

AV3900A

无线电监测接收机

用户手册



中电科仪器仪表有限公司

本手册是《AV3900A 无线电监测接收机用户手册》第一版，版本号是 1.2.0

本手册中的内容如有变更，恕不另行通知。



声明： 本手册内容及所用术语最终解释权属于中电科仪器仪表有限公司。

本手册版权属于中电科仪器仪表有限公司，任何单位或个人非经本所授权，不得对本手册内容进行修改或篡改，并且不得以赢利为目的对本手册进行复制、传播，中电科仪器仪表有限公司保留对侵权者追究法律责任的权利。

前言

非常感谢您选择中电科仪器仪表有限公司研制生产的 AV3900A 无线电监测接收机。

我们将以最大限度满足您的需求为己任，为您提供高品质的仪器，同时带给您一流的售后服务。我们的一贯宗旨是“质量优良，服务周到”，提供满意的产品和服务是我们对用户的承诺，我们竭诚欢迎您的垂询，联系方式：

服务咨询 0532-86889847
技术支持 0532-86891085
传 真 0532-86889056
网 址 <http://www.ei41.com>
电子信箱 eiqd@ei41.com
地 址 山东省青岛经济技术开发区香江路 98 号
邮 编 266555

本手册介绍了中电科仪器仪表有限公司研制生产的 AV3900A 无线电监测接收机的用途、使用方法、使用注意事项、性能特性、基本工作原理和故障诊断等内容，以帮助您尽快熟悉和掌握仪器的操作方法和使用要点。为方便您熟练使用该仪器，请仔细阅读本手册，并正确按照手册指导操作。

本用户手册共八章：

第一章概括地讲述了 AV3900A 无线电监测接收机的一些基本情况，包括主要技术特点、具备的或可以实现的各种功能、所能达到的技术指标。

第二章到第四章主要是使用说明部分。其中第二章说明了接收机的指标描述和使用注意事项；第三章主要介绍了整机的前面板和外部接口等；第四章是基本操作方法，着重介绍驱动库的使用。

第五章到第六章是技术说明部分，简要叙述了接收机的工作原理，给出了主要技术指标的测试方法。

第七章是维修说明部分，包括故障查询步骤及出错信息说明以及返修方法。

由于笔者水平有限，文字中疏漏和不当之处，恳请您批评指正！由于我们的工作失误给您造成的不便我们深表歉意。

编 者
2016 年 4 月

环境、安全说明

一、安全保护

1、仪器自身安全注意事项

- 1) 仪器运输过程请使用指定包装箱，搬运时避免跌落或碰撞造成仪器损伤；
- 2) 请选用 220V 交流三芯稳压电源为适配器供电，防止大功率尖峰脉冲干扰对仪器内部硬件造成毁坏；
- 3) 保证电源良好接地，接地不良或错误可能导致仪器损坏；
- 4) 操作仪器时请采取佩戴静电手腕等防静电措施，严防静电对仪器的损害；
- 5) 本仪器可使用内部电池供电，如需更换，请使用相同类型或推荐相当类型的电池进行替换，否则存在爆炸的危险；
- 6) 仪器使用时内部温度较高，设计中通过传导至机壳散热，当操作使用时，尤其装入机柜时，请为机箱保持至少 5cm 有效散热空间，并确保通风顺畅，或者可设计将机壳热量传导至有效散热导体上。

2、人身安全保护注意事项

- 1) 搬运仪器时请选取合适的搬运工具，并轻放，以免仪器跌落造成人身伤害；
- 2) 保证电源良好接地，接地不良或错误可能造成人身伤害；
- 3) 如果需要擦拭仪器，请断电操作，防止发生触电危险，可以用干的或稍微湿润的软布擦拭仪器外表，千万不要试图擦拭仪器内部。

二、环境保护

1、包装箱的处理

我单位承诺产品包装物为无害物，请保留好包装箱和衬垫，以备将来需要运输时使用，也可以按照当地环境法规要求处理产生的包装物。

2、报废处理

仪器在维修及升级过程中更换下来的零部件由中电科仪器仪表有限公司集中回收处理；仪器报废后禁止随意丢弃或处置，请通知中电科仪器仪表有限公司或交由具有资质的专业回收单位进行回收处理。

除非另有规定，以上操作请按照国家《废弃电器电子产品回收处理管理条例》和当地环境法律法规处置。

目 录_Toc453665193

环境、安全说明	1
第一篇 使用说明	2
第一章 概述	3
第一节 产品综述	4
第二节 主要技术指标	5
第二章 使用指南	7
第一节 开箱自检	7
第二节 安全须知	7
第三章 接口说明	9
第四章 操作指导	10
第一节 使用演示软件	10
第二节 程控程序开发	17
第二篇 技术说明	38
第五章 工作原理	39
第六章 性能特性测试	40
第三篇 维修说明	54
第七章 故障信息说明及返修方法	55
第一节 正常性检查	56
第二节 故障说明	57
第三节 返修方法	58

第一篇 使用说明

第一章 概述

本手册使用下面这些安全符号，操作仪器前请先熟悉这些符号及其含义！



“警告”标识表示存在危险。它提示用户注意某一操作过程、操作方法或者类似情况。若不能遵守规则或者正确操作，则可能造成人身伤害。在完全理解和满足所指出的警告条件之前，不要继续下一步。

注意

“注意”标识代表重要的信息提示，但不会导致危险。它提示用户注意某一操作过程、操作方法或者类似情况。若不能遵守规则或者正确操作，则可能引起仪器损坏或丢失重要数据。在完全理解和满足所指出的小心条件之前，不要继续下一步。

第一节 产品综述

1. 产品简介

AV3900A作为一款小型化的接收机，适用于新一代的监测方式——网格化无线电监测。通过该接收机集群来组建基于TCP/IP的频谱感知网络，可以对小区、城市甚至整个国家进行全面持久、真实可靠、灵活智慧的无线电监测。该接收机的频率覆盖范围20MHz~6GHz，实时带宽20MHz，10kHz分辨率时的扫描速度达到4GHz/s，能够胜任复杂多样的信号监测任务；配备了高精度的GPS时间同步模块，可以实现多台接收机的同步测量；具有2GB的记录深度，可以持续进行无缝采集，轻松捕获瞬时和跳频信号。

2. 产品特点

- 频率范围20MHz~6GHz，最大分析带宽20MHz
- 2个射频输入端口
- 具备多种触发捕获功能
- 2GB的数据记录深度
- 100纳秒的时间戳精度
- IQ/频谱数据流输出
- 分析带宽内任意频点AM/FM解调
- 传感器型的网络设备
- 工作温度范围-20℃~55℃，小型化易安装
- 丰富的程控接口，包括网络管理、频时域测量等功能

3. 产品功能

- 高速频谱扫描监测
- 固定频点大带宽实时监测
- IQ数据流无缝输出
- AM/FM解调音频流与频时域数据流并行输出
- 幅度、时间触发

4. 典型应用

- 网格化无线电监测

无线电发射机正在朝着低功率、高频段、大带宽和复杂调制方式的方向发展，这就要求监测接收机能够尽可能的接近发射源，以便获得更高的信噪比，并降低多径效应的发生概率。通过在多点布设低成本的小型接收机将监测区域网格化，不但因靠近发射源而获得了更好的监测效果，而且不再需要昂贵的低损耗电缆和专门的监测场所。

- 无线电发射源定位

利用时间同步和TDOA定位技术，监测网络可以迅速锁定发射源的位置，这既可用于排查干扰源及非法电台，也可用于探测敌舰敌机的方位。TDOA定位不再需要专门的定向天线和设备，简化了系统组成并降低了成本，此外被动接收就能定位目标方位的隐蔽特性对于军事国防也具有重要意义。

- 民航高铁通信监测

借助接收机良好的接收性能，可对民航甚高频的AM信号和高铁沿线的GSM-R信号进行监测，及时发现通信异常和干扰，保障交通安全。

第二节 主要技术指标

内部时基

老化率:	$0.5 \times 10^{-6}/\text{年}$
温度稳定度:	$\pm 0.2 \times 10^{-6}$ (-40℃~50℃, 参考 25℃)
初始频率准确度:	0.5ppm

