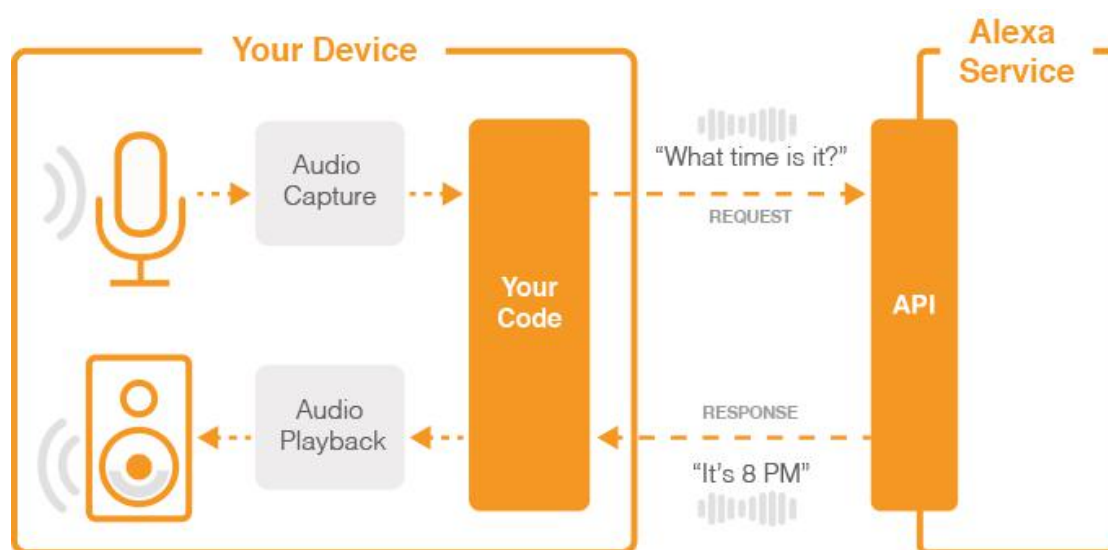
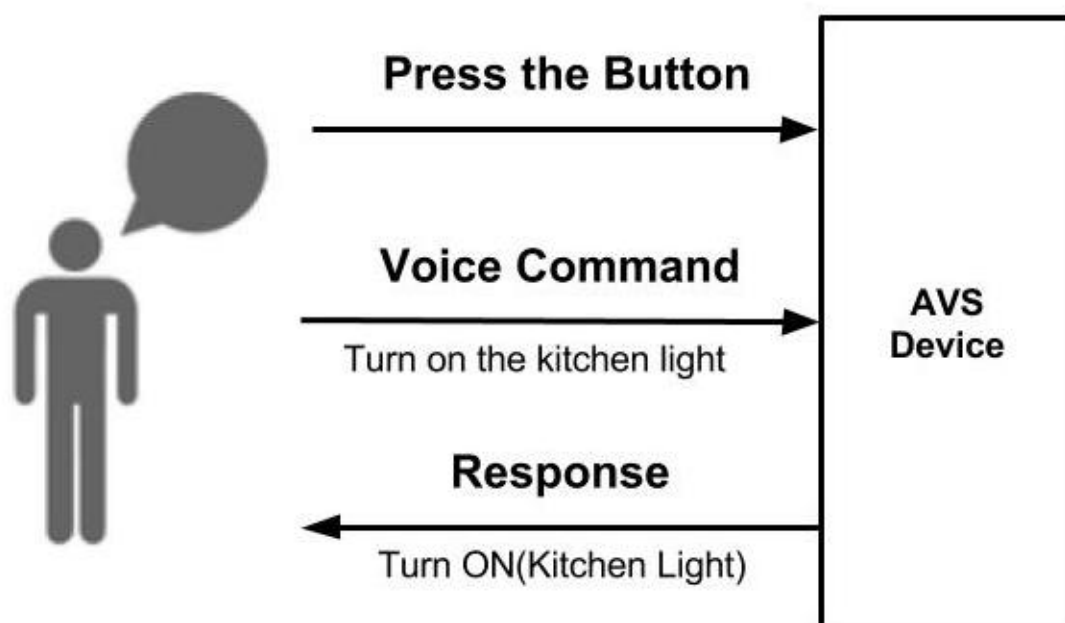


