Beacons

L2CAP

iOS 13 New Additions

Beacons

- Small, battery powered wireless device
- Advertises its presence and services via continuous broadcasting
- Used for proximity-aware applications
- Pseudo-standard running on BLE (e.g. iBeacon on iOS/OS X, Eddystone on Android)

Beacon Application Examples

Retail Shopping

Inventory Tracking

Point-of-sale systems

Indoor Navigation

Interactive Experiences

Fast-food drive-throughs

iBeacon Advertisement Packet

Field	Size	Description
UUID	16 bytes	Application specific identifier
Major	2 bytes	Specify specific iBeacon and use case
Minor	2 bytes	Allow further subdivision of region or use case

Scenario: Trade Show Event

Event Location		Las Vegas	Salt Lake City	Seattle
UUID		76AF7B38-3E95-439F-A879-8799A0DD964D		
Major		1	2	3
Minor	Registration	10	10	10
	Raffle	20	20	20
	Demo Area	30	30	30

iOS Apps Detect iBeacons With Core Location

CLLocationManager

CLBeaconRegion

CLBeacon

Beacon Monitoring

1. Determine availability and authorization status of region monitoring with CLLocationManager

2. Define beacon region to be monitored via CLBeaconRegion class

3. Register beacon region with location manager

4. Handle boundary-crossing events for a beacon region through CLLocationManagerDelegate callbacks

5. Range beacons to determine proximity with CLBeacon

Determine Availability and Authorization Status

```
if CLLocationManager.isMonitoringAvailable(for: CLBeaconRegion.self) {
    switch CLLocationManager.authorizationStatus() {
    case .authorizedWhenInUse, .authorizedAlways:
        startMonitoringBeaconRegion()
    default:
        // request authorization
    }
}
```

Define and Register Beacon Region

```
func startMonitoringBeaconRegion() {

    manager.delegate = self

    let uuidString = "BF276819-6939-4A79-AEEA-21F6BB27A901"

    let uuid = UUID(uuidString: uuidString)!

    let identifier = "myIdentifier"

    let region = CLBeaconRegion(proximityUUID: uuid, identifier:
identifier)

    // register the beacon region

    manager.startMonitoring(for: region)

}
```

Handle Boundary-Crossing Events

```
func locationManager(_ manager: CLLocationManager, didEnterRegion
region: CLRegion) {

    guard let region = region as? CLBeaconRegion else { return }

    if CLLocationManager.isRangingAvailable() {

        manager.startRangingBeacons(in: region)

    }

}
```

Handle Boundary-Crossing Events

```
func locationManager(_ manager: CLLocationManager, didExitRegion
region: CLRegion) {

    guard let region = region as? CLBeaconRegion else { return }

    manager.stopRangingBeacons(in: region)

}
```

Determine Proximity of Ranged Beacons

```
func locationManager(_ manager: CLLocationManager, didRangeBeacons
beacons: [CLBeacon], in region: CLBeaconRegion) {

    if let beacon = beacons.first {

        switch beacon.proximity {

            // proximity-based logic ...

        }

    }

}
```

iOS Devices Can Act As iBeacons

1. Generate Beacon Region
2. Build Peripheral Data
3. Start Advertising

Generate a Beacon Region

```
let uuidString = "BF276819-6939-4A79-AEEA-21F6BB27A901"

let uuid = UUID(uuidString: uuidString)!

let majorValue = CLBeaconMajorValue(1)

let minorValue = CLBeaconMinorValue(2)

let identifier = "myIdentifier"


let beaconRegion = CLBeaconRegion(proximityUUID: uuid, major:
majorValue, minor: minorValue, identifier: identifier)
```

Build a Peripheral Dictionary

```
let peripheralData: NSMutableDictionary =
beaconRegion.peripheralData(withMeasuredPower: nil)


/* peripheralData =

{

  kCBAdvDataAppleBeaconKey:

      <bf276819 69394a79 aeea21f6 bb27a901 00010002 c5>

}

*/
```

Start Advertising Beacon

```
peripheralManager.startAdvertising(peripheralData)
```