频率范围

20 MHz~6 GHz

边带噪声

$\leq -90 \text{ dBc/Hz}$	10kHz 以远
---------------------------	----------

显示平均噪声电平

前置放大器开:

$\leq -161 \text{ dBm/Hz}$	20MHz~2.7GHz
$\leq -158 \text{ dBm/Hz}$	2.0GHz~3.0GHz
$\leq -161 \text{ dBm/Hz}$	3.0GHz~4.0GHz
$\leq -158 \text{ dBm/Hz}$	4.0GHz~6.0GHz

前置放大器关:

$\leq -140 \text{ dBm/Hz}$	20MHz~2.0GHz
$\leq -136 \text{ dBm/Hz}$	2.0GHz~3.0GHz
$\leq -137 \text{ dBm/Hz}$	3.0GHz~4.0GHz
$\leq -135 \text{ dBm/Hz}$	4.0GHz~6.0GHz

二次谐波失真 (混频器电平-30dBm)

$< -80 \text{ dBc}$

三阶交调失真 (混频器电平-30dBm)

$< -80 \text{ dBc}$

镜频抑制

$\geq 110 \text{ dB}$	20MHz~3.0GHz
$\geq 85 \text{ dB}$	3.0GHz~6GHz

中频抑制

$\geq 100 \text{ dB}$

剩余响应

$< -85 \text{ dBm}$	前置放大器关
$< -105 \text{ dBm}$	前置放大器开

幅度准确度

$\pm 3.0 \text{ dB}$

衰减器切换不确定度

$\pm 1.20 \text{ dB}$

输入电压驻波比(输入衰减器 $\geq 10 \text{ dB}$)

< 2.5:1

最大输入电平

+20dBm

信号捕获

内存容量: 2GB

信号持续捕获时间:

最大带宽 20MHz 约 18.2 秒

最大带宽 10kHz 约 10.4 小时

数据时间戳准确度: <100ns

信号分析

最大分析带宽: 20MHz

扫描速度: >4GHz/Sec RBW 为 10kHz

射频输入端口

接口: 两路, N 型, 50 Ω

端口隔离:

>40dB 20MHz~600MHz

>30dB 600MHz~6GHz

第二章 使用指南

第一节 开箱自检

1. 开箱确认

当您打开包装箱后，您会看到一份装箱清单及以下物品：

- a) AV3900A 无线电监测接收机1 台
- b) 电源适配器1 个
- c) 电源线组件1 根
- d) 产品快速使用指南1 本
- e) 随机光盘1 张
- f) 选件（根据订货信息选配）若干
- g) 产品合格证1 张

请您根据订货合同和装箱清单仔细核对以上物品是否有误，如有问题，请根据前言中的联系方式与我所经营中心联系，我们将尽快予以解决。

注意

因仪器属于贵重物品，移动时，应轻拿轻放。

2. 外观检查

仔细观察仪器在运输过程中是否有损伤，当仪器有明显损伤时，严禁通电开机！请根据前言中的联系方式与我所经营中心联系。我们将根据情况进行迅速地维修或调换。

第二节 安全须知

AV3900A 无线电监测接收机安全性符合 GJB3947A-2009 规定要求。本仪器内部没有可供用户操作的零部件，请不要擅自打开仪器外壳，否则可能造成人身伤害。为保证您的安全以及正确使用仪器，使用前请务必仔细阅读以下安全须知事项。

1. 环境要求

为保证 AV3900A 的使用寿命及测量的有效性和准确性，请在以下环境条件下进行测试：

- 温度范围：
存储温度范围：-40℃~70℃
工作温度范围：-20℃~55℃
- 低气压：
低气压：0~4600m（海拔高度）

2. 电源线的选择

AV3900A 无线电监测接收机采用符合国际安全标准的三芯电源线。使用时，插入带有保护地的合适电源插座，以便电源线将仪器的机壳接地。推荐使用随机携带的电源线，在更换电源线时，建议使用同类型的 250V/10A 电源线。

3. 供电要求

AV3900A 无线电监测接收机采用交流电源、适配器供电，接收机在工作时必须使用随机配备的 AC-DC 适配器。适配器的输入为 100~240V、50/60Hz 交流电。接收机在接通电源的情况下会自动进入工作状态，所以在运输和携带过程中，为了避免仪器损坏，请不要接通仪器电源。

注意

工作电压与频率范围以所配电源适配器铭牌标识为准。

当采用非标准市电时（如车载逆变器），请先测试待接入接收机适配器的交流电是否符合要求，电压不稳定可能导致接收机某些模块工作异常。

4. 静电防护（ESD）

静电对电子元器件和设备存在极大的破坏性，所以仪器加电工作时必须在防静电工作台上操作。在使用仪器时，应注意静电防护。如条件允许，可采取如下静电防护措施：

- 保证所有仪器正确接地，防止静电生成。
- 工作人员在接触接头、芯线或做任何装配操作以前，必须佩带防静电腕带。
- 将电缆连接到仪器进行测试之前，一定要使电缆的中心导体首先接地。可以通过以下步骤来实现：在电缆的一端连上短路器使电缆的中心导体和外导体短路，当佩带防静电腕带时，抓紧电缆连接器的外壳，连好电缆的另一端，然后去掉短路器。

5. 输入端口保护

接收机拥有两路射频输入，射频端口标准阻抗是 50Ω。因此使用过程中应严格按照端口要求加入测试信号或者端接合适的负载阻抗，防止损坏后级电路。



警告

接收机的射频输入端有最大允许输入电平，严禁超限加入信号，否则会引起仪器损坏。

在市内的高层楼顶架设监测点进行调试时，请先将接收机的衰减器设置为最大值（30dB），以免 FM 广播等大信号的输入导致接收机过载。

第三章 接口说明

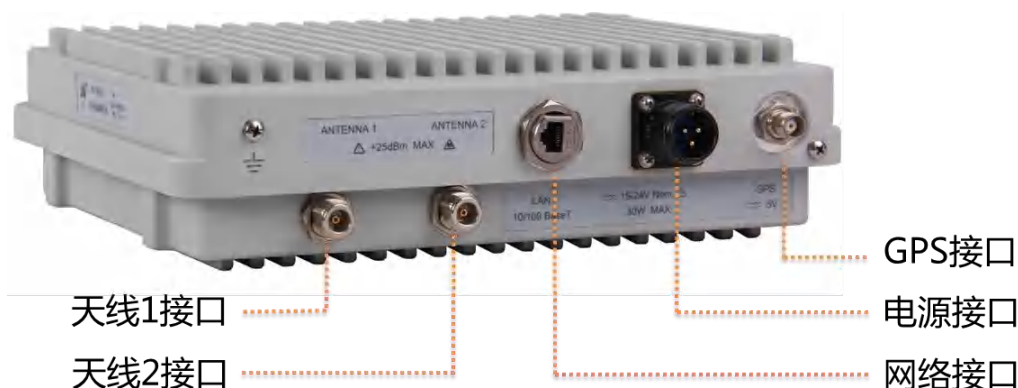


图 3-1 接收机接口

AV3900A 无线电监测接收机采用全密闭式的铝制外壳，所有的接口位于仪器的底部，如图 3-1 所示。

1. 天线接口

天线接口由“ANTENNA1”和“ANTENNA2”标签所标识，采用 N 型（阴）接头，最大输入功率是+25dBm。AV3900A 无线电监测接收机可同时连接两幅监测天线，并可由远程主机控制天线的切换，在某些监测场合中，可以采用一副覆盖低频段的天线和覆盖中高频段的的天线共同对 20MHz~6GHz 频段进行监测。

2. 网络接口

网络接口由“LAN”标签所标识，接收机可以连接任何符合 8P8C(RJ45)标准的网线插头。

3. 电源接口

电源接口由“15-24V Nom.”标签所标识，为了保证电源供电正常性和外壳的密闭性，请采用 AV3900A 专用的电源适配器进行连接。

4. GPS 接口

GPS 接口由“GPS”标签所标识，用于连接 GPS 天线，接收机内嵌 GPS 信号接收模块，可为接收机提供标准的 GPS 时钟信息。

注意

在接收机加电之后，其内部的 GPS 接收模块会有一个解析过程，通过跟踪 GPS 卫星并对卫星信号进行解析来获取标准 UTC 时间信息，这个过程可能会持续 10 分钟左右，在此期间的数据时间戳是不准确的。