RAK WisCore Alexa 语音开发开源套件

自从 Amazon Echo 于 2016 年 10 月份吸引各方目光以来，国内外各家巨头突然间意识到智能音响的战略价值，纷纷涌入此一市场，当然主要的目的，在于抢占智慧家居入口，毕竟语音操作是人类最直观的方式，并透过结合云端服务商与生态圈的方式，协助巨头的智慧助理落地，打造智慧家庭里的 AI 核心。Amazon Echo 的语音助理服务，主要是基于 Alexa Voice Service (AVS) 为基础，其基本架构与使用情境如下：

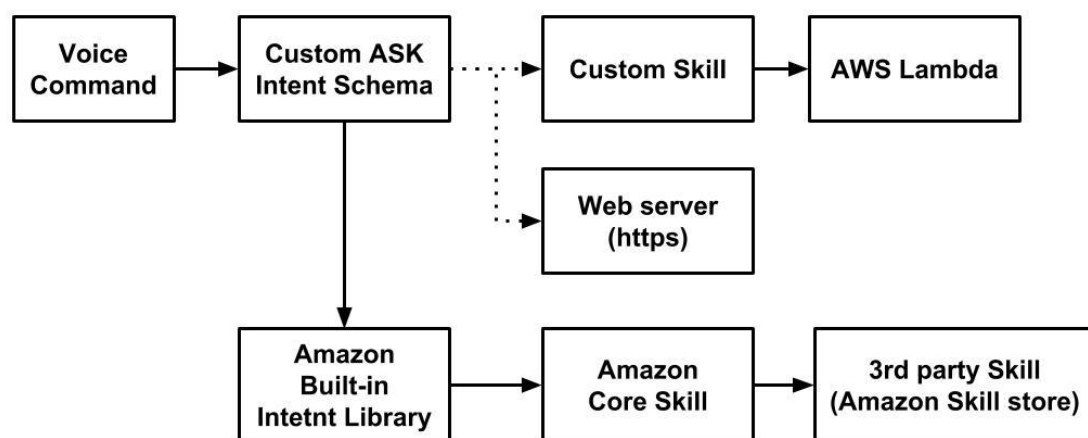


(From <https://developer.amazon.com/public/solutions/alexa/alexa-voice-service>)

