb | 1563 | -> Software Revision String |

# 2thprint by "Pics or it didn't happen"

📸

# 2thprint by looking up teardown pictures :P

## This is not scalable...unless we crowdsource it!

- When one wants to know what a very specific device has for a chip, one can just google for teardown pictures!
- Sometimes the FCC (or other wireless regulatory authorities') database "internal photos" are useful in this regard

# E Tu Rivian?

- Regex: ^Rivian Sensor [1234]$ e.g. Rivian Sensor 1
- Regex: ^Rivian Phone Key$
- Regex: ^Rivian Camp Speaker$

https://apps.fcc.gov/eas/GetApplicationAttachment.html?id=5383917

# E Tu Rivian?

- Regex: ^Rivian Sensor [1234]$ e.g. Rivian Sensor 1
- Regex: ^Rivian Phone Key$
- Regex: ^Rivian Camp Speaker$

Address type: **Public** (0x00)
Address: AC:4D:16:FD:40:93 (OUI AC-4D-16)
Name (complete): Rivian Sensor 3

https://apps.fcc.gov/eas/GetApplicationAttachment.html?id=5383917

# E Tu Rivian?

- Regex: ^Rivian Sensor [1234]$ e.g. Rivian Sensor 1
- Regex: ^Rivian Phone Key$
- Regex: ^Rivian Camp Speaker$

Address type: **Public** (0x00)
Address: AC:4D:16:FD:40:93 (OUI AC-4D-16) ← Actually Texas Instruments btmon just didn't have it in its vendor database
Name (complete): Rivian Sensor 3

```
For bdaddr = AC:4D:16:FD:40:93:
        Company Name by IEEE OUI (AC:4D:16): Texas Instruments

        No BTC Extended Inquiry Result Device info.

        DeviceName: Rivian Sensor 3
                In BT LE Data (LE_bdaddr_to_name), bdaddr_random = 0 (Public)
```

https://apps.fcc.gov/eas/GetApplicationAttachment.html?id=5383917

# E Tu

- Rege
- Rege
- Rege

Address
Address
Name (c

```
For bdaddr = A
        Compan

        No BTC

        Device
```

https://apps.fcc.gov/e

# E Tu

- Rege
- Rege
- Rege

Address
Address
Name (c

```
For bdaddr = A
        Compan

        No BTC

        Device
```

https://apps.fcc.gov/e

# Common Case

# Common Case

# Common Case

# Putting It All *Toothgether* 😜

# Putting It All *Toothgether* 😜

# Reminder, even if wireless links weren't *lossy,* you're not generally going to have all the data for every device

# Reminder, even if wireless links weren't *lossy,* you're not generally going to have all the data for every device

# Blue2thprint

# Recap

## Things that can narrow down to a silicon vendor or module maker

- BDADDR OUI (BTC & BLE Public)
- BTC/BLE Advertised Names
- BTC/BLE MSD Company ID
- UUID16s (advertised or GATT)
- UUID128s (advertised or GATT or MSD via beacons)

⭐ BLE LL_VERSION_IND CompId field

⭐ BTC LMP_vendor_rsp CompId field

- BTC EIR/SDP Device ID Profile data
- GATT Device Name
- GATT Manufacturer Name
- Looking up teardown pictures ;)

⭐ = Current highest S/N ratio

But what % of the time are each of these applicable?
And how often are none of them applicable?
The research will continue until security improves!

# Code to collect & analyze your own data

- TBD GitHub URL

# Call To Action!

# Conclusion

- Bluetooth *vulnerability assessment* **is not yet a thing you can really do**!
- This is an active research topic I'm working on, but it needs more researchers working on it
- The starting point, as always, is to read related work
- I've organized the related work into a TiddlyWiki that I will continue to update over the coming years, and which others can contribute to via github PRs
  - https://darkmentor.com/bt.html

# Ongoing work

- How should I structure the 2thprints data?!
- What % of devices respond to GATT/SDP/LMP2th/LL2th/etc requests?
  - Working with a student on a research project to investigate more rigorously
- What other packet types (teef!) can we add to improve the 2thprint? (e.g. L2CAP? RFCOMM?)
- Can we do the work they didn't do in [1] to automatically generate minimal chip model differentiation packet sequences from existing learned state machines?