第四章 操作指导

AV3900A 无线电监测接收机是一款传感器型的测量设备，与传统仪器不同，它没有屏幕和操作面板，取而代之的是功能强大的远程控制。为了方便您的使用，我们提供接收机的程控函数库和演示软件。通过程控函数库您可以快速开发自己的应用程序，灵活控制接收机完成监测任务。演示软件是一款基于上述程控驱动库所开发的应用程序，用来演示接收机的功能。本章的目的是通过演示软件让您熟悉接收机的操作，并指导您快速建立自己的程控程序。

第一节 使用演示软件

频谱演示软件

首先建立 PC 机的运行环境，选用一台安装了 Windows XP 或 Windows7 操作系统的 PC，把频谱演示软件安装到该 PC 机。设置 PC 机的 IP 地址为“172.141.11.1”，子网掩码为“255.255.255.0”。

第二步是建立连接，AV3900A 接入射频接收天线和 GPS 接收天线（选件），使用点对点网线连接 AV3900A 的网络接口与 PC 机的网口，接通接收机的电源，具体连接方式如图 4-1 所示。

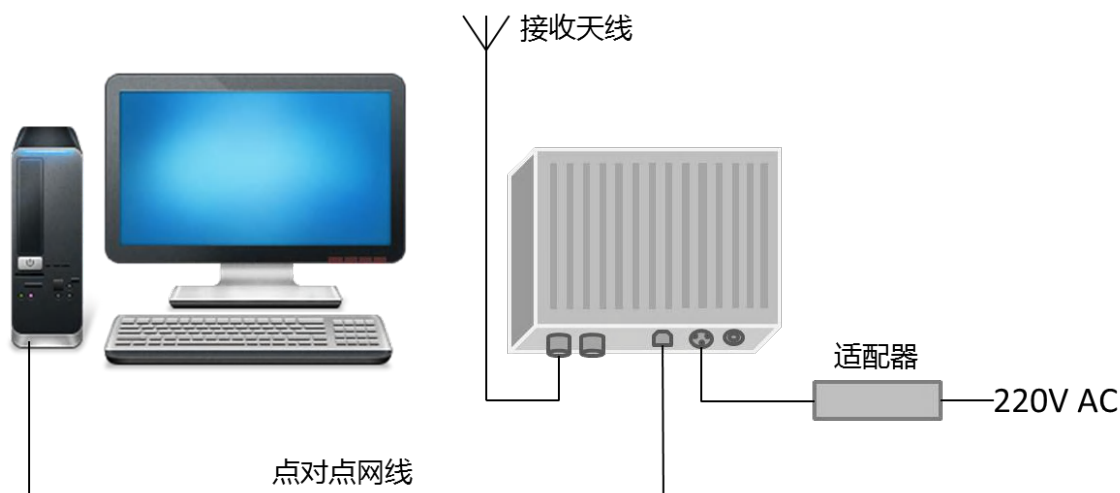


图 4-1 接收机与 PC 的直连形式

第三步是运行软件，在 PC 机桌面上双击演示软件的图标，演示程序开始运行，初始界面如图 4-2 所示。



图 4-2 演示程序初始界面

请在该窗口界面上遵照下面的指导操作，在进行三次鼠标点击之后，您就可以观察到接收机的频谱扫描迹线了。

1. 在功能区中，点击“IP 连接”的图标，如图 4-3 所示；



图 4-3 “连接”面板

2. 在弹出的对话框中，直接点击“连接”，如图 4-4 所示；



图 4-4 接收机 IP 设置对话框

3. 连接成功后，功能区会切换到【扫描控制】控制面板上，如图 4-5 所示，点击“启动扫描”，随即在图像区显示出发射机测量到的频谱数据，如图 4-6 所示。



图 4-5 “扫描控制”面板中的启动扫描

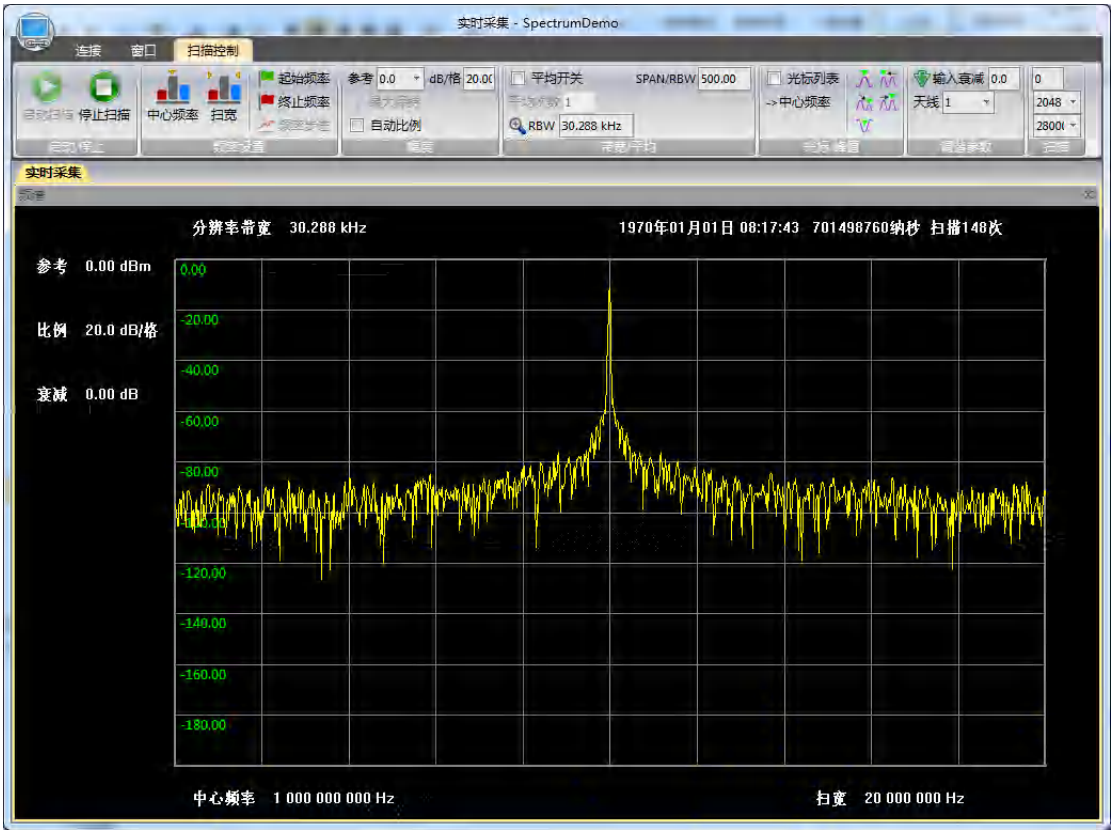


图 4-6 频谱迹线

功能与操作说明

1. 功能区

功能区位于窗口上部的带状区域，如果您操作过 Office2007 及以上版本的 Windows 办公软件，会对这种新型的操作界面比较熟悉。它糅合了传统应用程序的菜单栏和工具栏，将所有相关操作采用图标平铺的方式展现在用户面前，使得人机交互更加方便。

演示软件的功能区分为【连接】、【扫描控制】、【调试】等一级控制面板。【连接】用于设置 PC 机与接收机的连接；【扫描控制】主要用于设置频谱扫描参数，其次它还包含一些数据的分析工具如光标，峰值等；【调试】是开发人员用来设置接收机内部参数的接口。上述每个控制面板中都包含一些二级操作控件，用来进行更具体的单项设置。

1) 【连接】

初始的功能区处于【连接】控制面板，在该面板上您可以进行搜索、连接、修改接收机 IP 地址的操作。

a) [搜索]

如果您安装了接收机网络管理服务（SMS），就可以通过该服务在本子网内搜索所有接入网络的 AV3900A 接收机了。但在没用启用 SMS 服务时，[搜索]将不起作用。

b) [IP 连接]