L2CAP

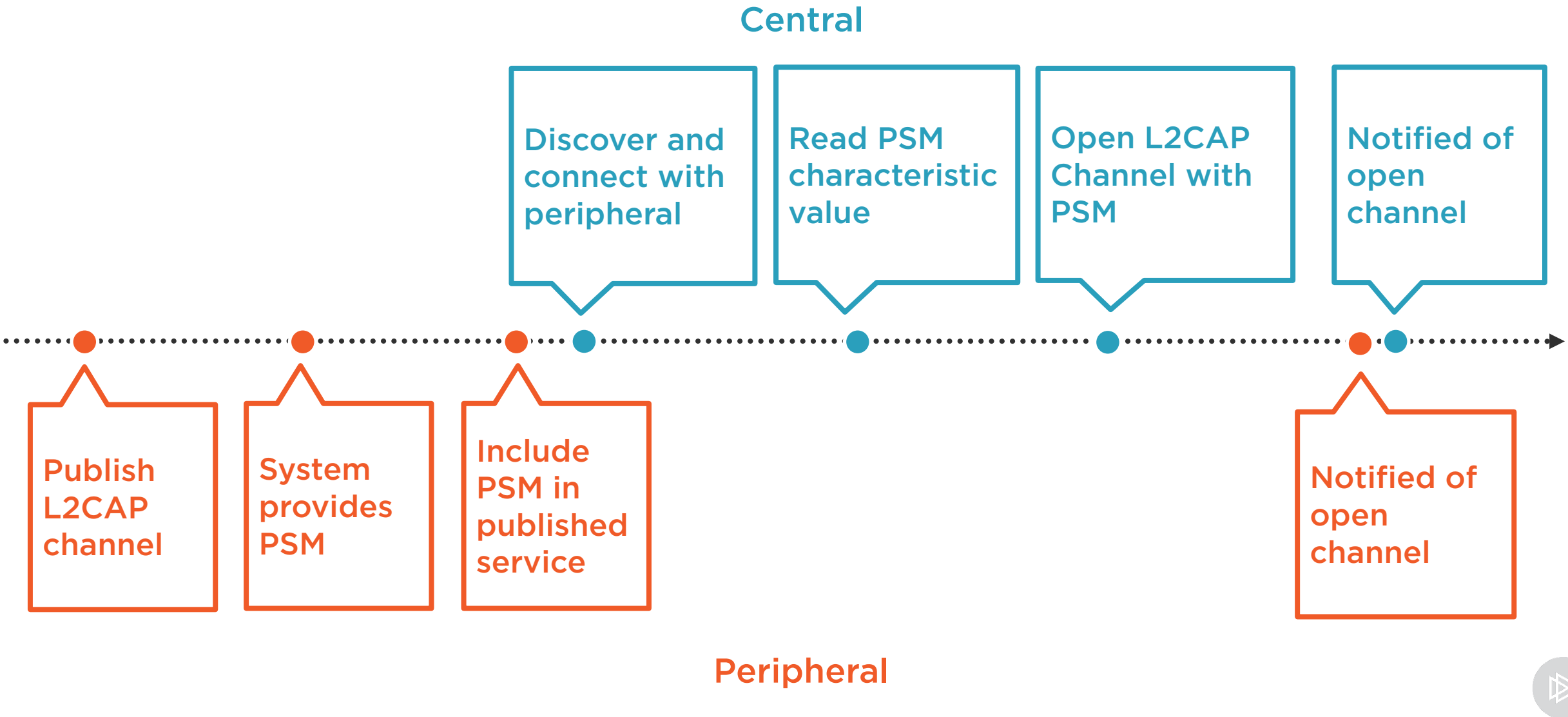
L2CAP

Logical Link Control and Adaptation Protocol

L2CAP Channel

- Stream of data between two devices
- Dynamically allocated channel
- Directly communicate with connected accessory without framing or packet size limitations
- Provides low overhead and high performance
- Recommended use case for large data transfers (e.g. firmware updates)

Core Bluetooth L2CAP Channel Flowchart



PSM

Protocol service multiplexer

```
public typealias CBL2CAPPSM = UInt16
```

PSM Can Be Likened to a TCP Port

A PSM channel is defined by the peripheral

```
class CBPeripheralManager : CBManager {
    @available(iOS 11.0, *)
    func publishL2CAPChannel(withEncryption encryptionRequired: Bool)
}
```

Peripherals Publish L2CAP Channels

Requiring encryption is recommended to prevent eavesdropping or man-in-the-middle (MITM) attacks

```
public protocol CBPeripheralManagerDelegate : NSObjectProtocol {

    @available(iOS 6.0, *)

    optional func peripheralManager(_ peripheral: CBPeripheralManager,
didPublishL2CAPChannel PSM: CBL2CAPPSM, error: Error?)

}
```

System Provides PSM When Published

```
@available(iOS 11.0, *)
public let CBUUIDL2CAPPSMCharacteristicString: String
```

CBUUIDL2CAPPSMCharacteristicString

The PSM (a little endian uint16_t) of an L2CAP Channel associated with the GATT service containing this characteristic

Reading PSM Characteristic Value

```
func peripheral(_ peripheral: CBPeripheral, didUpdateValueFor
characteristic: CBCharacteristic, error: Error?) {

    guard let value = characteristic.value, error == nil else {return}

    if characteristic.uuid.uuidString ==
CBUUIDL2CAPPSMCharacteristicString {

        guard let psm = try? JSONDecoder().decode(CBL2CAPPSM.self,
from: value) else { return }

        peripheral.openL2CAPChannel(psm)

    }

}
```

Generating a PSM Characteristic

```
var characteristics: [CBMutableCharacteristic] = ...

if let data = try? JSONEncoder().encode(PSM) {

    let psmCharacteristic = CBMutableCharacteristic(type:
CBUUID(string: CBUUIDL2CAPPSMCharacteristicString), properties: .read,
value: data, permissions: .readable) // .readEncryptionRequired

    characteristics.append(psmCharacteristic)
}

let service = CBMutableService(type: serviceUUID, primary: true)

service.characteristics = characteristics

manager.add(service)
```

```
class CBPeripheral : CBPeer {
    @available(iOS 11.0, *)
    func openL2CAPChannel(_ PSM: CBL2CAPPSM)
}
```

Open L2CAP Channel on Connected Peripheral

L2CAP Open Channel Delegates

```
public protocol CBPeripheralManagerDelegate :
NSObjectProtocol {

    optional func peripheralManager(_ peripheral:
CBPeripheralManager, didOpen channel: CBL2CAPChannel?,
error: Error?)

}

public protocol CBPeripheralDelegate : NSObjectProtocol {

    optional func peripheral(_ peripheral: CBPeripheral,
didOpen channel: CBL2CAPChannel?, error: Error?)

}
```

CBL2CAPChannel

```
@available(iOS 11.0, *)
open class CBL2CAPChannel : NSObject {
    open var peer: CBPeer! { get } // remote device
    open var inputStream: InputStream! { get } // read
    open var outputStream: OutputStream! { get } // write
    open var psm: CBL2CAPPSM { get }
}
```

Closing L2CAP Channels

Bluetooth link loss

Central manually closes channel

Peripheral unpublishes channel

Peripheral released from memory

iOS 13 Additions to Core Bluetooth

iOS 13 Additions to Core Bluetooth

- LE 2Mbps – faster and more power efficient device communication
- Discover and communicate with Bluetooth classic devices
- Privacy updates
- PacketLogger developer tool updates

Core Bluetooth Now Supports LE 2 Mbps

(iPhone 8 and later, Apple TV 4K, Apple Watch Series 4)

LE 2 Mbps	Advertising Extensions	Extended Scan	Extended Connections
		iPhone XS, latest iPad Pro	iPhone XS, latest iPad Pro
Physical layer rate increased from 1 to 2 Mbps	**Uses data channel to send larger payloads (31 -> 255 bytes)**	**Scan for extended advertisements**	**Improved connection process**