[1] https://link.springer.com/article/10.1007/s10703-023-00425-y

# Fin

- BT research is cool
- But *OpenSecurityTraining2 (https://ost2.fyi , @OpenSecTraining) is cooler!*
  - We'll have BT classes eventually, but in the meantime there's so much other stuff to learn! Reverse Engineering, Vulnerabilities, Firmware, System Architecture!
- You should take a class, or *teach* a class!

# Backup

# 2thprint by Class of Device (CoD)

## Primarily applicable to BTC (but some rare BLE devices use it too)

- BTC Extended Inquiry Response (EIR) packets contain a 24-bit CoD value

### 2.8 Class of Device

Referenced from the following:

- Bluetooth Core Specification [Vol 2] Part B, Section 6.5.1.4 [4].
- Supplement to the Bluetooth Core Specification Part A, Section 1.6.2 [22].

The Class of Device is composed of four fields: A Major Service Classes bitfield, a Major Device Class enumerated value, the Minor Device Classes, and a fixed value of 0b00 in the two least significant bits. The format of the Minor Device Class is determined by the Major Device Class value. The structure of the Class of Device is defined below:

| 23 – 13 | 12 – 8 | 7 – 2 | 1 – 0 |
|---|---|---|---|
| Major Service Classes | Major Device Class | Minor Device Class | 0b00 |

*Figure 2.2: Class of Device format*

### 2.8.2 Major Device Classes

The Miscellaneous major device class is used where a more specific Major Device Class code is not suitable. A device that does not have a major class code assigned can use the Uncategorizied: device code not specified until ”classified.”

Last Modified: 2022-05-25

| 12 | 11 | 10 | 9 | 8 | Major Device Class |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | Miscellaneous |
| 0 | 0 | 0 | 0 | 1 | Computer (desktop, notebook, PDA, organizer, ...) |
| 0 | 0 | 0 | 1 | 0 | Phone (cellular, cordless, pay phone, modem, ...) |
| 0 | 0 | 0 | 1 | 1 | LAN/Network Access point |
| 0 | 0 | 1 | 0 | 0 | Audio/Video (headset, speaker, stereo, video display, VCR, ...) |
| 0 | 0 | 1 | 0 | 1 | Peripheral (mouse, joystick, keyboard, ...) |
| 0 | 0 | 1 | 1 | 0 | Imaging (printer, scanner, camera, display, ...) |
| 0 | 0 | 1 | 1 | 1 | Wearable |
| 0 | 1 | 0 | 0 | 0 | Toy |
| 0 | 1 | 0 | 0 | 1 | Health |
| 1 | 1 | 1 | 1 | 1 | Uncategorized: device code not specified |

### 2.8.1 Major Service Classes

Last Modified: 2022-05-25

| Bit | Class of Device Major Service Class |
|---|---|
| 13 | Limited Discoverable Mode |
| 14 | LE audio |
| 15 | Reserved for future use |
| 16 | Positioning (Location identification) |
| 17 | Networking (LAN, Ad hoc, ...) |
| 18 | Rendering (Printing, Speakers, ...) |
| 19 | Capturing (Scanner, Microphone, ...) |
| 20 | Object Transfer (v-Inbox, v-Folder, ...) |
| 21 | Audio (Speaker, Microphone, Headset service, ...) |
| 22 | Telephony (Cordless telephony, Modem, Headset service, ...) |
| 23 | Information (WEB-server, WAP-server, ...) |

Individual Device → Device Type → Model
Firmware Version ← BT Chip ← Manufacturer

### 2.8.2 Major Device Classes

The Miscellaneous major device class is used where a more specific Major Device Class code is not suitable. A device that does not have a major class code assigned can use the Uncategorizied: device code not specified until ”classified.”