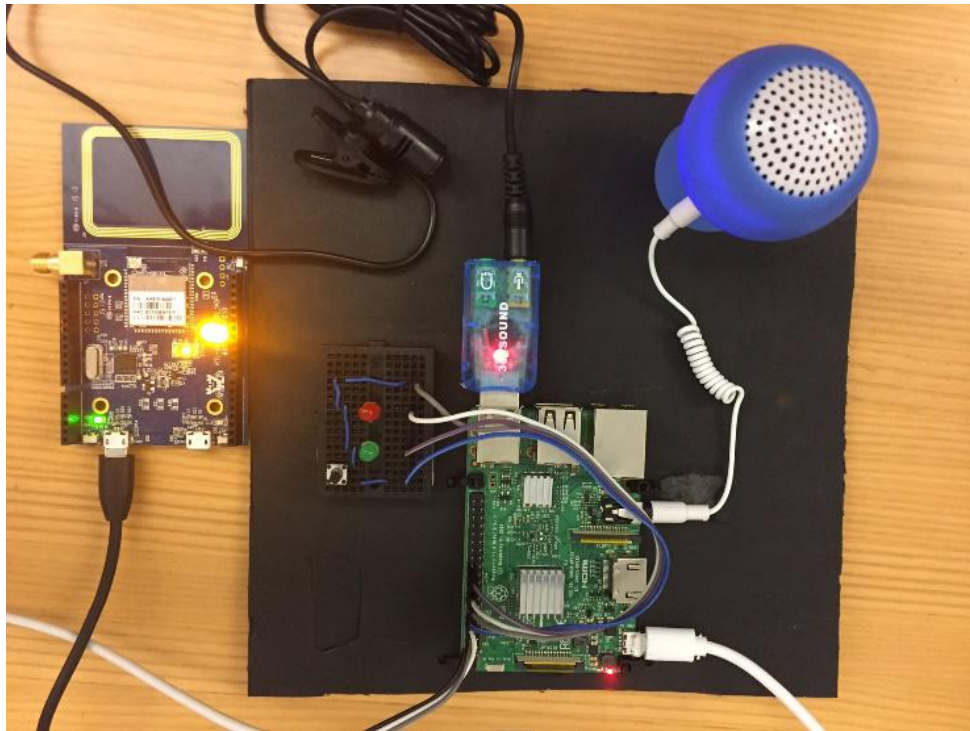


简单的来说，Amazon Alexa Service(AVS)主要对用户输入的语音进行处理，透过 ASR((automatic speech recognition))和 NLP(Natural language processing)，将其转换

成使用者的意图(Intent)，并传输到对应的 Alexa Skill Kit(ASK)，以便进行后续的动作(ex.以语音回复用户或打开厨房的电灯)，透过下图的流程，让用户轻松地透过语音来查询所需的信息，或是控制智能家庭的各种装置，亚马逊对 Alexa Skill 的开放分为两个方面：一个是允许第三方服务提供商在 Alexa 中加入更多的技能，这些技能完全是基于语音互动并直接面向消费者的，比如说透过 Alexa 让用户使用 Uber 提供的叫车服务。二是针对应用程序开发者，Amazon 提供了一个名为 Alexa Skills Kit（简称 ASK）的开发包，类似 Amdroid APP 生态圈的方式来，让开发者可以将自行开发的 CustcomASK 上架到 ASK store(https://www.amazon.com/alexa-skills/b/ref=topnav_storetab_a2s?ie=UTF8&node=13727921011)上，让一般消费者可以同安装 APP 的方式去安装 ASK，安装 ASK 后消费者的 Echo 就会听懂相对应的语音指令，来进行对应的语音操作。



同时之间，Amazon 释出了基于 Raspberry Pi 的语音助理开发工具包(<https://github.com/alexa/alexa-avs-sample-app>)，让 Maker 可以透过透过树梅派与 USB 声卡，来实作 DIY 版的 Amazon Echo，并实作相关 Alexa Skill Kit(ASK)来触发 IOT 相关的装置举例来说，下图为笔者实作用来控制 AWS IOT 的周末小项目，可以透过语音指令进行开关灯的操作。

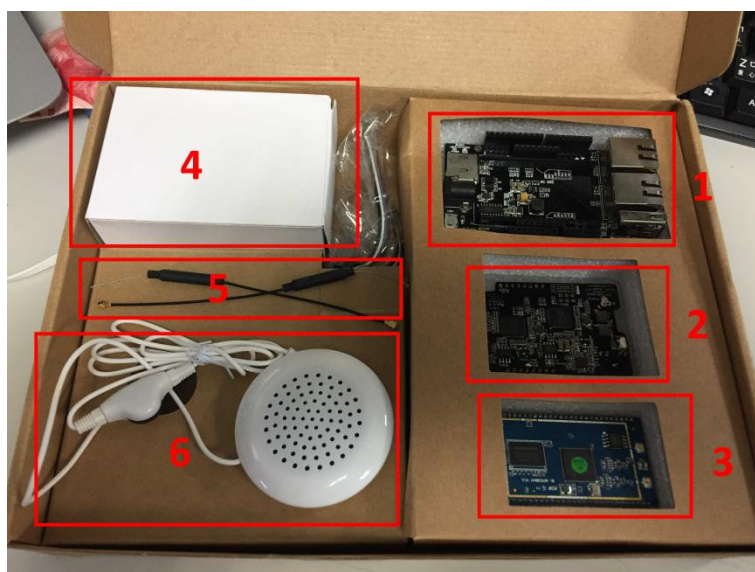


但 Alexa AVS Sample APP 主要目的为提供开发者展示 Alexa Service 的认证与操作细节，所以跟实际上 Amazon Echo 使用上仍有一段差距：