点击该按钮，会弹出一个对话框要求您填入一项 IP 地址。该地址是您希望连接的接收机的 IP。该对话框中的 IP 地址默认为“172.141.11.202”，此 IP 即接收机的默认出厂 IP。所以在“快速使用指南”一节中，当您已用点对点网线将 PC 机与接收机直连的前提下，直接在对话框中点击“连接”，即可建立与接收机的连接了。

c) [修改 IP 地址]

当建立连接后，您可以通过程控修改接收机的网络参数。在设置对话框中填入欲设置的接收机网络参数，点击确认，接收机会根据您的设置修改自己的网络参数，当前的连接也会中断。在等待 20 秒左右的时间后，您可以使用[IP 连接]重新进行连接。

2) 【扫描控制】

当连接建立起来后，软件会自动切换到【扫描控制】面板。在此您可以对接收机的扫描参数进行设置。整个【扫描控制】面板如图 4-7 所示。



图 4-7 “扫描控制”面板

它包含了“启动/停止”、“频率设置”、“幅度”、“带宽/平均”、“光标/峰值”、“调谐参数”、“扫描”七个子面板。

启动/停止

[启动扫描]用来启动接收机进行扫描，注意只有在扫描中才可以更改扫描参数。

[停止扫描]用来停止接收机扫描，停止之后整个面板只有[启动扫描]和[光标列表]处于使能状态。

频率设置

频率设置面板用于设置频谱扫描的频率范围，设置有两种方式，一是通过中心频率和带宽进行设置，二是通过起始终止频率进行设置。



图 4-8 “频率设置” 面板

点击任意一个频率设置项，都会弹出一个对话框，要求您输入频率值，如图 4-9 所示。在对话框中输入一个频率值需要填入频率数值，然后选择频率单位。如图所示的频率设置将会将接收机的中心频率设置为 2.4GHz。

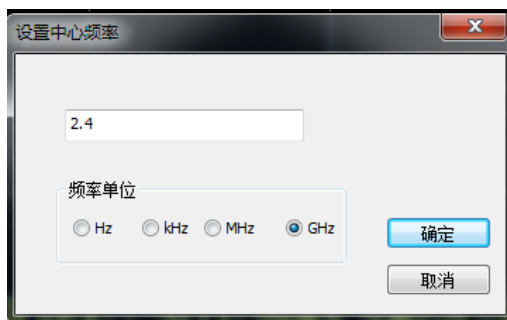


图 4-9 “频率设置” 对话框

幅度

幅度用于设置图像区的幅度比例，参考电平等。如图 4-10 所示。



图 4-10 “幅度” 面板

a) [参考]

您可以自己手动输入参考电平，也可以从下拉列表框中选择参考值。参考的范围是[-80, 20]，注意该参考只是调整图像区的幅度显示上限，并没有和接收机的调谐参数进行关联，修改参考并不会改变调谐电路的增益或衰减。

b) [刻度/格]

您可以在该编辑框内设置显示区中每纵格代表的幅度值。可设置范围是[0.1, 20]。

c) [自动比例]

您可以点击该控件，让程序自己设置合适的参考电平和刻度/格。调整之后，迹线将显示在图像区靠中心的位置，迹线跨度为 8 个纵格。

带宽/平均

带宽/平均子面板用来设置扫描的分辨率带宽和平均。如图 4-11 所示。



图 4-11 “带宽平均” 对话框

您可以通过[RBW]手动直接设置分辨率带宽,由于接收机的采样率和 FFT 点数都是基 2 变化的,这意味着接收机可接受的分辨率带宽是离散的,程序将会自适应的将接收机设置到与您的输入最接近的分辨率带宽上。

通过设置[SPAN/RBW],您可以让程序根据当前的扫宽,按照 SPAN/RBW 的比例关系,自动设置分辨率带宽。[SPAN/RBW]和[RBW]是关联的,您修改其中的一项,有可能会影响到另外一项的设置。

[平均次数]用来设置接收机本次扫描的平均次数。当[平均开关]打开时,接收机对每一频段,都会执行[扫描次数]所标识的 N 次扫描,把 N 次扫描的 FFT 结果叠加在一起,然后做线性平均之后,再把数据输出。平均功能对于发现靠近底噪的小信号很有帮助,也可以用来观察信号在一段时间内的平均幅度。

光标/峰值

光标峰值面板用来分析图像区的迹线,通过光标您可以观察迹线任意频点的频率幅度,以及两个频点之间的频率幅度偏移。通过峰值您可以快速定位当前频谱图像上的信号位置及幅度。



图 4-12 “光标峰值”对话框

当您点击[光标列表],显示区底部会出现一个光标列表的操作窗口。在这个窗口中您可以进行光标设置,如图 4-13 所示。

光标

光标ID	普通	差值	基准频率	基准幅度	差值频率	差值幅度	噪声光标
1	☒	☐	998 235 294 Hz	-99.77 dBm	--	--	☐
2	☐	☐	--	--	--	--	☐
3	☐	☐	--	--	--	--	☐
4	☐	☐	--	--	--	--	☐

图 4-13 光标列表

光标列表共提供了 4 个光标,每个光标都可以独立设置基准光标、插值光标和噪声光标模式,每个光标占列表的一行,行中的“普通”、“差值”、“噪声光标”用来设置光标模式,“基准频率”和“基准幅度”用来显示基准光标的位置信息,“差值频率”和“差值幅度”用来显示差值光标的位置信息,“差值频率”是可设置的,您可双击这一格编辑差值光标的位置。

点击[->中心频率]将会把中心频率设置到当前光标所标注的频率上去。这个功能对于观察信号及其信号附近的噪声很有帮助。

该面板的右栏是峰值功能,[峰值]将把当前光标定位在当前迹线的最大峰值处,[次峰值]将把光标定位比当前光标幅度小的所有峰值的最大点,[左邻峰值]和[右邻峰值]会把光标移到与其位置最邻近的峰值处,[最小值]把光标定位于迹线的幅度最低点。

调谐参数和扫描

调谐参数和扫描用来设置接收机的扫描参数。其中“调谐参数”用来设置射频通道的增益、衰减、天线输入端口;“扫描”用来设置数字中频电路的采样率、FFT 点数和扫描次数。如图 4-14 所示。



图 4-14 调谐参数和扫描参数

[输入衰减]用来设置通道的衰减和增益，范围是[-20, 30]，在接收机中由前置放大器和衰减器共同起作用来调整通道的总衰减满足您的衰减设置要求，当[输入衰减]为负时，前置放大器起主要作用，当[输入衰减]为正时，衰减器起主要作用。

在扫描子面板中，[扫描次数]用来设定本次扫描会扫几次，接收机从起始频率扫描到终止频率算作一次扫描，如果扫描次数为 N（N 不为零），那么接收机会在执行 N 次扫描之后停止，如果 N 为零，那接收机会连续扫描直到您点击[停止扫描]。[FFT 点数]用来设置数字中频电路中进行 FFT 运算的帧长度，这个参数与采样率共同决定了分辨率带宽，FFT 点数只可以设置成 2 的幂，范围是[8, 8192]。采样率用来设置接收机的数字中频的抽取倍数，只能是 28MHz 的 2 分数，例如 14MHz、7MHz、3.5MHz 等等。

2. 图像区

图像区位于功能区的下方，占据了窗口的大部分区域，主要用来观察接收机的测量结果，如图 4-15 所示。它的主体是一个 10x10 方格的迹线区，测量结果将以迹线的形式在该区域内显示，迹线的幅度刻度显示在迹线区的左边框，此外，光标以及光标读数也会显示在迹线区，当打开光标功能时，您可以直接使用鼠标拖拽光标到指定位置。迹线区的左方是与幅度显示范围相关的参数信息，上方是分辨率带宽信息，下方是频率范围信息，双击频率的标签，同样可以弹出对话框设置频率范围。