Last Modified: 2022-05-25

| 12 | 11 | 10 | 9 | 8 | Major Device Class |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | Miscellaneous |
| 0 | 0 | 0 | 0 | 1 | Computer (desktop, notebook, PDA, organizer, ...) |
| 0 | 0 | 0 | 1 | 0 | Phone (cellular, cordless, pay phone, modem, ...) |
| 0 | 0 | 0 | 1 | 1 | LAN/Network Access point |
| 0 | 0 | 1 | 0 | 0 | Audio/Video (headset, speaker, stereo, video display, VCR, ...) |
| 0 | 0 | 1 | 0 | 1 | Peripheral (mouse, joystick, keyboard, ...) |
| 0 | 0 | 1 | 1 | 0 | Imaging (printer, scanner, camera, display, ...) |
| 0 | 0 | 1 | 1 | 1 | Wearable |
| 0 | 1 | 0 | 0 | 0 | Toy |
| 0 | 1 | 0 | 0 | 1 | Health |
| 1 | 1 | 1 | 1 | 1 | Uncategorized: device code not specified |

### 2.8.1 Major Service Classes

Last Modified: 2022-05-25

| Bit | Class of Device Major Service Class |
|---|---|
| 13 | Limited Discoverable Mode |
| 14 | LE audio |
| 15 | Reserved for future use |
| 16 | Positioning (Location identification) |
| 17 | Networking (LAN, Ad hoc, ...) |
| 18 | Rendering (Printing, Speakers, ...) |
| 19 | Capturing (Scanner, Microphone, ...) |
| 20 | Object Transfer (v-Inbox, v-Folder, ...) |
| 21 | Audio (Speaker, Microphone, Headset service, ...) |
| 22 | Telephony (Cordless telephony, Modem, Headset service, ...) |
| 23 | Information (WEB-server, WAP-server, ...) |

Individual Device → Device Type → Model → Manufacturer → BT Chip → Firmware Version

# BTW

The Bluetooth *Ecosystem* Is ***ARCHITECTURALLY UNSECURABLE***

# UEFI Vulnerability Proliferation

- **If an attacker finds a vulnerability in the UEFI "reference implementation," its proliferation across IBVs and OEMs would potentially be wide spread.**
  - More on how this theory works "in practice" later…

© 2014 The MITRE Corporation. All rights reserved.

https://www.blackhat.com/docs/us-14/materials/us-14-Kallenberg-Extreme-Privilege-Escalation-On-Windows8-UEFI-Systems.pdf

# UEFI Vulnerability Proliferation

- **If an attacker finds a vulnerability in the UEFI "reference implementation," its proliferation across IBVs and OEMs would potentially be wide spread.**
  - More on how this theory works "in practice" later…

© 2014 The MITRE Corporation. All rights reserved.

https://www.blackhat.com/docs/us-14/materials/us-14-Kallenberg-Extreme-Privilege-Escalation-On-Windows8-UEFI-Systems.pdf

# UEFI Vulnerability Proliferation

SIG

- **If an attacker finds a vulnerability in the UEFI "reference implementation," its proliferation across IBVs and OEMs would potentially be wide spread.**
  - More on how this theory works "in practice" later…

© 2014 The MITRE Corporation. All rights reserved.

https://www.blackhat.com/docs/us-14/materials/us-14-Kallenberg-Extreme-Privilege-Escalation-On-Windows8-UEFI-Systems.pdf

# UEFI Vulnerability Proliferation

SIG Silicon

- **If an attacker finds a vulnerability in the UEFI "reference implementation," its proliferation across IBVs and OEMs would potentially be wide spread.**
  - More on how this theory works "in practice" later…

© 2014 The MITRE Corporation. All rights reserved.

https://www.blackhat.com/docs/us-14/materials/us-14-Kallenberg-Extreme-Privilege-Escalation-On-Windows8-UEFI-Systems.pdf

# UEFI Vulnerability Proliferation

SIG
Silicon
Intermediaries

- **If an attacker finds a vulnerability in the UEFI "reference implementation," its proliferation across IBVs and OEMs would potentially be wide spread.**
  - More on how this theory works "in practice" later…

© 2014 The MITRE Corporation. All rights reserved.

https://www.blackhat.com/docs/us-14/materials/us-14-Kallenberg-Extreme-Privilege-Escalation-On-Windows8-UEFI-Systems.pdf

# UEFI Vulnerability Proliferation

SIG
Silicon
Intermediaries
Device
Makers

- **If an attacker finds a vulnerability in the UEFI "reference implementation," its proliferation across IBVs and OEMs would potentially be wide spread.**
  - More on how this theory works "in practice" later…