1. 无法开机自动启动，使用上仍需透过 UI 与刘染器进行操作。
2. 需透过外部按钮或 UI APP 上的按钮才能说 Alexa~，无法 Hand Free 操作。
3. 麦克风为一般电容是模拟麦克风，无法 Far-Field 操作。

基于以上的缺点，一般 Maker 不易将 Alexa Voice Service，透过创意整合到自己的物联网项目中，基于 Maker 这样的需求，RAK 推出的一款具备 Amazon Alexa Voice Service 接入的全新开源硬件产品 WisCore，默认内建完整的 AVS 服务功能与 OpenWRT 操作系统，让 Maker 可以专注在应用层面的设计，并快速导入 Alexa 语音助理于物联网的应用之中。今天有幸收到 WisCore 测试机，马上打开来试试。





RAK WisCore 语音开发工具包内含：

1. WisCore 底板
2. WisCore 语音板(MicroSemiZL38062)
3. WisCore CPU 主板 (MT7682A，类似大家熟悉的 Linkit 7688)
4. 变压器 5V/2A
5. Wi-Fi 天线
6. 喇叭

RAK WisCore 语音开发工具包规格：

组件	规格
CPU	MT7628A(MIPS24KEc CPU with 580MHz)
Codec	MicroSemi ZL38062，音频输入/输出 MicroSemi ZL38067，支持“Alexa”关键词唤醒
Microphone	Digital Mic x 2
Power Supply	DC 5V/2A
Wireless	Wi-Fi: 802.11 a/b/g/n, 2.4GHz, 2x2 MIMO, 300Mbps Bluetooth: UART 预留外接模块, SDK 支持 Zigbee: UART 预留外接模块, SDK 支持 Z-Wave: UART 预留外接模块, SDK 支持
Memory	Flash: 16MB DDR: 128MB