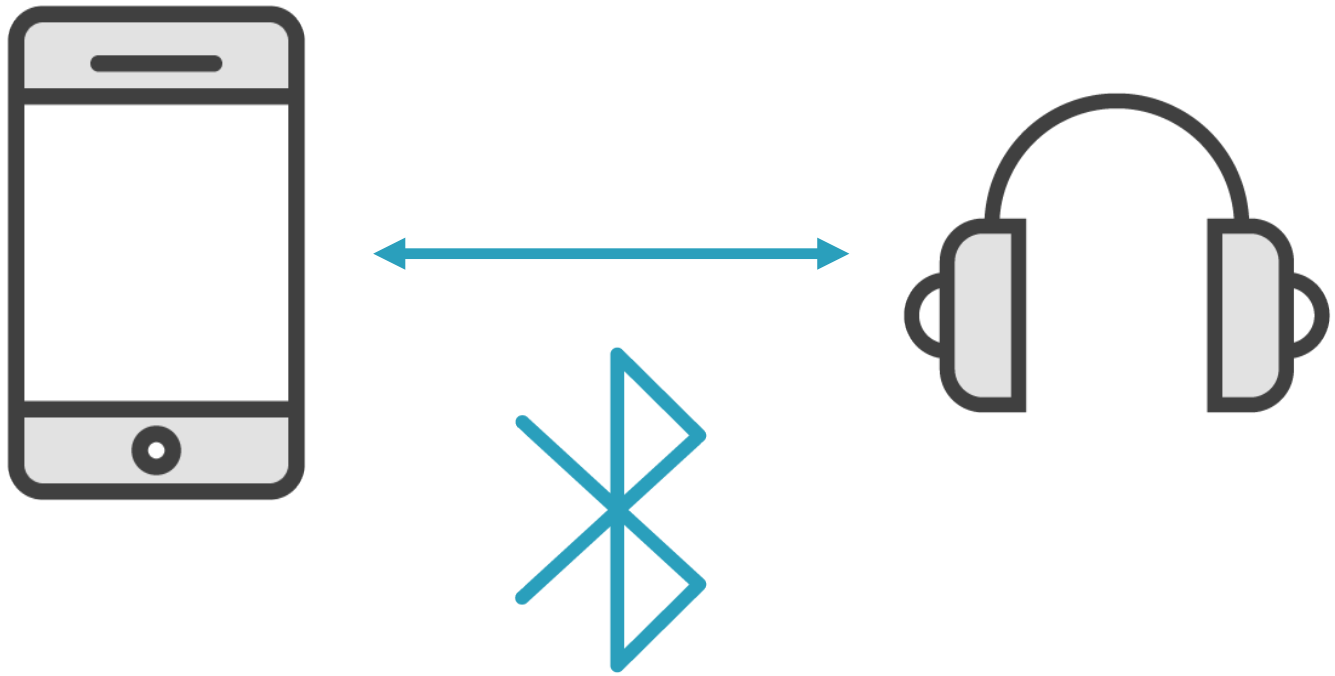
Transparent to application – no API changes required

iOS13 – Core Bluetooth can now be used with Classic Bluetooth devices

GATT runs over BR/EDR protocol

CBPeripheral APIs are unchanged

CBCentralManager can now be notified when a Classic Bluetooth connection occurs



Core Bluetooth with BR/EDR

1. **Register for connection events with CBCentralManager**

2. **Delegate callback sent when system finds a matching connection**
   - also sent after registration if matching connection is already established

Registering for Connection Events

Service UUID

Peripheral UUID

New CBCentralManager API

```
open class CBCentralManager : CBManager {

    @available(iOS 13.0, *)

    open func registerForConnectionEvents(options:
    [CBConnectionEventMatchingOption : Any]?)

}

extension CBConnectionEventMatchingOption {

    @available(iOS 13.0, *)

    public static let serviceUUIDs: CBConnectionEventMatchingOption

    @available(iOS 13.0, *)

    public static let peripheralUUIDs: CBConnectionEventMatchingOption

}
```

Registering for Connection Events

```
let serviceUUIDs: [CBUUID] = [CBUUID(string: uuidString)]
let options: [CBConnectionEventMatchingOption: Any] =
    [CBConnectionEventMatchingOption.serviceUUIDs: serviceUUIDs]
manager.registerForConnectionEvents(options: options)
```

New CBCentralManagerDelegate API

```
public protocol CBCentralManagerDelegate : NSObjectProtocol {

    @available(iOS 13.0, *)

    optional func centralManager(_ central: CBCentralManager,
connectionEventDidOccur event: CBConnectionEvent, for peripheral:
CBPeripheral)

}

public enum CBConnectionEvent : Int {

    case peerDisconnected

    case peerConnected

}
```

Listening for BR/EDR Connection Events

```
func centralManager(_ central: CBCentralManager,
connectionEventDidOccur event: CBConnectionEvent, for peripheral:
CBPeripheral)

    switch event {

    case .peerConnected:

        peripheral.connect()

        manager.registerForConnectionEvents(options: nil)

    case .peerDisconnected:

        // Perform cleanup ...

    }

}
```

Incoming BR/EDR Connection Flow

Register for connection events with CBCentralManager

User attempts to connect to a discovered BR/EDR device in Bluetooth Settings

Pairing request is triggered

After connection, system runs service discovery of GATT services

CBCentralManagerDelegate callback sent when system finds a connection

Handle BR/EDR connection event (e.g. calling connect on CBPeripheral)

