KFEI's Terminal App for iOS™

Source code Document

Version 1.0

2023/4/11

Table of Contents

Hierarchical Index

Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Class Index

Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

File Index

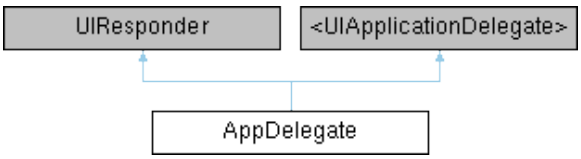
File List

Here is a list of all documented files with brief descriptions:

Class Documentation

AppDelegate Class Reference

Inheritance diagram for AppDelegate:



Properties

UIWindow * **window**

The documentation for this class was generated from the following file:

KFEI's Terminal App/AppDelegate.h

AppDelegate() Category Reference

The documentation for this category was generated from the following file:

KFEI's Terminal App/AppDelegate.m

DeviceData Struct Reference

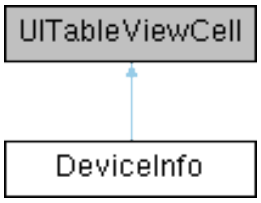
Public Attributes

CBPeripheral * **peripheral**
NSNumber * **rssi**

The documentation for this struct was generated from the following file:
KFEI's Terminal App/Devices.m

DeviceInfo Class Reference

Inheritance diagram for DeviceInfo:



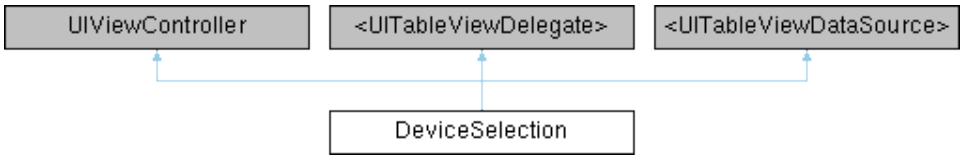
Properties

IBOutlet UIImage * **icon**
IBOutlet UILabel * **name**
IBOutlet UILabel * **rssi**
IBOutlet UILabel * **ident**

The documentation for this class was generated from the following file:
KFEI's Terminal App/CustomCells.h

DeviceSelection Class Reference

Inheritance diagram for DeviceSelection:



Instance Methods

(void) - **refreshView**

Method Documentation

- (void) refreshView

Clears all currently discovered devices and the UITableView and restarts scanning.

The documentation for this class was generated from the following files:

KFEI's Terminal App/Devices.h
KFEI's Terminal App/Devices.m

DeviceSelection() Category Reference

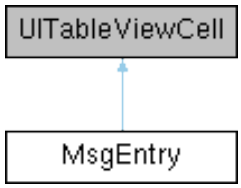
Properties

IBOutlet **DeviceInfo * cbDeviceEntry**
IBOutlet UITableView * **tableView**
NSMutableArray * **deviceList**
NSMutableArray * **rssiList**
CBPeripheral * **connected_peripheral**

The documentation for this category was generated from the following file:
KFEI's Terminal App/Devices.m

MsgEntry Class Reference

Inheritance diagram for MsgEntry:



Properties

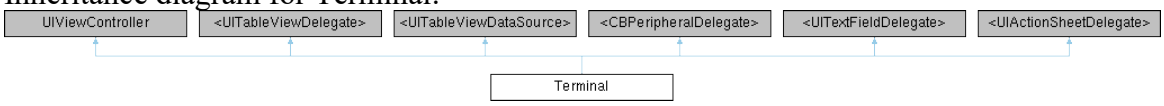
IBOutlet UILabel * **msg**

The documentation for this class was generated from the following file:

KFEI's Terminal App/CustomCells.h

Terminal Class Reference

Inheritance diagram for Terminal:



Instance Methods

(BOOL) - **writeMessageToPeripheral:**
(void) - **printMessageToScreen:characteristic:**
(void) - **clearScreen**
(NSData *) - **ascii2hex:**
(NSData *) - **hex2string:**
(char) - **getValidHex:**

Protected Types

enum **format_t** { **Fmt_Ascii**, **Fmt_Hex** }
enum **eol_t** { **EOL_NONE**, **EOL_CR**, **EOL_LF**, **EOL_CRLF** }

Properties

CBPeripheral * **peripheral**

Member Enumeration Documentation

- (enum) eol_t `[protected]`

End of line character selection enumerators

- (enum) format_t `[protected]`

Numeric format enumerators [Hex, Ascii]

Method Documentation

- (NSData *) ascii2hex: (NSData*) *message*

Converts an ascii sequence to hexadecimal ie. 'AB' -> 0xAB. The conversion will fail if there are any non-hexadecimal characters present. Special charactes (e.g. spaces) will be ignored. Used when transmitting hex data

Parameters

message	ascii formatted input sequence.