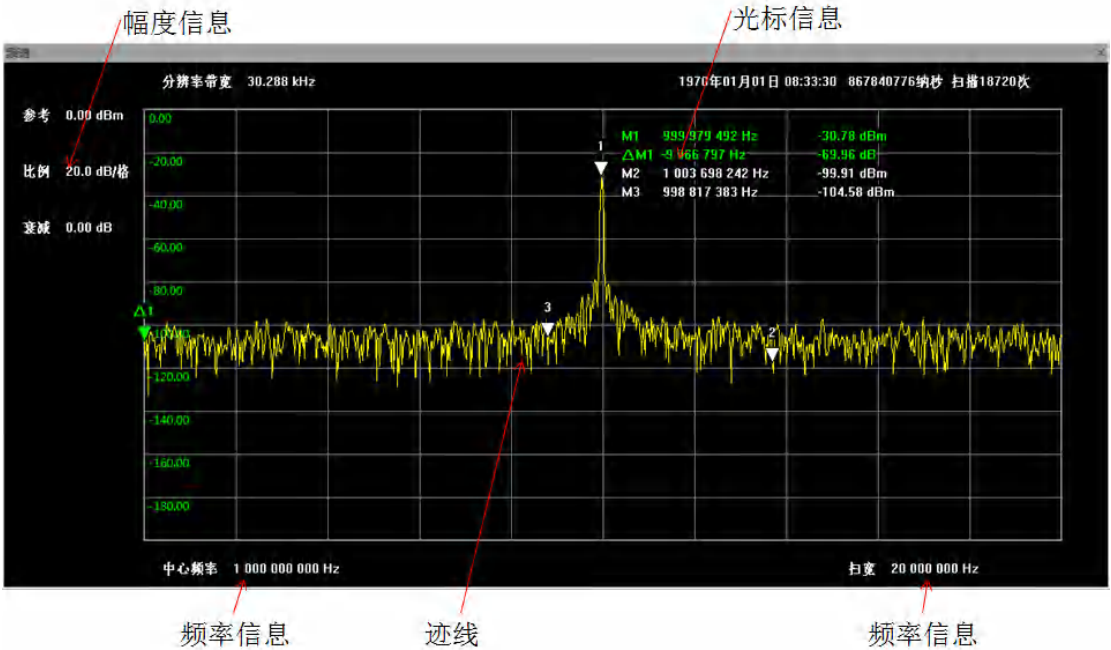


图 4-15 图像区

注意

当迹线没有正常出现在屏幕上时，检查参考电平和标尺比例是否合适。

第二节 程控程序开发

通过上一节的介绍，您已经对接收机的使用有了基本的了解。但这只是接收机提供的一部分功能，通常无法满足您的测量需求。如果需要使用更高级的功能，或者您要进行二次开发，就需要借助我们提供的驱动函数库，该库提供了接收机的所有控制接口。熟悉这些接口以及开发流程，您将可以充分利用接收机的测量能力，组建分布式的监测网络和监测系统。

本节以微软 Visual Studio 的 VC 开发环境为例，说明程控程序的开发过程。本章将通过 win32 的控制台程序，向您展示如何建立与接收机的连接以及如何采集频谱和 IQ 数据，在这个过程中会向您介绍接收机重要的 API 函数及其用法。本库同样支持 CVI、VB、Java 或 .net 等 Windows 平台下的其他开发环境，如果您使用这些环境进行二次开发，请参照相关例程进行操作，也可以联系我们寻求帮助。

驱动库的组成

驱动库由 6 个文件组成，分别是 EISAL.dll、EISAL.lib、EISAL.h、SalFrequency.h、SalTimeData.h、SalDeveloper.h。EISAL.dll 和 EISAL.lib 分别是驱动函数库的动态链接库和导入库；剩下的 4 个是头文件，包含了库中 API 的函数原型，以及 API 用到的结构体、枚举及其它一些常量的定义。

EISAL.dll 和 EISAL.lib 是驱动库的库文件，EISAL.dll 中包含了所有 API 的函数实体；EISAL.lib 用来帮助您的程序正确引用到动态库的 API（如果使用动态加载机制或其他语言如 .net，EISAL.lib 可以不用）。

头文件中定义了 API 的接口形式以及相关的结构体，枚举和常量。按照 API 的功能划分，共使用了四个头文件：

- EISAL.h 中主要定义了接收机通用的功能，例如接收机的连接、断开、接收机网络的管理等；
- SalFrequency.h 中定义了接收机频域数据采集的功能；
- SalTimeData.h 中定义了接收机时域数据采集的功能；
- SalDeveloper.h 中定义了接收机的扩展功能和调试功能。

驱动库的使用

首先使用 Visual Studio(在此以 VS2008 为例)创建一个 VC 的控制台工程。打开 VS2008 后，依次选择[File]->[new]->[Project]，在弹出的对话框中选择“Win32”工程类型，在右侧的工程模板中选择“Win32 Console Application”，即 Win32 控制台应用程序，在下方填入工程名字 AV3900A_TEST，点击“OK”，在后面的向导中直接选择“Finish”。如图 4-16 所示。

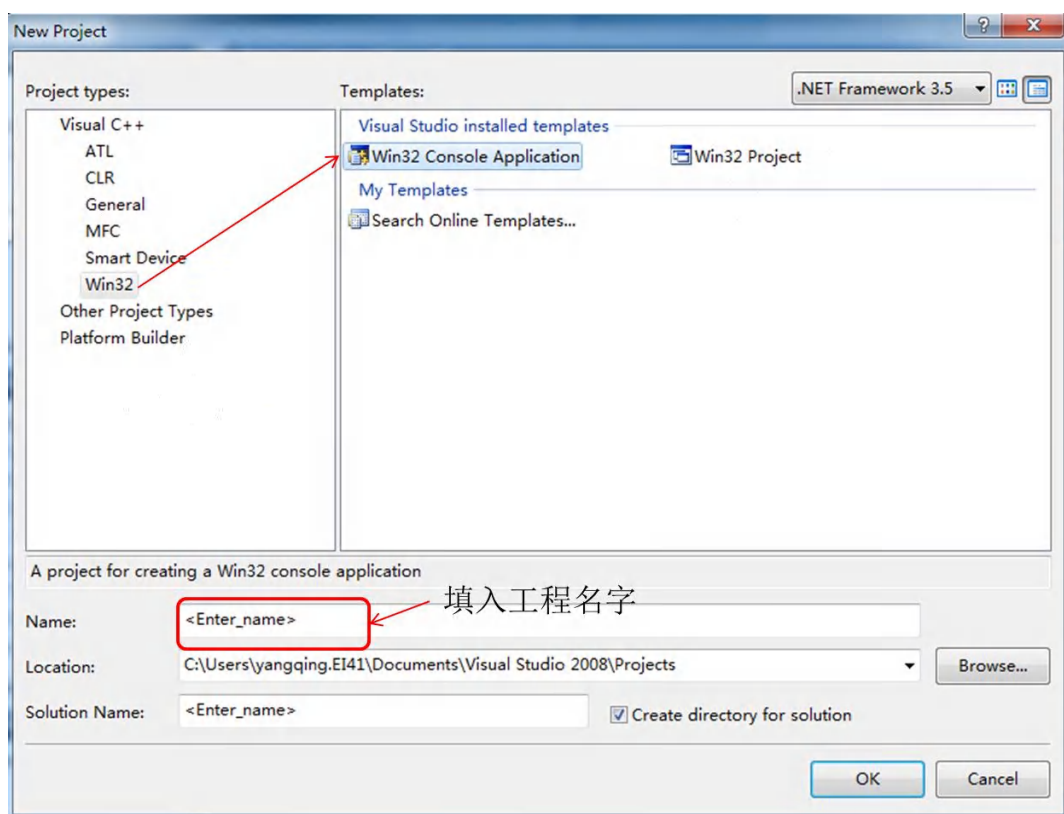


图 4-16 创建工程

然后，把驱动库中的所有文件复制到该工程源代码文件所在的文件夹中。在“Solution Explorer”窗口的工程名字处，点击鼠标右键，在弹出的菜单中点击“Add”->“Existing Item...”，把库中的头文件加入工程列表中。如图 4-17 所示。