© 2014 The MITRE Corporation. All rights reserved.

https://www.blackhat.com/docs/us-14/materials/us-14-Kallenberg-Extreme-Privilege-Escalation-On-Windows8-UEFI-Systems.pdf

Bluetooth®

DST2.FYI

SIG

Bluetooth®

SIG

Bluetooth®

## Silicon Vendors (>20)

BROADCOM
Qualcomm CSR
Texas Instruments
Silicon Labs
Realtek
Nordic Semiconductor
Samsung Semiconductors
intel
MediaTek
Espressif
Cypress Embedded in Tomorrow
NXP
Telink
Renesas
Microchip
Infineon
freescale semiconductor
YiCHiP
oppo
Marvell
ISSC Technologies Corp.
BES Technic
Beken
em microelectronic
A COMPANY OF THE SWATCH GROUP

SIG

Bluetooth®

## Silicon Vendors (>20)

Silicon

BROADCOM
Qualcomm
CSR
Texas Instruments
Silicon Labs
Realtek
Samsung Semiconductors
intel
MediaTek
Espressif
Cypress Embedded in Tomorrow
Infineon
NXP
freescale semiconductor
Telink
Renesas
Microchip
YiChiP
oppo
Marvell
ISSC Technologies Corp.
BES Technic
Beken
em microelectronic
A company of the Swatch Group

SIG

Bluetooth®

## Silicon Vendors (>20)

Silicon

## *Module*-Makers (IDEK how many)

SIG

Bluetooth®

## Silicon Vendors (>20)

Silicon

## *Module*-Makers (IDEK how many)

Intermediaries

# SIG

Bluetooth®

## Silicon Vendors (>20)

# Silicon

## *Module*-Makers (IDEK how many)

# Intermediaries

## Product-Makers

***3330*** *registered* with Bluetooth SIG as of the time of writing!

SIG

Bluetooth®

**Silicon Vendors (>20)**

Silicon

**Module-Makers (IDEK how many)**

Intermediaries

**Product-Makers**

Device Makers ***3330*** *registered* with Bluetooth SIG as of the time of writing!

# LG Innotek

"LG Innotek has developed a Nordic nRF51822 SoC-based Bluetooth Smart (previously known as Bluetooth low energy) module"

So for my purposes, it will almost certainly present as a Nordic chip, even though it's technically custom LG silicon.

From
https://www.embedded.com/lg-innotek-develops-bluetooth-smart-module-that-measures-just-5-3-x-4-7-x-1-0-mm/

*n*Blue BR-LE5.0-S1A Module Summary Datasheet
Copyright © 2002-2021 BlueRadios, Inc.
v1.3

# BT5.0 Low Energy Single Mode Class 1 SoC Module

# *n*Blue™ BR-LE5.0-S1A (nRF52840)

- ***AT HOME. AT WORK. ON THE ROAD. USING BT5.0 LOW ENERGY WIRELESS TECHNOLOGY MEANS TOTAL FREEDOM FROM THE CONSTRAINTS AND CLUTTER OF WIRES IN YOUR LIFE.***
- **FCC, IC, CE, RoHS, and BT5.0 Certified ISM 2.4GHz module supporting BT5.0 high speed mode, long range mode and advertising extensions. Can also support BT5.0 Mesh, 802.15.4 for Thread and Zigbee, ANT or proprietary 2.4Ghz.**
- **Utilizes the Nordic nRF52840 SoC. 64Mhz ARM® Cortex™ M4F 32-bit processor with FPU, 1MB Flash, 256K RAM, built in DC-DC converter and ARM CryptoCell cryptographic accelerator.**
- **Programmable output power from -40dBm to +8dBm for short to long range applications.**
- **Over 1000 meter line of site distance with integrated antenna. External antenna can be connected to RF_OUT pad or through optional u.FL connector (requires moving RF path resistor).**
- **Can be externally controlled via simple ASCII AT commands over UART, USB and BT5.0, or programmed with custom applications embedded in the module.**

FCC CE -Certified- RoHS Compliant

https://www.blueradios.com/nBlue%20BR-LE5.0-S1A%20Summary%20Datasheet.pdf

*n*Blue BR-LE5.0-S1A Module Summary Datasheet
Copyright © 2002-2021 BlueRadios, Inc.
v1.3