Returns

hex formatted sequence, null if invalid.

- (void) clearScreen

Clears the message lists and empties the message table

- (char) getValidHex: (char) *b*

Checks if a byte is a valid hexadecimal character [0-9, A-F].

Parameters

b	byte to check.

Returns

Ascii value of valid character, 0xFE if special character, 0xFF if invalid.

- (NSData *) hex2string: (NSData *) *message*

Converts a hex sequence to string format for display. Used when receiving hex data.

Parameters

message	Input sequence to be converted.

Returns

Ascii formatted sequence

- (void) printMessageToScreen: (NSString *) *message* characteristic: (CBCharacteristic *) *characteristic*

Displays the specified characters on screen. If the message source is the same as the previous, the characters are appended to the end of the string. Otherwise a new table entry is created.

Parameters

message	The message to display on screen @oaram characteristic The message source (TX or RX)

- (BOOL) writeMessageToPeripheral: (NSString *) *string*

Transmits a string of characters to the connected peripheral using the TX characteristic. Characters are transmitted in blocks conforming to maximum BLE packet size. The string will be converted if a non-Ascii format is selected.

Parameters

string	Message input by user

Returns

True on success, False on failure

The documentation for this class was generated from the following files:

KFEI's Terminal App/Terminal.h
KFEI's Terminal App/Terminal.m

Terminal() Category Reference

Properties

IBOutlet UIButton * **Send**
IBOutlet UIBarButtonItem * **OPT**
IBOutlet UIButton * **EOL**
IBOutlet UIButton * **Format**
IBOutlet UIButton * **Clear**
IBOutlet UITextField * **txt_message**
IBOutlet NSLayoutConstraint * **ControlsViewBtmConstraint**
IBOutlet UITableView * **messageTable**
NSMutableArray * **messages**

The documentation for this category was generated from the following file:
KFEI's Terminal App/Terminal.m

BeaconLog Class Reference [Swift]

Class for iBeacon formatted data.

Instance Methods

init(uuid: UUID, minor: NSNumber, major: NSNumber, rssi: Int)

Properties

var uuid: UUID
var minor: NSNumber
var major: NSNumber
var rssi: Int

Method Documentaion

init(uuid: UUID, minor: NSNumber, major: NSNumber, rssi: Int)

Initialize iBeacon data

The documentation for this category was generated from the following file:

KFEI's Terminal App/Beacons.swift

BeaconCell: UITableViewCell Class Reference [Swift]

Setup iBeacon table cell.

Instance Methods

override func awakeFromNib()

Properties

@IBOutlet weak var header: UILabel!
@IBOutlet weak var minor: UILabel!
@IBOutlet weak var major: UILabel!
@IBOutlet weak var rssi: UILabel!

The documentation for this category was generated from the following file:

KFEI's Terminal App/Beacons.swift

Beacons: UIViewController, UITextViewDelegate, UITableViewDelegate, UITableViewDataSource, CLLocationManagerDelegate Class Reference [Swift]

Detecting iBeacon by using LacationManager.

Instance Methods

- override func viewDidLoad()
- func locationManager(_ manager: CLLocationManager, didChangeAuthorization status: CLAuthorizationStatus)
- func startScanning(uuidString: String)
- func stopScaning()
- func locationManager(_ manager: CLLocationManager, didRangeBeacons beacons: [CLBeacon], in region: CLBeaconRegion)
- @objc func refreshTable(notification: NSNotification)
- func clearTable()
- func newBeaconData(new: BeaconLog)
- func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int
- func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell
- func scrollToBottom()
- override func viewWillAppear(_ animated: Bool)
- @objc func proximityUuidButtonTapped()

Properties

@IBOutlet weak var b_navbar: UINavigationBar!
@IBOutlet weak var beaconTable: UITableView!
var beaconData: Array<BeaconLog>!
var locationManager: CLLocationManager!
var beaconRegion: CLBeaconRegion!
var uuidString: String!
var isMonitoring: Bool!

Method Documentation

override func viewDidLoad()

Called when this view is loaded into the application. Setup location manager.

func locationManager(_ manager: CLLocationManager, didChangeAuthorization status: CLAuthorizationStatus)

Check authorization and start iBeacon scanning.

func startScanning(uuidString: String)

Start scanning for iBeacon.

Parameters

uuidString	UUID string for detecting iBeacon.

func stopScaning()

Stop iBeacon scanning

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

Set detected iBeacon to table.