Clear registration of connection events

Outgoing BR/EDR Connection Flow

Want to connect to a known BR/EDR paired device

Tell CBCentralManager to connect to the CBPeripheral

If app is foregrounded, system attempts connection by paging device

If connection successful, CBCentralManagerDelegate is notified

iOS 13 Privacy and Developer Tools Updates

Privacy Updates – User Authorization

iOS 12 and earlier – only required for background advertising

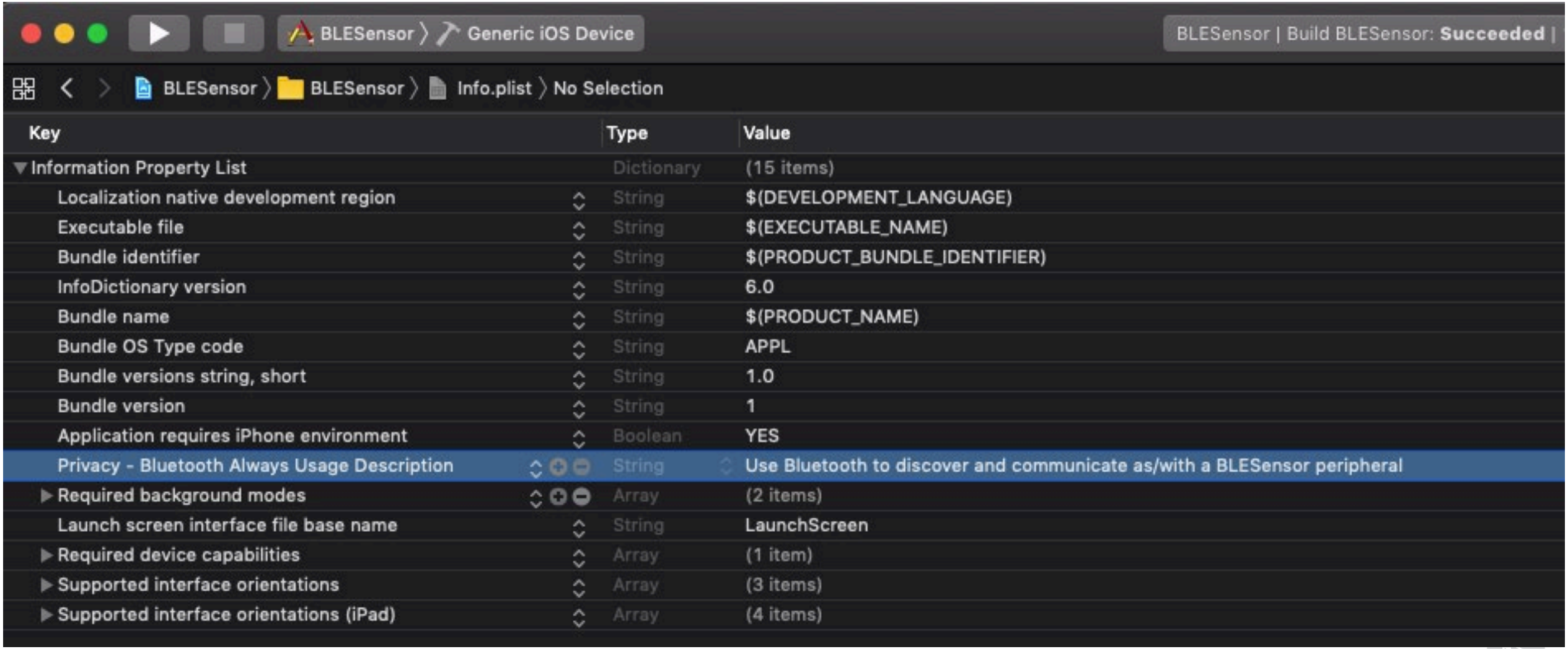
iOS 13 – required when using ANY Core Bluetooth API

- (also applies to apps built on older SDKs)

Required on iOS, watchOS, tvOS

Can be modified in the Settings app

Privacy Updates – Usage Description String



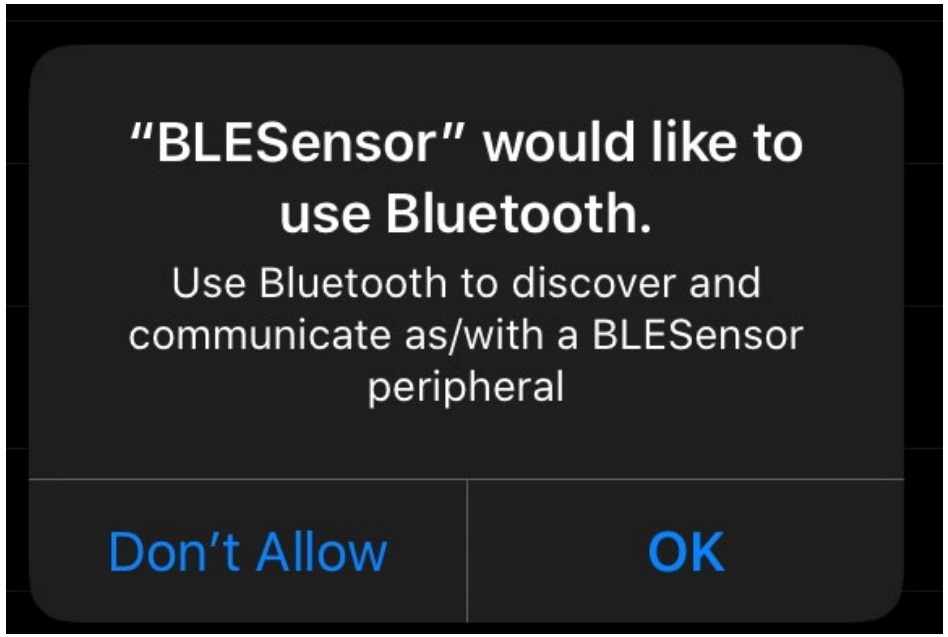
NSBluetoothAlwaysUsageDescription

Results of Using CoreBluetooth API in iOS 13

Apps Without Privacy Usage Key

BLESensor[7360:563969] [access] This app has crashed because it attempted to access privacy-sensitive data without a usage description. The app's Info.plist must contain an NSBluetoothAlwaysUsageDescription key with a string value explaining to the user how the app uses this data.