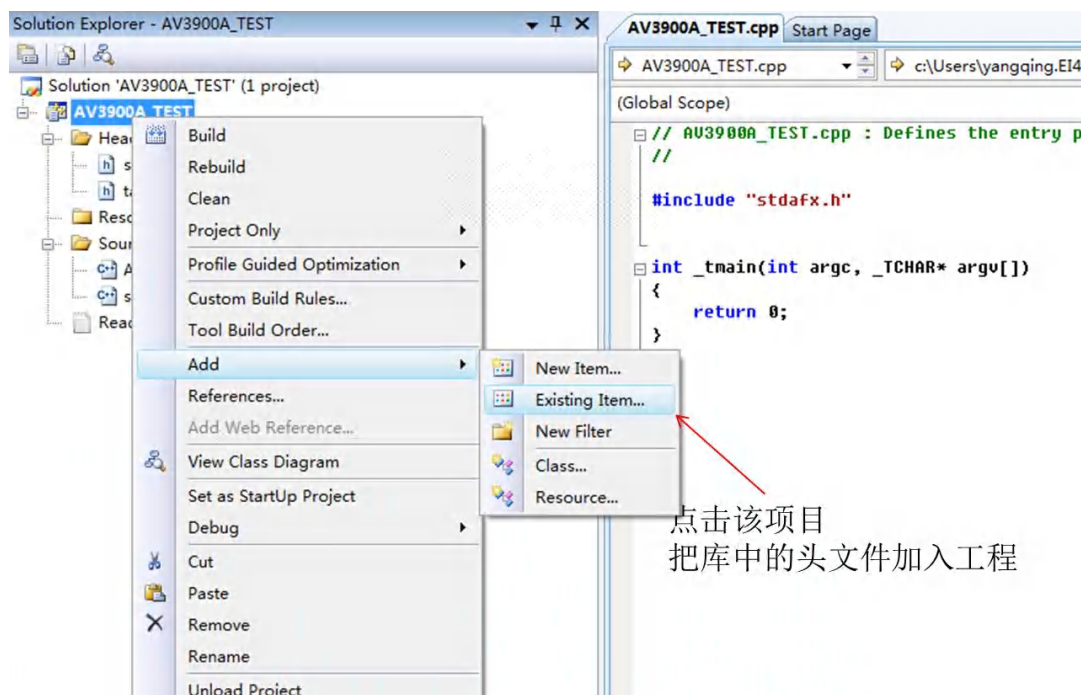


图 4-17 添加文件

在【Project】菜单的底部，点击“Properties”，在弹出的对话框中依次点击[Configuration Properties]->[Linker]->[Input]，在右侧的列表框的第一项里添加驱动库的导入库文件“EI-

SAL.lib”。如图 4-18 所示。

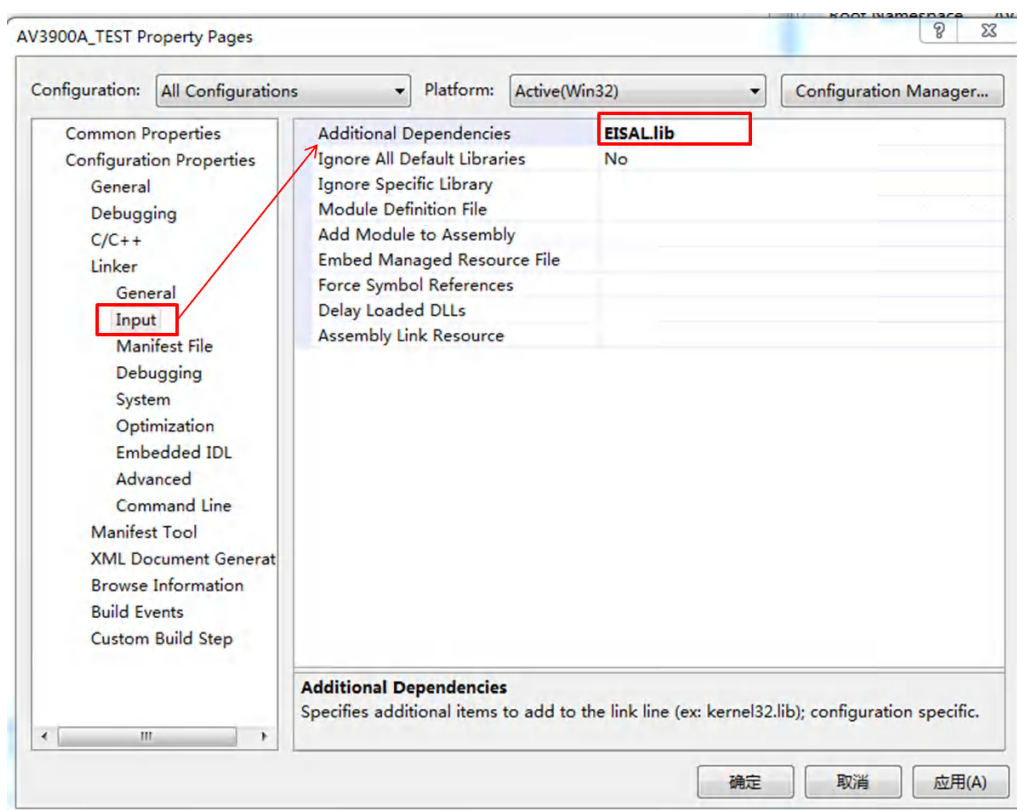


图 4-18 添加驱动库的导入库

至此，开发所使用的环境已经搭建完毕，Build 一次这个工程，确保其编译通过，在此基础上，我们可以开始程控程序的开发了。在进行下面的步骤之前，请先确定接收机已连接到 PC 机，并且接收机处于开机状态，PC 机上的演示软件可以正常工作。

连接例程

指定 IP 连接

用于连接接收机的 API 叫做 `salConnectSensor2()`，位于 `eisal.h` 头文件中。程序中调用该函数，可以建立与一个接收机的连接，如果连接成功，函数会返回一个句柄，这个句柄在您的程序中就代表了这台接收机，您可以在后续代码中使用该句柄，调用其他的 API 程控接收机了，在控制完成后，您一定要使用 `salClose()` 关闭这个句柄，以释放该连接所占用的资源。

我们可以在工程的 `main` 函数中调用这两个函数，为了更好的观察程序的运行情况，请先添加 C++ 标准库输入输出流头文件 `<iostream>`，以便将运行信息打印到控制台中。我们还需要包含驱动函数库头文件“`eisal.h`”，然后就可以在 `main` 函数的 `return` 语句之前加入如图所示的代码。

代码键入完毕后，按 F5 编译并运行，如果运行正常，在控制台上将打印出“已连接”的信息，说明接收机已经连接上 PC 机，随后您可以敲击键盘任意键退出程序。如图 4-19 所示。

```

#include "stdafx.h"
#include <iostream> //包含标准库的输入输出流
#include "EISAL.h" //包含库中的头文件

using namespace std;

int _tmain(int argc, _TCHAR* argv[])
{
    //定义一个代表接收机的句柄变量
    salHandle sensorHandle;
    //连接接收机
    salErrorType error = salConnectSensor2(&sensorHandle, NULL, "172.141.11.202", "", 0);
    if(error == SAL_ERR_NONE)
    {
        cout << "已连接\n";
        //断开与接收机的连接
        salClose(sensorHandle);
    }
    else
    {
        cout << "连接失败\n";
    }

    char input;
    cin >> input;
    return 0;
}

```

图 4-19 连接例程代码

■ SalConnectSensor2

函数声明:

```

SAL_API salConnectSensor2(salSensorHandle *sensorHandle,
                           salHandle smsHandle,
                           char *sensorName,
                           char *applicationName,
                           salInt32 options)

```

函数功能:

连接由 sensorName 指定的接收机。

参数列表:

参数类型	参数名	参数说明
salSensorHandle*	sensorHandle	[out] 接收机 handle 的指针
salHandle	smsHandle	[in] salOpenSms() 返回的 handle 或是 NULL
char*	sensorName	[in] 接收机的 IP 地址字符串
char*	applicationName	[in] 用户定义的应用名称
salInt32	options	[in] 保留扩展。设置为 0

备注:

在主机与AV3900A直连时，请把smsHandle设为NULL，SMS（Sensor Management Service）是一个Windows服务，用来管理整个接收机网络，后面会有详细介绍。但在这里，即使接收机没有被SMS所管理，也可以将sensorName设置成接收机的IP地址，让客户机与接收机直接建立连接。

■ salClose

函数声明:

salErrorType **salClose**(salHandle handle)

函数功能:

关闭一个句柄。

参数列表:

参数类型	参数名	参数说明
salHandle	handle	[in]接收机的程控句柄

备注:

此函数会关闭一个handle，关闭handle时会释放所有与这个handle关联的资源，例如内存和socket连接。类似于Windows的API函数，一个句柄代表了一个被使用的资源，例如一个文件或一个IO外设，在Windows操作系统的API中都以句柄表示。

在本例中，建立连接后，接收机本身被一个SensorHandle的句柄所代表。除了这种句柄，在驱动函数库中还有其他一些实体，如测量任务，SMS连接等也会抽象成一个句柄，您对这些句柄操作，就相当于操作它背后的那个对象。

注意

资源使用完毕时，一定要使用 salClose()关闭句柄，释放资源，否则会导致内存资源或网络传输资源泄露，如果大量的网络连接得不到释放，接收机端的响应可能会变迟缓。

使用 SMS 服务

SMS 是一个运行在远程主机上的 Windows 服务，该服务用来管理接入网络的接收机集群，一般情况下，它与用户程控软件运行在同一台主机上。当您使用接收机搭建分布式的监测系统时，SMS 起到了网络管家的角色，它可以帮您搜索网络内的接收机，也可将指定的接收机加入测量组。接收机、SMS、驱动库和用户应用的关系如图 4-20 所示。我们通过下面的几个例子说明 SMS 所起的作用。

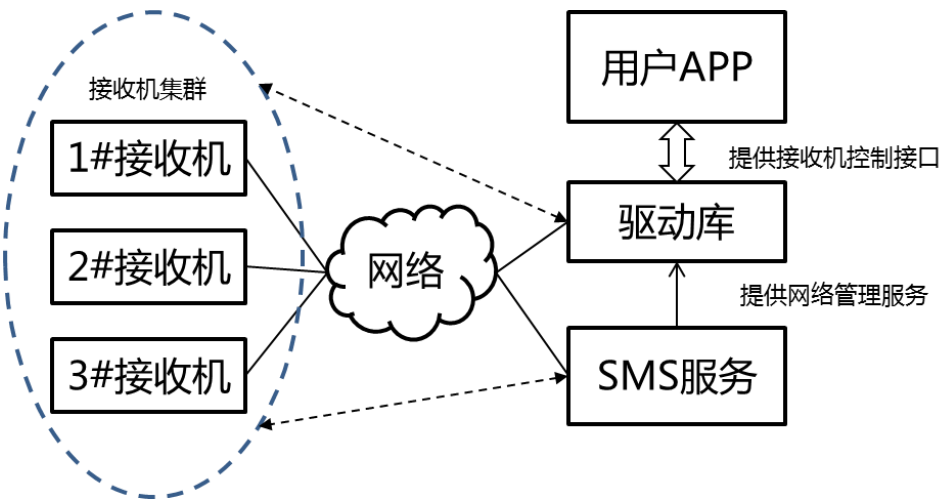


图 4-20 AV3900A 各软件之间的关系

发现网络内的接收机

首先，准备两台或以上接收机，通过以太网交换机同主控 PC 连接。主控 PC 要求已安装 SMS 服务并且 SMS 服务处于运行状态。连接形式如图 4-21 所示。例子中 PC 机所用的 IP 地址是“172.141.11.45”，两台接收机与 PC 机在同一网段。

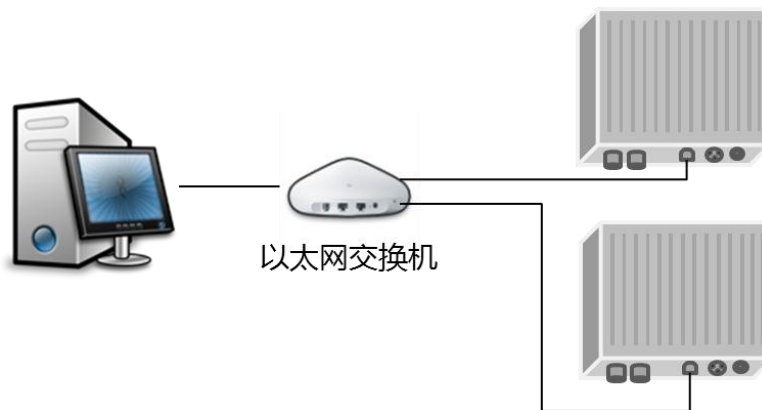


图 4-21 连接多台接收机

然后我们借助 SMS 的 API，来搜索发现这两台接收机。代码如图 4-22 所示。

```
int _tmain(int argc, _TCHAR* argv[])
{
    //定义一个代表SMS的句柄变量
    salHandle smsHandle;
    //打开SMS
    salOpenSms(&smsHandle, "172.141.11.45", 8089, "");
    if(smsHandle == NULL)
        return 0;
    salHandle discoverHandle;
    salUInt32 num;
    //调用salDiscoverSensors函数
    salDiscoverSensors(&discoverHandle, smsHandle, &num);
    if(discoverHandle == NULL)
        return 0;
    salSensorInfo info;
    cout << "Discover函数(发现局域网内所有的接收机):" << endl;
    for(unsigned long i = 0; i < num; i++)
    {
        if(salGetNextDiscoveredSensor(discoverHandle, &info) == SAL_ERR_NONE)
        {
            cout << "Sensor" << i+1 << ":[ip=" << info.ipAddress << "]" << endl;
        }
        else
        {
            break;
        }
    }
    //关闭句柄
    salClose(smsHandle);
    salClose(discoverHandle);
    char readKey;
    cin >> readKey;
    return 0;
}
```

图 4-22 利用 SMS 发现网络内接收机

在这段代码中，我们用到了三个与 SMS 相关的 API，分别是：

- salOpenSms()用于建立应用程序与 SMS 服务的连接；
- salDiscoverSensors()用来获取 SMS 发现的网络内的接收机列表；
- salGetNextDiscoveredSensor()用来枚举列表。

在 main()函数一开始，我们先调用 salOpenSms()建立起与 SMS 服务的连接，该函数接受三个参数，其中第一个是输出参数，代表 SMS 服务的句柄，第二个是字符串型的参数，

填入 SMS 服务所在主机的 IP，在本例中该 IP 是“172.141.11.45”，第三个是端口参数，即 SMS 服务所使用的端口号，其默认值为 8089，最后一个参数可选，设为空字符串即可。如果该函数调用成功，那么第一个参数就是代表 SMS 服务的句柄了。

与 SMS 建立起连接之后，我们再调用 `salDiscoverSensor()` 来查找 SMS 已找到的网络内的接收机的信息。该函数第一个参数也是输出参数，代表查找结果的句柄；第二个参数是前面建立连接生成的 SMS 服务的句柄；第三个参数是输出参数，代表 SMS 目前发现的接收机的数量。

`salDiscoverSensor()` 函数调用成功后，我们就可以使用 `salGetNextDiscoveredSensor()` 来枚举发现的接收机的信息了。该函数的第一个参数是 `salDiscoverSensor()` 所输出的发现结果句柄，第二个参数是一个 `salSensorInfo` 结构体的变量，用来存放每次枚举的一台接收机的连接信息。

本例程的运行结果如图 4-23 所示。

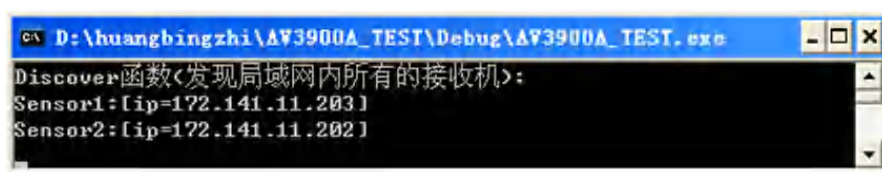


图 4-23 发现的接收机列表信息

在获取了所有网络内接收机的 IP 地址之后，我们可以参照“指定 IP 连接”一节中的方法来直接连接某台接收机。此外，我们还可以继续借助 SMS 提供的其它服务，更方便的操作接收机，例如下面提到的建立和管理一个接收机组。