@objc func refreshTable(notification: NSNotification)

Called on reflashing table

func clearTable()

Clear iBeacon data and table.

func newBeaconData(new: BeaconLog)

Add or update iBeacon to beacon data.

func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int

Return the number of beacon data.

func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell

Return beacon data's parameters.

func scrollToBottom()

Function for iBeacon table scrolled to bottom.

override func viewWillAppear(_ animated: Bool)

Redraw beacon table.

@objc func proximityUuidButtonTapped()

Callback for UUID button pushed. Create dialog for inputing UUID, and start iBeacon scanning.

The documentation for this category was generated from the following file:

KFEI's Terminal App/Beacons.swift

BeaconData Structure Reference [Swift]

Beacon data struct.

Properties

let name: String
let time: Date
let data: Data
let rssi: Int

The documentation for this category was generated from the following file:
KFEI's Terminal App/TabController.swift

DeviceData: NSObject Class Reference [Swift]

Peripheral data struct.

Instance Methods

init(peripheral: CBPeripheral, rssi: NSNumber)

Properties

@objc var peripheral: CBPeripheral
@objc var rssi: NSNumber

The documentation for this category was generated from the following file:
KFEI's Terminal App/TabController.swift

KAGAFEI Class Reference [Swift]

Definition for UUID.

Properties

static let SERVICE_UUID
static let RXCHAR_UUID
static let TXCHAR_UUID
static let TXCHARWITHRES_UUID

The documentation for this category was generated from the following file:
KFEI's Terminal App/TabController.swift

TabController: CBCentralManagerDelegate Extension Reference [Swift]

Bluetooth Central Delegate for Scanning

Instance Methods

- func centralManagerDidUpdateState(_ central: CBCentralManager)
- func centralManager(_ central: CBCentralManager, didDiscover peripheral: CBPeripheral, advertisementData: [String : Any], rssi RSSI: NSNumber)
- func centralManager(_ central: CBCentralManager, didConnect peripheral: CBPeripheral)
- func centralManager(_ central: CBCentralManager, didDisconnectPeripheral peripheral: CBPeripheral, error: Error?)
- func centralManager(_ central: CBCentralManager, didFailToConnect peripheral: CBPeripheral, error: Error?)
- func centralManager(_ central: CBCentralManager, willRestoreState dict: [String : Any])

Method Documentation

func centralManager(_ central: CBCentralManager, didDiscover peripheral: CBPeripheral, advertisementData: [String : Any], rssi RSSI: NSNumber)

Called when device is discovered. If device is connectable, added to device list.

func centralManager(_ central: CBCentralManager, didConnect peripheral: CBPeripheral)

Called when connected with periferal . Start service discovery.

func centralManager(_ central: CBCentralManager, didDisconnectPeripheral peripheral: CBPeripheral, error: Error?)

Called when disconnected.

func centralManager(_ central: CBCentralManager, didFailToConnect peripheral: CBPeripheral, error: Error?)

Called when connection failed.

func centralManager(_ central: CBCentralManager, willRestoreState dict: [String : Any])

Called when state restored.

The documentation for this category was generated from the following file:

KFEI's Terminal App/TabController.swift

TabController: CBPeripheralDelegate Extension Reference [Swift]

Peripheral Delegate

Instance Methods

- func peripheral(_ peripheral: CBPeripheral, didDiscoverServices error: Error?)
- func peripheral(_ peripheral: CBPeripheral, didDiscoverCharacteristicsFor service: CBService, error: Error?)

Method Documentation

func peripheral(_ peripheral: CBPeripheral, didDiscoverServices error: Error?)

Called when device's service UUID is discovered. If service UUID matches with specified, start characteristic discovery.

func peripheral(_ peripheral: CBPeripheral, didDiscoverCharacteristicsFor service: CBService, error: Error?)

Called when device's characteristics UUID is discovered.

The documentation for this category was generated from the following file:

KFEI's Terminal App/TabController.swift

TabController: UITabBarController Class Reference [Swift]

TabController

Instance Methods

- override func viewDidLoad()
- @objc func disconnectPeripheral(notification: NSNotification)
- @objc func stopScanning(notification: NSNotification)
- @objc func startAdvertisement(notification: NSNotification)
- func getDevices() -> Array<CBPeripheral>
- func getRssi() -> Array<NSNumber>
- @objc func connectToDevice(notification: NSNotification)

Properties

var centralManager: CBCentralManager!
var deviceList: Dictionary<String,DeviceData>!
var rssiList: Array<Int>!
var beaconList: Array<BeaconData>!
var services: Array<CBUUID>!
var connected_peripheral: CBPeripheral!

Method Documentation

override func viewDidLoad()

Called when this controller is loaded into the application. Setup BLE Central manager.

@objc func disconnectPeripheral(notification: NSNotification)

Start disconnection.