Apps With Privacy Usage Key



New Type: CBManagerAuthorization

```
// Represents the current authorization state of a CBManager
@available(iOS 13.0, *)
public enum CBManagerAuthorization : Int {
    case notDetermined
    case restricted
    case denied
    case allowedAlways
}
```

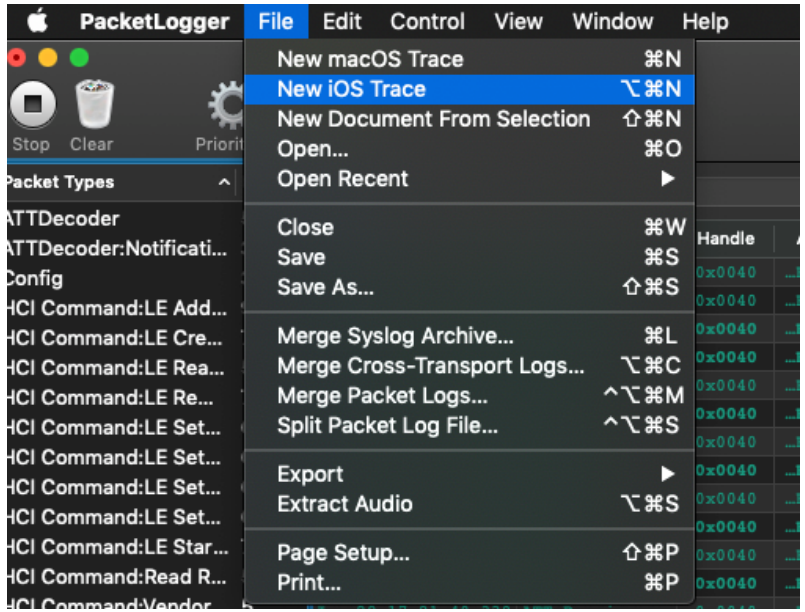
Privacy Updates – iOS 12

```
func centralManagerDidUpdateState(_ central: CBCentralManager) {
    if central.state == .poweredOn {
        scanForPeripherals()
    } else {
        print("central is unavailable: \(central.state.rawValue)")
    }
}
```