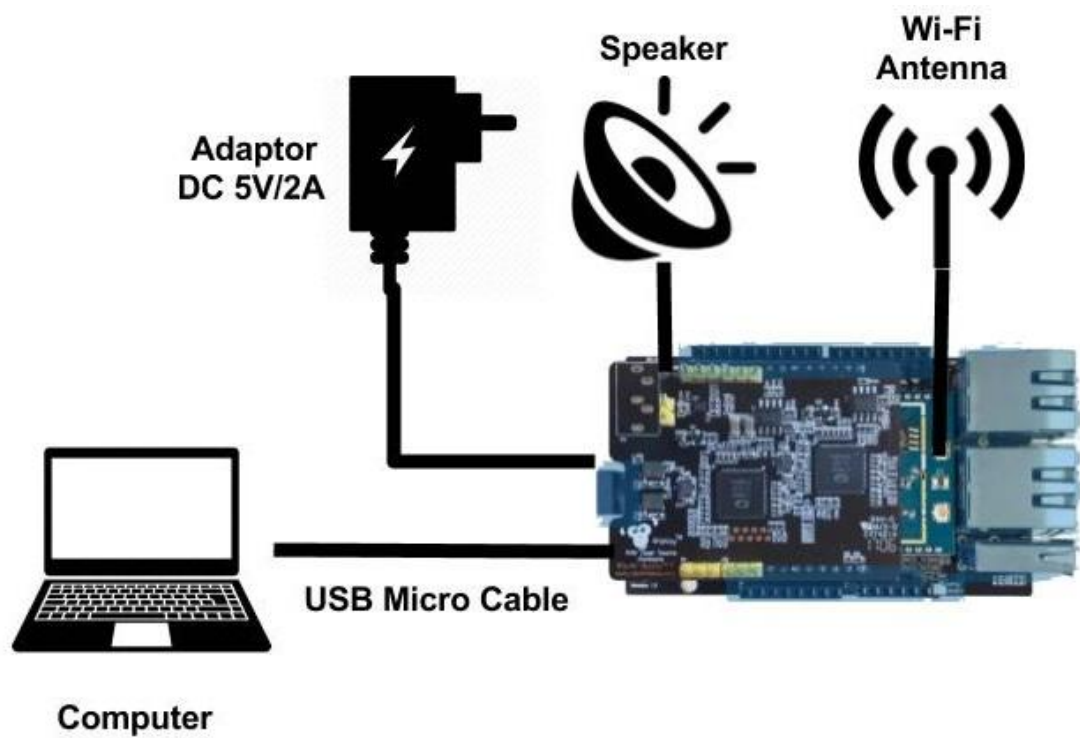
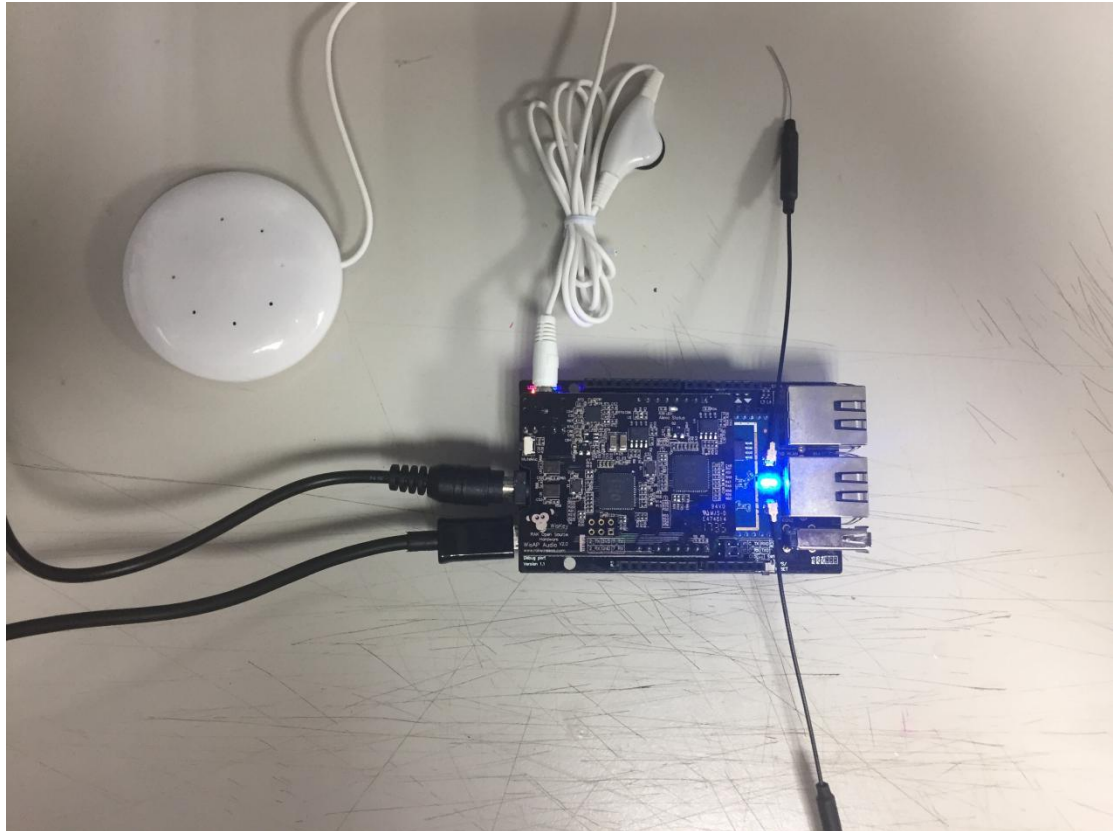
WisCore 开发工具包包括一个 WisCore 核心模块，基于 MTK7628 芯片组，具备 2 组 UART，还包括 Microsemi 的 Voice Processor，支持 ASR 功能与远距离语音唤醒，可通过「Alexa」关键词实现远距离设备唤醒。