@objc func stopScanning(notification: NSNotification)

Stop BLE scanning.

@objc func startAdvertisement(notification: NSNotification)

Start scanning for detecting BLE device.

func getDevices() -> Array<CBPeripheral>

Get device list.

func getRssi() -> Array<NSNumber>

Get rssi list.

@objc func connectToDevice(notification: NSNotification)

Start connection.

The documentation for this category was generated from the following file:

KFEI's Terminal App/TabController.swift

File Documentation

AppDelegate.h

```
1 //
2 //  AppDelegate.h
3 //  kfeis-terminal-app
4 //
5 //  Copyright © 2023 KAGA FEI Co., Ltd. All rights reserved.
6 //
7
8 #import <UIKit/UIKit.h>
9
10 @interface AppDelegate : UIResponder <UIApplicationDelegate>
11
12 @property (strong, nonatomic) UIWindow *window;
13
14
15 @end
16
```

CustomCells.h

```
1 //
2 //  CustomCells.h
3 //  kfeis-terminal-app
4 //
5 //  Copyright © 2023 KAGA FEI Co., Ltd. All rights reserved.
6 //
7
8 #ifndef __CustomCells_H_
9 #define __CustomCells_H_
10
11 #import <UIKit/UIKit.h>
12
13 @interface DeviceInfo : UITableViewCell
14
15 @property (nonatomic, weak) IBOutlet UIImage *icon;
16 @property (nonatomic, weak) IBOutlet UILabel *name;
17 @property (nonatomic, weak) IBOutlet UILabel *rssi;
18 @property (nonatomic, weak) IBOutlet UILabel *ident;
19
20 @end
21
22 @interface MsgEntry : UITableViewCell
23
24 @property (nonatomic, weak) IBOutlet UILabel *msg;
25
26 @end
27
28 #endif
```

Devices.h

```
1 //
2 //  Devices.m
3 //  kfeis-terminal-app
4 //
5 //  Copyright © 2023 KAGA FEI Co., Ltd. All rights reserved.
6 //
7 #ifndef __ViewController_H_
8 #define __ViewController_H_
9
10 #import <UIKit/UIKit.h>
11 @import CoreBluetooth;
12
13 #define KAGAFEI_SERVICE_UUID @"442F1570-8A00-9A28-CBE1-E1D4212D53EB"
14 #define KAGAFEI_RXCHAR_UUID @"442F1571-8A00-9A28-CBE1-E1D4212D53EB"
15 #define KAGAFEI_TXCHAR_UUID @"442F1572-8A00-9A28-CBE1-E1D4212D53EB"
16 #define KAGAFEI_TXCHARWITHRES_UUID @"442F1574-8A00-9A28-CBE1-E1D4212D53EB"
17
18 @interface DeviceSelection : UIViewController <UITableViewDelegate,
UITableViewDataSource>
19
20
21 - (void) refreshView;
22
23 @end
24
25 #endif
```

KFEI's Terminal App-Bridging-Header.h

```
1 //
2 //  Use this file to import your target's public headers that you would like to expose
to Swift.
3 //
4
```

Terminal.h

```
1 //
2 //  Terminal.h
3 //  kfeis-terminal-app
4 //
5 //  Copyright © 2023 KAGA FEI Co., Ltd. All rights reserved.
6 //
7
8 #ifndef __Terminal_H_
9 #define __Terminal_H_
10
11 #import <UIKit/UIKit.h>
12 @import CoreBluetooth;
13
14 @interface Terminal : UIViewController <UITableViewDelegate, UITableViewDataSource,
CBPeripheralDelegate, UITextFieldDelegate, UIActionSheetDelegate>
15
16 @property (strong, nonatomic) CBPeripheral *peripheral;
17
21 typedef enum {
22    Fmt_Ascii,
23    Fmt_Hex,
24 } format t;
25
29 typedef enum {
30    EOL_NONE,
31    EOL_CR,
32    EOL LF,
33    EOL_CRLF,
34 } eol_t;
35
36 // Macros to return string representation
37 #define format2str(enum) (NSString*)[@[@"Ascii",@"Hex"] objectAtIndex:enum]
38 #define eol2str(enum) [@[@"None",@"CR",@"LF",@"CRLF"] objectAtIndex:enum]
39
40 - (BOOL)writeMessageToPeripheral:(NSString *)string;
41 - (void)printMessageToScreen:(NSString *)message characteristic:(CBCharacteristic
*)characteristic;
42 - (void)clearScreen;
43 - (NSData *) ascii2hex: (NSData*) message;
44 - (NSData *) hex2string:(NSData *)message;
45 - (char) getValidHex:(char) b;
46 @end
47
48 #end
```