# BT5.0 Low Energy Single Mode Class 1 SoC Module

# *n*Blue™ BR-LE5.0-S1A (nRF52840)

- *AT HOME. AT WORK. ON THE ROAD. USING BT5.0 LOW ENERGY WIRELESS TECHNOLOGY MEANS TOTAL FREEDOM FROM THE CONSTRAINTS AND CLUTTER OF WIRES IN YOUR LIFE.*
- FCC, IC, CE, RoHS, and BT5.0 Certified ISM 2.4GHz module supporting BT5.0 high speed mode, long range mode and advertising extensions. Can also support BT5.0 Mesh, 802.15.4 for Thread and Zigbee, ANT or proprietary 2.4Ghz.
- Utilizes the Nordic nRF52840 SoC. 64Mhz ARM® Cortex™ M4F 32-bit processor with FPU, 1MB Flash, 256K RAM, built in DC-DC converter and ARM CryptoCell cryptographic accelerator.
- Programmable output power from -40dBm to +8dBm for short to long range applications.
- Over 1000 meter line of site distance with integrated antenna. External antenna can be connected to RF_OUT pad or through optional u.FL connector (requires moving RF path resistor).
- Can be externally controlled via simple ASCII AT commands over UART, USB and BT5.0, or programmed with custom applications embedded in the module.

FCC CE -Certified- RoHS Compliant

https://www.blueradios.com/nBlue%20BR-LE5.0-S1A%20Summary%20Datasheet.pdf

# REALTEK → Ai-Thinker Technology

| RTL87XX series | | | | |
|---|---|---|---|---|
| **Model** | BW12 | BW14 | BW15 | BW16（hot） |
| **Picture** | | | | |
| **Chip** | RTL8710BX | RTL8710BX | RTL8720CF | RTL8720DN |
| **Package** | SMT-16 | / | SMD-16 | SMD-16 |
| **Size** | 24 x16x 3mm (LxWxH) ±0.2mm | 50.5*29.2*3.3（±0.2）mm | 24*16*3(±0.2)MM | 24*16*3(±0.2)MM |
| **Antenna** | on-board PCB/ IPEX antenna | on-board PCB/ IPEX antenna | on-board PCB/ IPEX antenna | on-board PCB/ IPEX antenna |
| **Frequency range** | 2400-2483.5MHz | 2400 ~ 2483.5MHz | 2.402GHz -2.480GHz | 2400-2483.5MHz or 5180-5825MHz |
| **Bluetooth** | / | / | Bluetooth4.2 BLE | BLE and BT5.0 |
| **Operating temperature** | -20~+85° C | -20℃~70℃ | -40 ℃ ~ 85 ℃ | -20 ℃ ~ 70 ℃ |
| **Storage temperature** | -40 ~125°C | -40℃~125℃ | -40 ℃ ~ 125 ℃ , < 90%RH | -40 ℃ ~ 125 ℃ , < 90%RH |
| **Power supply** | 3.3±10%V | 5V ~ 12V | Voltage 3.0 V ~3.6 V, current >500 mA | Voltage 3.0V ~ 3.6V，Typical 3.3V，Current >450mA |
| **Interface** | UART,I2C, SPI, GPIO, SWD, PWM | UART | UART/GPIO/ADC/PWM /IIC /SPI | UART/GPIO/ADC/PWM/IIC/SPI/SWD |

→ ???

From
https://docs.ai-thinker.com/en/rtl87xx

DST2.FYI

晶讯 JINGXUN

TOSPO
FSL 佛山照明
阳光照明
NVC 雷士照明
美的 Midea
BULL公牛
HONYAR 鸿雁
CDN LIGHT
GE Lighting
UOX
EGLO my light | my style
EAGLERISE 伊戈尔
HARMAN A SAMSUNG COMPANY
ONKYO
TCL
3nod
酷狗音乐
Pioneer 先锋
COSONIC
audio-technica
JVC The Perfect Experience
PHILIPS
TOSHIBA
步步高
SONY
Yanan 亚南
Panasonic
1+

From
https://www.jingxun.xyz/en/portal/page/index/id/34.html