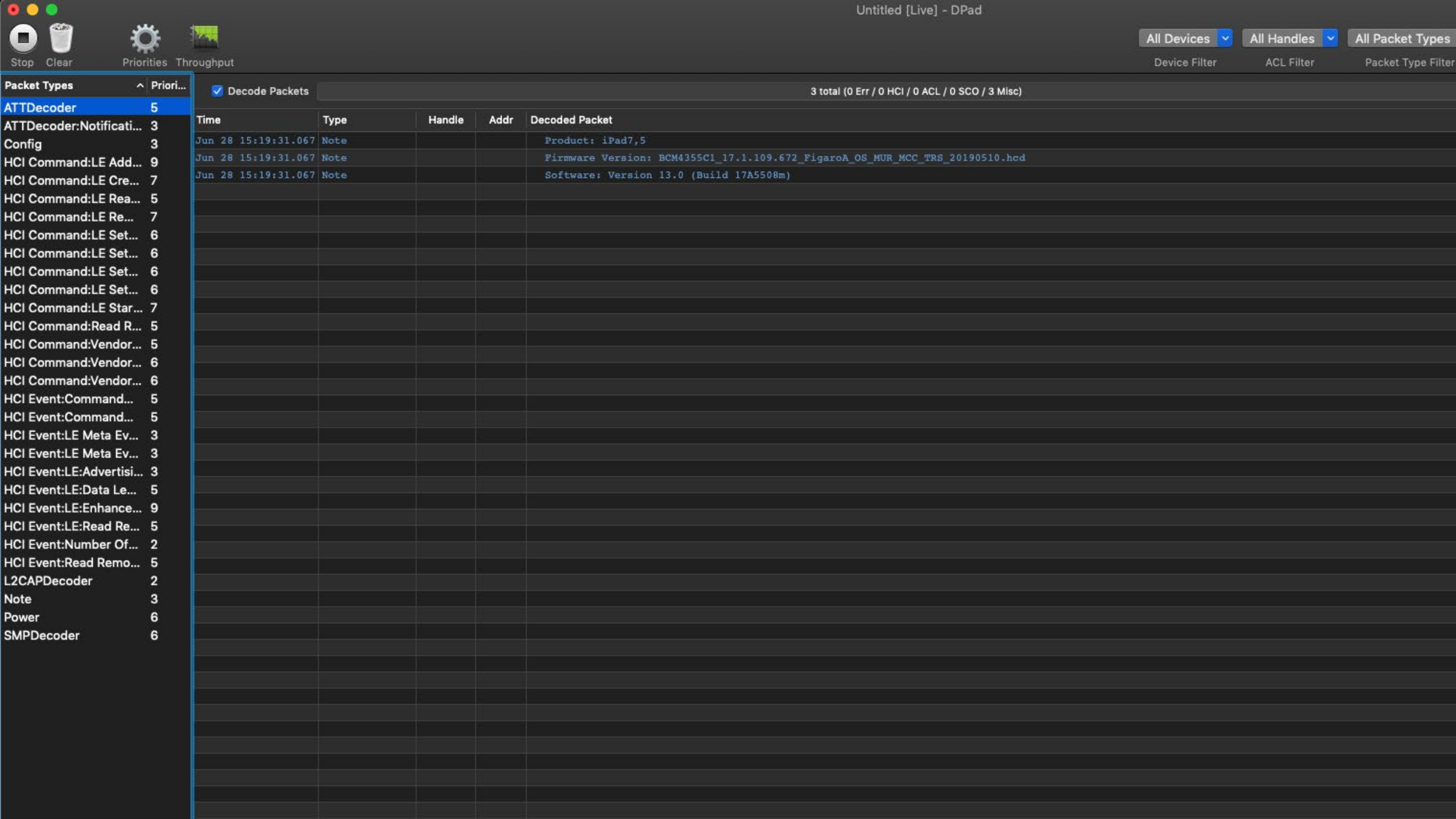
Privacy Updates – iOS 13

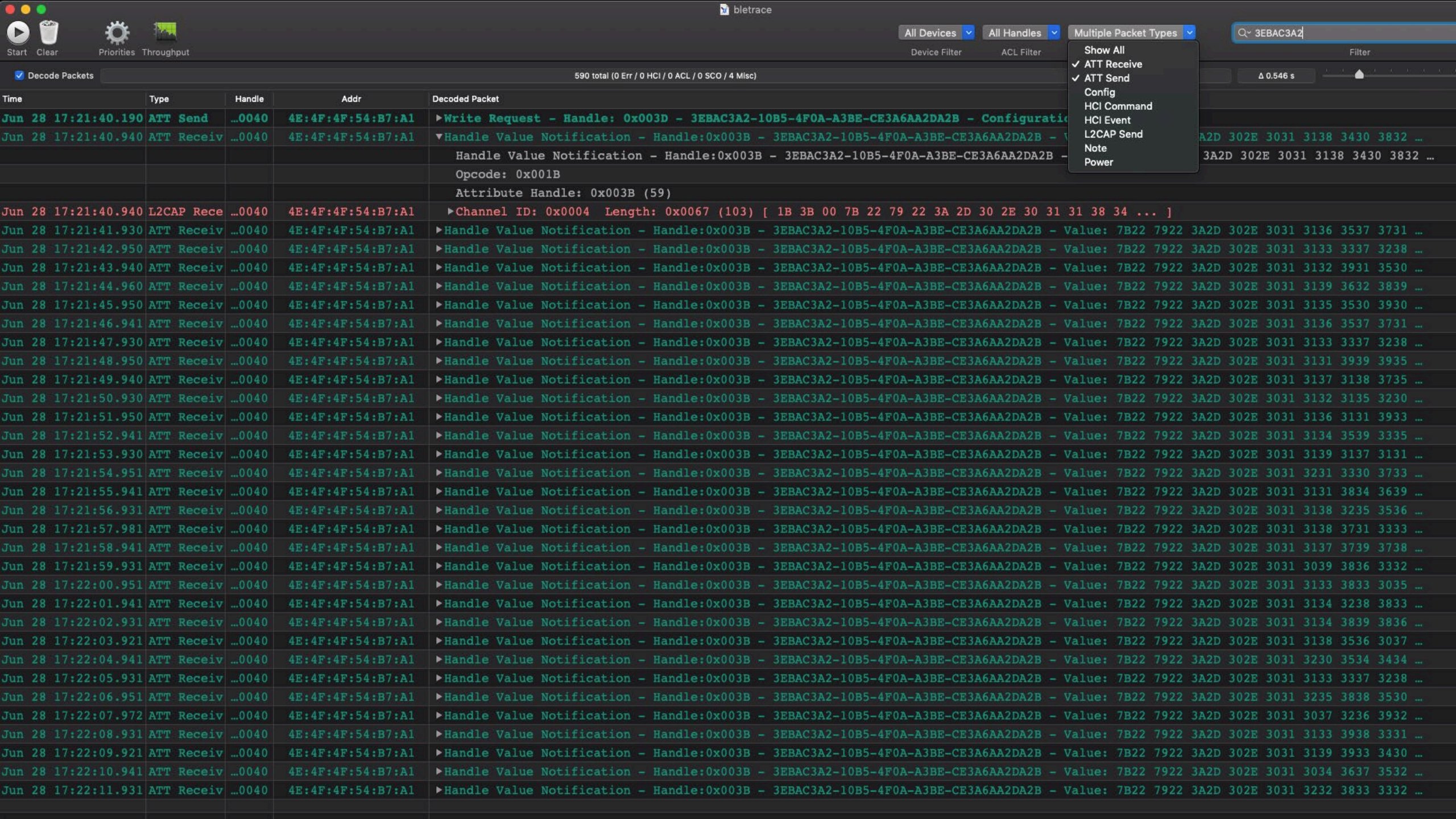
```
func centralManagerDidUpdateState(_ central: CBCentralManager) {
    switch central.state {
    case .unauthorized:
        if central.authorization != .allowedAlways {
            // prompt user for permission
        }
    // handle each case
    ...
    }
}
```

PacketLogger Updates – Live Capture



1. Install iOS 13 developer beta on device
2. Install iOS Bluetooth developer logging profile on device
3. Download “Additional Tools for Xcode”
4. Connect iOS device to Mac
5. Launch PacketLogger
6. Select File -> “New iOS Trace”
7. Indicator will appear on iOS device


Untitled [Live] - DPad
Stop
Clear
Priorities
Throughput
All Devices
All Handles
All Packet Types
Device Filter
ACL Filter
Packet Type Filter
Packet Types
Priori...
ATTDecoder 5
ATTDecoder:Notificati... 3
Config 3
HCI Command:LE Add... 9
HCI Command:LE Cre... 7
HCI Command:LE Rea... 5
HCI Command:LE Re... 7
HCI Command:LE Set... 6
HCI Command:LE Set... 6
HCI Command:LE Set... 6
HCI Command:LE Set... 6
HCI Command:LE Star... 7
HCI Command:Read R... 5
HCI Command:Vendor... 5
HCI Command:Vendor... 6
HCI Command:Vendor... 6
HCI Event:Command... 5
HCI Event:Command... 5
HCI Event:LE Meta Ev... 3
HCI Event:LE Meta Ev... 3
HCI Event:LE:Advertisi... 3
HCI Event:LE:Data Le... 5
HCI Event:LE:Enhance... 9
HCI Event:LE:Read Re... 5
HCI Event:Number Of... 2
HCI Event:Read Remo... 5
L2CAPDecoder 2
Note 3
Power 6
SMPDecoder 6
Decode Packets
3 total (0 Err / 0 HCI / 0 ACL / 0 SCO / 3 Misc)
Time
Type
Handle
Addr
Decoded Packet
Jun 28 15:19:31.067 Note Product: iPad7,5
Jun 28 15:19:31.067 Note Firmware Version: BCM4355C1_17.1.109.672_FigaroA_OS_MUR_MCC_TRS_20190510.hcd
Jun 28 15:19:31.067 Note Software: Version 13.0 (Build 17A5508m)


