```
private func getFilePath() -> NSURL {
        let fileName = "\(self.user.task)-\(recordCount).m4a"
        let documentsPath = NSSearchPathForDirectoriesInDomains(.DocumentDirectory,
.UserDomainMask, true).first! as String
        let filePath = (documentsPath as
NSString).stringByAppendingPathComponent(fileName)
        return NSURL(fileURLWithPath: filePath)
    }

    //------------------------------------------------------------------------------
    // MARK: EZMicrophoneDelegate
    //------------------------------------------------------------------------------
    func microphone(microphone: EZMicrophone!, changedPlayingState isPlaying: Bool)
{
        //
    }

    func microphone(microphone: EZMicrophone!, hasAudioReceived buffer:
UnsafeMutablePointer<UnsafeMutablePointer<Float>>, withBufferSize bufferSize:
UInt32, withNumberOfChannels numberOfChannels: UInt32) {
        dispatch_async(dispatch_get_main_queue(), { () -> Void in
            self.inputVoiceView.voicePlotGL.updateBuffer(buffer[0], withBufferSize:
bufferSize)
        });
    }

    func microphone(microphone: EZMicrophone!, hasBufferList bufferList:
UnsafeMutablePointer<AudioBufferList>, withBufferSize bufferSize: UInt32,
withNumberOfChannels numberOfChannels: UInt32) {
        if isRecording {
            recorder.appendDataFromBufferList(bufferList, withBufferSize:
bufferSize)
        }
    }

    //------------------------------------------------------------------------------
    // MARK: EZRecorderDelegate
    //------------------------------------------------------------------------------
    func recorderDidClose(recorder: EZRecorder!) {
        recorder.delegate = nil
    }

    func recorderUpdatedCurrentTime(recorder: EZRecorder!) {
        let currentTime = recorder.formattedCurrentTime

        dispatch_async(dispatch_get_main_queue(), { () -> Void in
            self.inputVoiceView.timeLabel.text = currentTime
        });
    }
}
```