组装流程与初步的 APP 设定流程可以参考 Lanma 的教学(【实测心得】智能语音 WisCore 的安装笔记 <https://makerpro.cc/2017/10/wiscore-installation-notes/>)

身为 Maker，当然不会满足于原厂提供的 APP 和 Alexa Service，自己做才是王道，WisCore 为 Maker 提供丰富的软硬件支持，内建 OpenWRT 路由器操作系统，具备量产级的稳定性，以及 OpenWRT 生态系丰富的套件与软件架构，OpenWrt 是一种针对嵌入式设备，且具备高度可扩展性的 GNU/Linux 发行版，提供高度模块化、网络组件和扩展性的嵌入式 Linux 系统，常常被用于无线 AP、路由器、工控设备、网络电话、小型机器人、智能家电以及 VOIP 设备中。那怎么建构开发的第一步呢？首先，嵌入式统开发多透过 UART 的终端机操作接口来进行，WisCore 本身已具备 USB-UART 的功能，也就是开发者不需要透过额外的转接板或复杂的接线，就可以接上 Linux 系统的终端之上，那我们来试看看吧，请准备一条 USB Micro Cable(即手机的充电线，如下图)



将变压器、USB Micro Cable、Wi-Fi 天线、喇叭与主板模块接成下图：

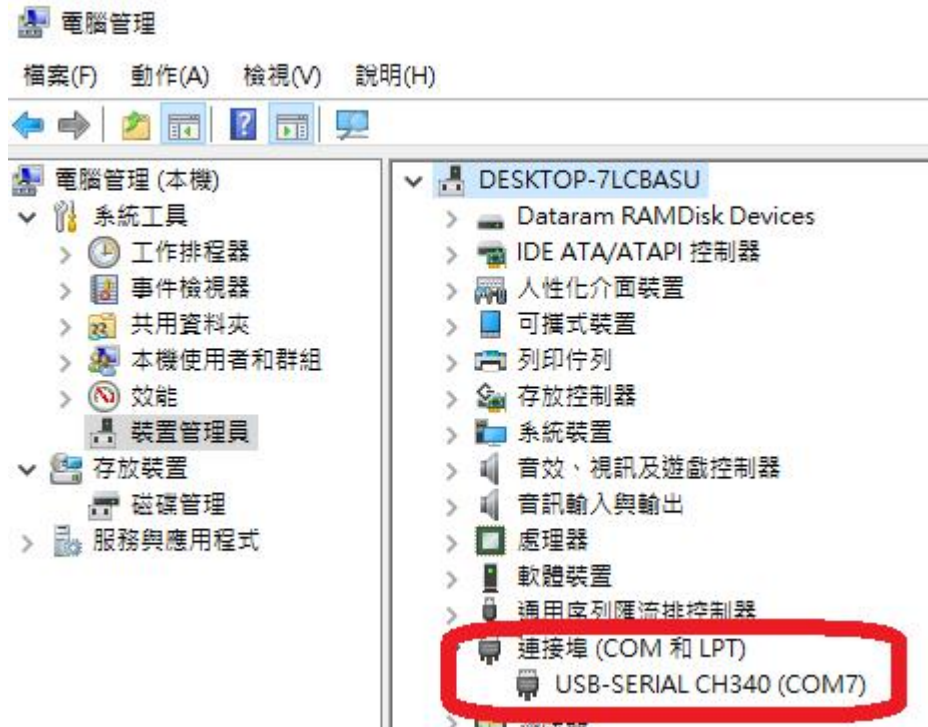


依照下列步骤:

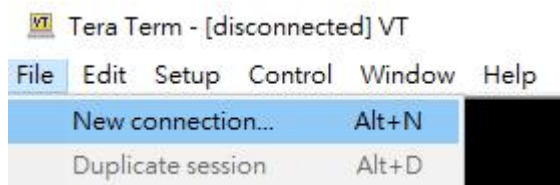
1. 计算机安装 USB-UART CH340 Driver

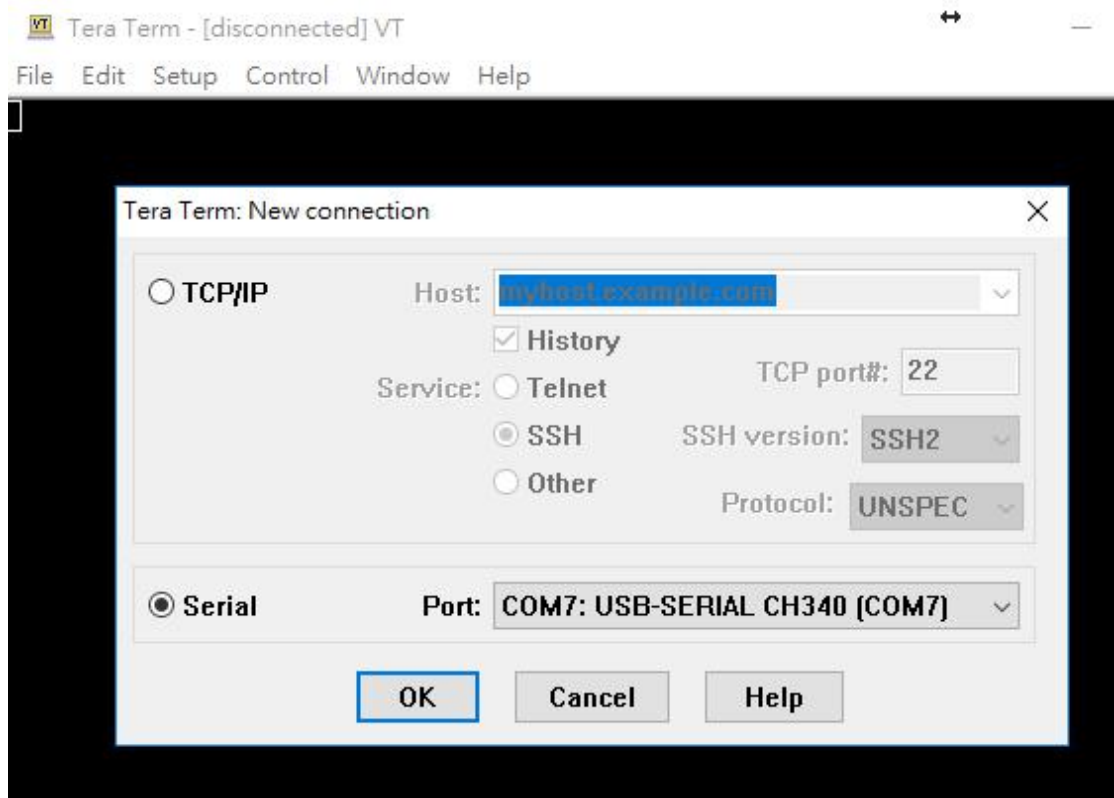
https://sparks.gogo.co.nz/assets/_site/downloads/CH34x_Install_Windows_v3_4.zip

2. 计算机安装终端机软件 TeraTerm
<https://osdn.net/projects/ttssh2/downloads/68252/teraterm-4.96.exe/>
3. 将 Wi-Fi 天线、喇叭接上 WisCore 主机
4. 变压器接上 WisCore 主机上电
5. 将 USB Micro Cable 接上 WisCore 主机
6. 计算机打开设备管理器，查询端口为 COM7(数字依计算机和 USB 接孔而易)