bletrace
Start
Clear
Priorities
Throughput
All Devices
Device Filter
All Handles
ACL Filter
Multiple Packet Types
Show All
ATT Receive
ATT Send
Config
HCI Command
HCI Event
L2CAP Send
Note
Power
3EBAC3A2
Filter
Decode Packets
590 total (0 Err / 0 HCI / 0 ACL / 0 SCO / 4 Misc)
Δ 0.546 s
Time
Type
Handle
Addr
Decoded Packet
Jun 28 17:21:40.190 ATT Send …0040 4E:4F:4F:54:B7:A1 ▶Write Request - Handle: 0x003D - 3EBAC3A2-10B5-4F0A-A3BE-CE3A6AA2DA2B - Configuratio
Jun 28 17:21:40.940 ATT Receiv …0040 4E:4F:4F:54:B7:A1 ▼Handle Value Notification - Handle:0x003B - 3EBAC3A2-10B5-4F0A-A3BE-CE3A6AA2DA2B - V A2D 302E 3031 3138 3430 3832 …
Handle Value Notification - Handle:0x003B - 3EBAC3A2-10B5-4F0A-A3BE-CE3A6AA2DA2B - 3A2D 302E 3031 3138 3430 3832 …
Opcode: 0x001B
Attribute Handle: 0x003B (59)
Jun 28 17:21:40.940 L2CAP Rece …0040 4E:4F:4F:54:B7:A1 ▶Channel ID: 0x0004 Length: 0x0067 (103) [1B 3B 00 7B 22 79 22 3A 2D 30 2E 30 31 31 38 34 ...]
Jun 28 17:21:41.930 ATT Receiv …0040 4E:4F:4F:54:B7:A1 ▶Handle Value Notification - Handle:0x003B - 3EBAC3A2-10B5-4F0A-A3BE-CE3A6AA2DA2B - Value: 7B22 7922 3A2D 302E 3031 3136 3537 3731 …
Jun 28 17:21:42.950 ATT Receiv …0040 4E:4F:4F:54:B7:A1 ▶Handle Value Notification - Handle:0x003B - 3EBAC3A2-10B5-4F0A-A3BE-CE3A6AA2DA2B - Value: 7B22 7922 3A2D 302E 3031 3133 3337 3238 …
Jun 28 17:21:43.940 ATT Receiv …0040 4E:4F:4F:54:B7:A1 ▶Handle Value Notification - Handle:0x003B - 3EBAC3A2-10B5-4F0A-A3BE-CE3A6AA2DA2B - Value: 7B22 7922 3A2D 302E 3031 3132 3931 3530 …
Jun 28 17:21:44.960 ATT Receiv …0040 4E:4F:4F:54:B7:A1 ▶Handle Value Notification - Handle:0x003B - 3EBAC3A2-10B5-4F0A-A3BE-CE3A6AA2DA2B - Value: 7B22 7922 3A2D 302E 3031 3139 3632 3839 …
Jun 28 17:21:45.950 ATT Receiv …0040 4E:4F:4F:54:B7:A1 ▶Handle Value Notification - Handle:0x003B - 3EBAC3A2-10B5-4F0A-A3BE-CE3A6AA2DA2B - Value: 7B22 7922 3A2D 302E 3031 3135 3530 3930 …
Jun 28 17:21:46.941 ATT Receiv …0040 4E:4F:4F:54:B7:A1 ▶Handle Value Notification - Handle:0x003B - 3EBAC3A2-10B5-4F0A-A3BE-CE3A6AA2DA2B - Value: 7B22 7922 3A2D 302E 3031 3136 3537 3731 …
Jun 28 17:21:47.930 ATT Receiv …0040 4E:4F:4F:54:B7:A1 ▶Handle Value Notification - Handle:0x003B - 3EBAC3A2-10B5-4F0A-A3BE-CE3A6AA2DA2B - Value: 7B22 7922 3A2D 302E 3031 3133 3337 3238 …
Jun 28 17:21:48.950 ATT Receiv …0040 4E:4F:4F:54:B7:A1 ▶Handle Value Notification - Handle:0x003B - 3EBAC3A2-10B5-4F0A-A3BE-CE3A6AA2DA2B - Value: 7B22 7922 3A2D 302E 3031 3131 3939 3935 …
Jun 28 17:21:49.940 ATT Receiv …0040 4E:4F:4F:54:B7:A1 ▶Handle Value Notification - Handle:0x003B - 3EBAC3A2-10B5-4F0A-A3BE-CE3A6AA2DA2B - Value: 7B22 7922 3A2D 302E 3031 3137 3138 3735 …
Jun 28 17:21:50.930 ATT Receiv …0040 4E:4F:4F:54:B7:A1 ▶Handle Value Notification - Handle:0x003B - 3EBAC3A2-10B5-4F0A-A3BE-CE3A6AA2DA2B - Value: 7B22 7922 3A2D 302E 3031 3132 3135 3230 …
Jun 28 17:21:51.950 ATT Receiv …0040 4E:4F:4F:54:B7:A1 ▶Handle Value Notification - Handle:0x003B - 3EBAC3A2-10B5-4F0A-A3BE-CE3A6AA2DA2B - Value: 7B22 7922 3A2D 302E 3031 3136 3131 3933 …
Jun 28 17:21:52.941 ATT Receiv …0040 4E:4F:4F:54:B7:A1 ▶Handle Value Notification - Handle:0x003B - 3EBAC3A2-10B5-4F0A-A3BE-CE3A6AA2DA2B - Value: 7B22 7922 3A2D 302E 3031 3134 3539 3335 …
Jun 28 17:21:53.930 ATT Receiv …0040 4E:4F:4F:54:B7:A1 ▶Handle Value Notification - Handle:0x003B - 3EBAC3A2-10B5-4F0A-A3BE-CE3A6AA2DA2B - Value: 7B22 7922 3A2D 302E 3031 3139 3137 3131 …
Jun 28 17:21:54.951 ATT Receiv …0040 4E:4F:4F:54:B7:A1 ▶Handle Value Notification - Handle:0x003B - 3EBAC3A2-10B5-4F0A-A3BE-CE3A6AA2DA2B - Value: 7B22 7922 3A2D 302E 3031 3231 3330 3733 …
