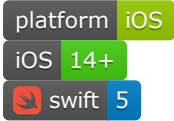


# SpeechMike Ambient

---



The SpeechMike Ambient is engineered specifically to capture high-fidelity, single and multi-speaker audio in real-world environments—perfect for AI transcription, conversational AI and ambient scribe scenarios in clinical, and virtual assistant functions. Its superior input quality significantly enhances AI model accuracy, making it ideal for generating clinical notes, transcriptions, and real-time documentation More information is available [here](#).

## Bluetooth SDK library

---

The SpeechMike Ambient iOS SDK provides a comprehensive solution for integrating Philips SpeechMike Ambient devices with iOS applications via Bluetooth Low Energy (BLE). This SDK enables real-time audio streaming, device management, and advanced audio processing capabilities for professional dictation and voice recording applications. The SDK itself is distributed as a prebuilt `.xcframework` library (written in Swift).

### Note:

The SDK source code is maintained in a private repository.

This public repository is intended to showcase integration, usage, and behavior of the SDK through a demo application.

### Project Structure

```
root
├── sma-ios-sdk-test-swiftui           # Demo iOS application
├── SpeechMikeAmbientSDK.xcframework # Library file
└── README.md
```

### SDK Overview


- **Language:** Swift
- **Distribution:** `.xcframework` iOS library
- **Communication:** Bluetooth Low Energy (BLE)

- **Target Device:** Philips SpeechMike Ambient
- 

The SDK abstracts:

- BLE scanning and connection
- GATT services and characteristics
- Device-specific commands and events

This allows client applications to focus only on **business logic**, not low-level BLE handling.

 SDK Integration (XCFramework)


The SDK is included as a local `.xcframework` dependency. Follow these steps to include it in your project:

---

## 1 Download the SDK (XCFramework)

The SDK is distributed as a zipped XCFramework in GitHub [Releases](#).

- Open this repository's Releases tab
- Download the asset ZIP (e.g. SpeechMikeAmbientSDK.zip)
- Unzip it to get: SpeechMikeAmbientSDK.xcframework

 This and the following paragraphs apply only to your private applications. This demo application has xcframework and BLE permissions already included.

## 2 Add SpeechMikeAmbientSDK.xcframework to your app

- Open your iOS app project in Xcode
- Drag `SpeechMikeAmbientSDK.xcframework` into the Project Navigator (e.g. into a Frameworks/ group)
- In the dialog, enable:

1. Copy items if needed
2. Add to targets (select the app target)

- Verify:

`Target` → `General` → `Frameworks, Libraries, and Embedded Content`  
`SpeechMikeAmbientSDK.xcframework` should be set to `Embed & Sign`

## 3 Info.plist Permissions


Add the following keys to Info.plist:

`NSBluetoothAlwaysUsageDescription`

Example: "Bluetooth is used to connect to a SpeechMike Ambient device."

## Permissions & Requirements










Component	Version
Xcode	16+
iOS	14+
Swift	5+

 The app must be run on a real iOS device. The iOS Simulator does not provide reliable CoreBluetooth support.

## Demo App

The repository includes a fully functional demo application showcasing all SDK features. You can try the demo app by cloning this repository.

## Features Demonstrated

-  BLE device scanning with filtering
-  Device connection and disconnection
-  Real-time audio recording
-  Audio mode switching
-  LED control
-  Power event monitoring
-  Out of reach event monitoring
-  PCM to WAV conversion
-  File management and playback

## Running the Demo

1. Clone the repository
2. Open in Xcode
3. Build and run the app
4. Grant required permissions
5. Scan for your SpeechMike Ambient device

## 6. Connect and start recording!

### Demo App Structure

```
root/
├── sma-ios-sdk-test-swiftui
│   ├── sma_ios_sdk_test_swiftuiApp.swift    # Main App class
│   ├── BLEViewModel.swift                  # ViewModel that communicates
with SDK and UI
│   ├── ConnectView.swift                  # First screen - handles BLE
connection and scanning
│   └── DeviceView.swift                    # Second screen - interaction
with SMA device
```

---



## Documentation

### Core Classes

#### SpeechMikeAmbientBleManager

Main SDK class - Singleton pattern for managing device operations.

```
// Get instance
let smaBleManager = SpeechMikeAmbientBleManager.shared

// Conform to SMABleManagerDelegate protocol
init() {
    smaBleManager.delegate = self
}

// Main methods / Public API
smaBleManager.scanForDevices(nameFilter: String? = nil, onDeviceFound: @e
smaBleManager.stopScan()
smaBleManager.connectToDevice(to device: BleDevice)
smaBleManager.disconnectDevice()
smaBleManager.startAudioStreaming()
smaBleManager.stopAudioStreaming()
smaBleManager.setLed(greenState: LedState, redState: LedState)
smaBleManager.setAudioMode(mode: AudioMode)
smaBleManager.getFirmwareVersion()
```

## SMABleManagerDelegate

Interface for receiving SDK events.

```
func smaManager(didChangeState state: SMAState)
func smaManager(didChangeRecordingState isRecording: Bool)
func smaManager(didChangeAudioMode mode: AudioMode)
func smaManager(didReceiveButtonEvent buttons: [ButtonType])
func smaManager(log line: String)
func smaManager(didReceiveAudioFrame frame: AudioFrame)
func smaManager(didFinishRecordingWav wavData: Data)
func smaManagerDidDetectOutOfReach()
```

### Limitations

- SDK source code is not included in this repository
- Demo app focuses on core BLE functionality only
- UI is intentionally minimal

## Troubleshooting

### Common Issues

#### Device Not Found During Scan

- Ensure Bluetooth is enabled
- Verify BLUETOOTH permission
- Make sure the device is powered on and not connected to another device

#### Connection Fails

- Check BLUETOOTH permission
- Ensure device is within range (typically 10 meters)
- Try resetting the SpeechMike Ambient device
- Verify the device is not in deep sleep mode

#### No Audio Frames Received

- Verify `startAudioStreaming()` was called after connection
- Check that the recording button on the device was pressed
- Ensure the listener is set before starting streaming
- Verify the device has sufficient battery

## Audio Quality Issues

- Experiment with different `AudioMode` settings
  - Check sample rate configuration
  - Ensure proper PCM to WAV conversion parameters
  - Monitor connection quality
- 



## Technical Specifications

### BLE Characteristics

Service	Characteristic	Type	Description
Audio Service	Audio Stream	Notify	Real-time audio data
Control Service	Audio Mode	Read/Write	Audio mode configuration
Control Service	LED Control	Write	LED state control
Status Service	Power Events	Notify	Device power state changes
Status Service	Out Of Reach Events	Notify	Device connectivity state changes

### Audio Specifications

Parameter	Value
Sample Rates	8 kHz, 16 kHz
Bit Depth	16-bit
Frame Size	56 bytes (mono), 112 bytes (stereo)

---



## Acknowledgments

- Designed for [Philips SpeechMike Ambient](#) devices
-