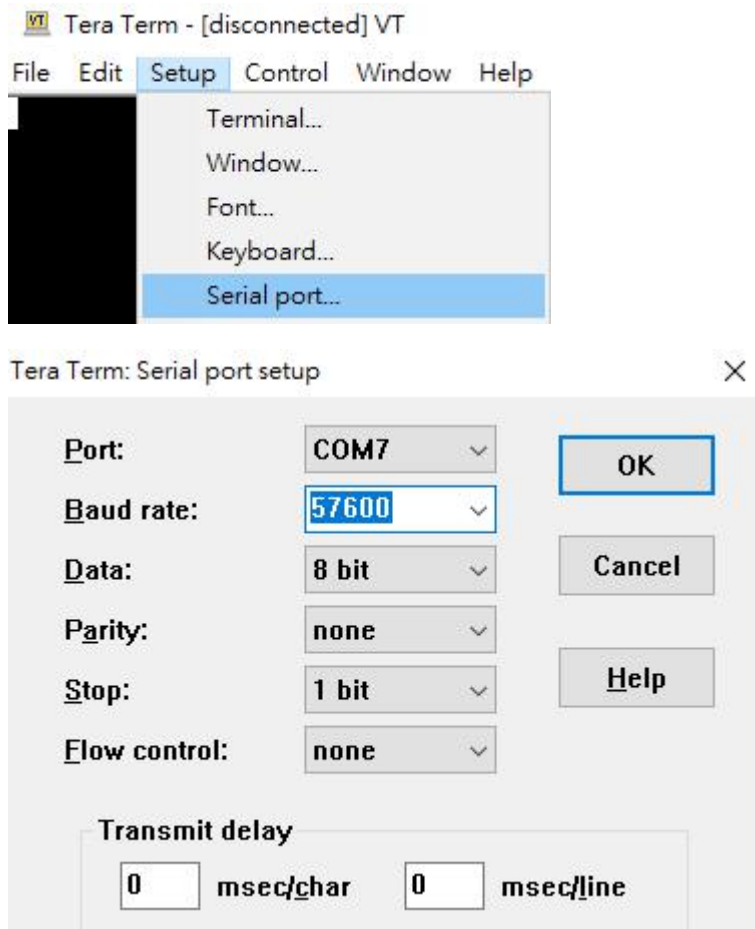


7. 开启 TeraTerm -> File -> New Connection，开启对应的 COM port





8. Setup -> Serial Port 设定 COM port 参数，Baud rate 为 57600 8N1 如下图



这块板子整合了量产等级的语音助理 SDK，与具备 Far-field/Hand-free 的声控硬件，对 Maker 而言是一块深具开发潜力的平台，但在尝试开发 WisCore 的物联网应用时，发现几个仍需加强的部分，首先，该平台虽然基于 OpenWRT 为基底，但系统软件不支持 Python 或 NodeJS，官方也未经营该平台的 opkg 套件库，opkg 软件套件数目远少于同属 MTK 的 Linkit 7688 以及 Respeaker，导致物联网相关应用软件开发不易，另外，截至目前最新版的韧体(release - Eng-rc503)，仍有开发板上 USB 以及 SD 卡无法运作的 Bug，也让开发者处处受限，无法加上额外的装置(如 USB WEBCAM 或 USB Bluetooth Dongle)和储存空间，硬件设计优秀，但软件开发不足，一向是两岸电子厂商需要加强的部分，也希望 RAK Wiresess 继续秉持着支持开源的态度，加强软件支持度的开发，让 Maker 有更多好玩好用的开发版。

RAKWisCore 官方网站:

<http://www.rakwireless.com/cn/WisKeyOSH/WisCore>

RAK WisCoreWiki:

<https://github.com/RAKWireless/WisCore/wiki>

RAK WisCoreGithub:

<https://github.com/RAKWireless/WisCore>