Jun 28 17:21:55.941 ATT Receiv …0040 4E:4F:4F:54:B7:A1 ▶Handle Value Notification - Handle:0x003B - 3EBAC3A2-10B5-4F0A-A3BE-CE3A6AA2DA2B - Value: 7B22 7922 3A2D 302E 3031 3131 3834 3639 …
Jun 28 17:21:56.931 ATT Receiv …0040 4E:4F:4F:54:B7:A1 ▶Handle Value Notification - Handle:0x003B - 3EBAC3A2-10B5-4F0A-A3BE-CE3A6AA2DA2B - Value: 7B22 7922 3A2D 302E 3031 3138 3235 3536 …
Jun 28 17:21:57.981 ATT Receiv …0040 4E:4F:4F:54:B7:A1 ▶Handle Value Notification - Handle:0x003B - 3EBAC3A2-10B5-4F0A-A3BE-CE3A6AA2DA2B - Value: 7B22 7922 3A2D 302E 3031 3138 3731 3333 …
Jun 28 17:21:58.941 ATT Receiv …0040 4E:4F:4F:54:B7:A1 ▶Handle Value Notification - Handle:0x003B - 3EBAC3A2-10B5-4F0A-A3BE-CE3A6AA2DA2B - Value: 7B22 7922 3A2D 302E 3031 3137 3739 3738 …
Jun 28 17:21:59.931 ATT Receiv …0040 4E:4F:4F:54:B7:A1 ▶Handle Value Notification - Handle:0x003B - 3EBAC3A2-10B5-4F0A-A3BE-CE3A6AA2DA2B - Value: 7B22 7922 3A2D 302E 3031 3039 3836 3332 …
Jun 28 17:22:00.951 ATT Receiv …0040 4E:4F:4F:54:B7:A1 ▶Handle Value Notification - Handle:0x003B - 3EBAC3A2-10B5-4F0A-A3BE-CE3A6AA2DA2B - Value: 7B22 7922 3A2D 302E 3031 3133 3833 3035 …
Jun 28 17:22:01.941 ATT Receiv …0040 4E:4F:4F:54:B7:A1 ▶Handle Value Notification - Handle:0x003B - 3EBAC3A2-10B5-4F0A-A3BE-CE3A6AA2DA2B - Value: 7B22 7922 3A2D 302E 3031 3134 3238 3833 …
Jun 28 17:22:02.931 ATT Receiv …0040 4E:4F:4F:54:B7:A1 ▶Handle Value Notification - Handle:0x003B - 3EBAC3A2-10B5-4F0A-A3BE-CE3A6AA2DA2B - Value: 7B22 7922 3A2D 302E 3031 3134 3839 3836 …
Jun 28 17:22:03.921 ATT Receiv …0040 4E:4F:4F:54:B7:A1 ▶Handle Value Notification - Handle:0x003B - 3EBAC3A2-10B5-4F0A-A3BE-CE3A6AA2DA2B - Value: 7B22 7922 3A2D 302E 3031 3138 3536 3037 …
Jun 28 17:22:04.941 ATT Receiv …0040 4E:4F:4F:54:B7:A1 ▶Handle Value Notification - Handle:0x003B - 3EBAC3A2-10B5-4F0A-A3BE-CE3A6AA2DA2B - Value: 7B22 7922 3A2D 302E 3031 3230 3534 3434 …
Jun 28 17:22:05.931 ATT Receiv …0040 4E:4F:4F:54:B7:A1 ▶Handle Value Notification - Handle:0x003B - 3EBAC3A2-10B5-4F0A-A3BE-CE3A6AA2DA2B - Value: 7B22 7922 3A2D 302E 3031 3133 3337 3238 …
Jun 28 17:22:06.951 ATT Receiv …0040 4E:4F:4F:54:B7:A1 ▶Handle Value Notification - Handle:0x003B - 3EBAC3A2-10B5-4F0A-A3BE-CE3A6AA2DA2B - Value: 7B22 7922 3A2D 302E 3031 3235 3838 3530 …
Jun 28 17:22:07.972 ATT Receiv …0040 4E:4F:4F:54:B7:A1 ▶Handle Value Notification - Handle:0x003B - 3EBAC3A2-10B5-4F0A-A3BE-CE3A6AA2DA2B - Value: 7B22 7922 3A2D 302E 3031 3037 3236 3932 …
Jun 28 17:22:08.931 ATT Receiv …0040 4E:4F:4F:54:B7:A1 ▶Handle Value Notification - Handle:0x003B - 3EBAC3A2-10B5-4F0A-A3BE-CE3A6AA2DA2B - Value: 7B22 7922 3A2D 302E 3031 3133 3938 3331 …
Jun 28 17:22:09.921 ATT Receiv …0040 4E:4F:4F:54:B7:A1 ▶Handle Value Notification - Handle:0x003B - 3EBAC3A2-10B5-4F0A-A3BE-CE3A6AA2DA2B - Value: 7B22 7922 3A2D 302E 3031 3139 3933 3430 …
Jun 28 17:22:10.941 ATT Receiv …0040 4E:4F:4F:54:B7:A1 ▶Handle Value Notification - Handle:0x003B - 3EBAC3A2-10B5-4F0A-A3BE-CE3A6AA2DA2B - Value: 7B22 7922 3A2D 302E 3031 3034 3637 3532 …
Jun 28 17:22:11.931 ATT Receiv …0040 4E:4F:4F:54:B7:A1 ▶Handle Value Notification - Handle:0x003B - 3EBAC3A2-10B5-4F0A-A3BE-CE3A6AA2DA2B - Value: 7B22 7922 3A2D 302E 3031 3232 3833 3332 …

Summary

Beacons – small, battery powered device useful for proximity-based applications

L2CAP Channels – direct communication between central and connected peripheral without GATT limitations

iOS 13 Enhancements

- LE 2 Mbps
- Classic Bluetooth connections
- Privacy updates
- Developer tools