```
private func appendResult() {
        let inputSecond = getCalclatedVoiceInputTime()

        let now = NSDate()
        let formatter = NSDateFormatter()
        formatter.dateFromString("YY-MM-DD HH:mm:ss")
        let dateString = formatter.stringFromDate(now)

        let dataString = dateString.stringByAppendingFormat("%.2fmin, UX: %.2f%%,
SliderInput: %.2fsec, VoiceInput: %.2fsec.\n",
                                                            self.plotMinutes,
                                                            uxplotView.slider.value,
                                                            self.operationSecond,
                                                            inputSecond)

        self.resultMeasurement.appendString(dataString)
        print(self.resultMeasurement)
    }

    private func getCalclatedVoiceInputTime() -> Float {
        guard let startTime = voiceInputStartTime else {
            return 0.0
        }

        let now = NSDate()
        let elapdedTime = now.timeIntervalSinceDate(startTime)
        let hour = Int(elapdedTime / 3600)
        let minutes = Int((elapdedTime - Double(hour)) / 60)
        let second = elapdedTime - (Double(hour * 3600 + minutes * 60))
        return Float(second)
    }

    private func clearInputVoiceView() {
        microphone.stopFetchingAudio()
        inputVoiceView.voicePlotGL.clear()
        inputVoiceView.recordButton.setImage(UIImage(named:"record"), forState:
.Normal)
        inputVoiceView.pulsator.stop()
        isRecording = false

        if recorder != nil && recorder.delegate != nil {
            recorder.closeAudioFile()
        }
    }

    private func activeInputVoiceView() {
        inputVoiceView.explainLabel.text = "中央のボタンをタップすると\n音声入力が完了しま
す"
        microphone.startFetchingAudio()
        inputVoiceView.recordButton.setImage(UIImage(named:"record-active"),
forState: .Normal)
        inputVoiceView.pulsator.start()
        recorder = EZRecorder(URL: getFilePath(), clientFormat:
microphone.audioStreamBasicDescription(), fileType: .M4A, delegate: self)
        isRecording = true
    }
```