将接收机纳入 SMS 管理

在进行 TDOA 测量时，您往往有这样的需求：只挑选网络内的三到四台接收机，进行同步测量，而且希望能够给接收机起一个便于记忆的名字，例如接收机所在的地点，而不是直接使用 IP 进行连接。SMS 为这种需求提供了很方便的服务，就是下面要介绍的 `salAddSensor()` 和 `salRemoveSensor()` 函数。

`salAddSensor()` 可以向 SMS 服务的管理组中添加一台接收机，在添加的同时也为该接收机设置了别名，`salRemoveSensor()` 可以从管理组中删除一台接收机。


```

int _tmain(int argc, _TCHAR* argv[])
{
    //定义一个代表SMS的句柄变量
    salHandle smsHandle;
    //打开SMS
    salOpenSms(&smsHandle, "172.141.11.45", 8089, "");
    if(smsHandle == NULL)
        return 0;
    salHandle sensorListHandle; //管理组接收机列表句柄
    salUInt32 num; //纳入管理的接收机数量
    //把发现的两个接收机纳入管理
    salAddSensor(smsHandle, "172.141.11.202", "KeYanLou");
    salAddSensor(smsHandle, "172.141.11.203", "JingJiaGong");
    salOpenSensorList2(&sensorListHandle, smsHandle, &num);
    cout << "添加了2个接收机后的SensorList中的Sensor数量: " << num << endl;
    //关闭发现列表句柄
    salClose(sensorListHandle);
    //从管理组中删除其中的一个接收机
    salErrorType error=salRemoveSensor(smsHandle, "KeYanLou");
    if(error != SAL_ERR_NONE)
        return 0;
    salOpenSensorList2(&sensorListHandle, smsHandle, &num);
    cout << "删除掉1个接收机后的SensorList中的Sensor数量: " << num << endl;
    if(sensorListHandle == NULL)
        return 0;
    salSensorStatus2 status;
    for(unsigned long i = 0; i < num; i++)
    {
        if(salGetNextSensor2(sensorListHandle, &status) == SAL_ERR_NONE)
        {
            cout << "Sensor" << i+1 << ":[ip=" << status.ipAddress
                << ",name=" << status.name << "]" << endl;
        }
    }
    //关闭句柄
    salClose(sensorListHandle);
    salClose(smsHandle);
    char readKey;
    cin >> readKey;
    return 0;
}

```

图 4-24 将接收机纳入和移除 SMS 管理

在图 4-24 的代码中,我们用到了四个新的 API,分别是 salAddSensor(), salRemoveSensor(), SalOpenSensorList2() 和 salGetnextSensor2()。其中 salAddSensor()向 SMS 添加一个接收机纳入管理组, salRemoveSensor()从 SMS 的管理组中删除一个受管理的接收机; SalOpenSensorList2()和 salGetnextSensor2()的作用与前一个例子中的发现和枚举网络内接收机的 API 类似,用来枚举当前管理组中的接收机。

在 main 函数的一开始,我们同样先打开 SMS 服务,然后使用 salAddSensor()把上一个例子中发现的两台接收机纳入 SMS 的管理。salAddSensor()需要两个参数,分别是接收机的 IP 地址和接收机的别名。在某台接收机使用 salAddSensor()纳入 SMS 的管理之后,我们在程序中就可以使用接收机的别名来标识这台接收机了。

在添加完两个接收机后,我们使用 SalOpenSensorList2()验证 SMS 中纳入管理的接收机的数量,然后在使用 salRemoveSensor()来删除别名为“KeYanLou”的接收机,再次调用 SalOpenSensorList2()查看删除一台接收机后的管理组中接收机的数量,最后使用 salGetNextSensor()来枚举管理组中的接收机。该例程的运行结果如图 4-25 所示。



图 4-25 运行结果

当我们把某台接收机纳入管理之后，salConnectSensor2()函数就有了第二种使用方式，即在参数中提供 SMS 服务句柄和接收机别名。还有其他一些 API 可以使用接收机的别名操作，具体可参见编程手册。

■ salOpenSms

函数声明：

```
salErrorType salOpenSms (salHandle* smsHandle,
                           char* hostname,
                           salUInt16 port,
                           char* directory)
```

函数功能：

建立应用程序与 SMS 服务的连接。

参数列表：

参数类型	参数名	参数说明
salHandle*	smsHandle	[out]指向保存 SMS handle 的指针
char*	hostname	[in]SMS 服务的主机 IP 地址。
salUInt16	port	[in]使用的端口号 (默认端口号 8089)
char*	directory	[in]SMS 在服务器上的地址 (可选的参数，为 NULL 时使用默认的数值)

salDiscoverSensors

函数声明：

```
salErrorType salDiscoverSensors (salHandle* discoveryHandle,
                                   salHandle smsHandle,
                                   salUInt32* numSensors)
```

函数功能：

发现网络内的接收机。

参数列表：

参数类型	参数名	参数说明
salHandle*	discoveryHandle	[out]指向发现接收机列表句柄的指针
salHandle	smsHandle	[in]salOpenSms() 返回的 handle
salUInt32*	numSensors	[out]发现的接收机数量

■ salGetNextDiscoveredSensor

函数声明:

```
salErrorType salGetNextDiscoveredSensor(salHandle discoveryHandle,
                                          salSensorInfo* sensorInfo)
```

函数功能:

函数取得 salDiscoverSensors() 打开的接收机列表中的下一个接收机信息。

参数列表:

参数类型	参数名	参数说明
salHandle	discoveryHandle	[in]salDiscoverSensors() 返回的 handle
salSensorInfo*	sensorInfo	[out]指向接收机状态信息结构体的指针

■ salAddSensor

函数声明:

```
salErrorType salAddSensor(salHandle smsHandle,
                           char* hostname,
                           char* sensorName)
```

函数功能:

向 SMS 的管理组添加一个接收机。

参数列表:

参数类型	参数名	参数说明
salHandle	smsHandle	[in]由 salOpenSms() 得到的 SMS handle
char*	hostname	[in]接收机的 IP 地址
char*	sensorName	[in]接收机的别名

■ salRemoveSensor

函数声明:

```
salErrorType salRemoveSensor(salHandle smsHandle,
                               char* sensorName)
```

函数功能:

从 SMS 的管理组移除一个接收机。

参数列表:

参数类型	参数名	参数说明
salHandle	smsHandle	[in]salOpenSms() 返回的 handle
char*	sensorName	[in]要移除接收机的别名。

■ salOpenSensorList2

函数声明:

```
salErrorType salOpenSensorList2(salHandle* sensorList,
                                salHandle smsHandle,
                                salUInt32* numSensors)
```

函数功能:

查询当前 SMS 管理的所有接收机状态。

参数列表:

参数类型	参数名	参数说明
salHandle*	sensorList	[out] 指向接收机列表句柄的指针
salHandle	smsHandle	[in] salOpenSms() 返回的 handle
salUInt32*	numSensors	[out] 指向保存接收机数量变量的指针

■ salGetNextSensor2

函数声明:

```
salErrorType salGetNextSensor2(salHandle sensorList,
                                salSensorStatus2* sensorStatus)
```

函数功能:

函数取得 salOpenSensorList2() 打开的接收机列表中的下一个接收机状态。

参数列表:

参数类型	参数名	参数说明
salHandle	sensorList	[in] salOpenSensorList() 返回的接收机状态列表 handle
salSensorStatus2*	sensorStatus	[out] 指向保存接收机状态信息结构体的指针

频谱扫描例程

在尝试了连接功能后，我们可以继续在这个基础上加入新的控制功能，例如演示软件的频率扫描功能。用于频谱扫描的API函数是salStartSweep()和salGetSegmentData()。这两个函数位于“salFrequency.h”中，要使用它们，我们首先需要包含该头文件。下面我们就借助这两个函数，对第一个例子进行扩展，使其能够完成以下扫描：

- ◆ 扫描范围：100MHz~200MHz
- ◆ 帧长度：1024点FFT
- ◆ 采样率：28MHz
- ◆ 扫描次数：3次

我们要求程序每次扫描结束后打印出各个频段的起始频率，采集时刻和前25个频点的幅度。该程序的源代码如图4-26、图4-27、图4-28所示。

```

int _tmain(int argc, _TCHAR* argv[])
{
    //定义一个代表接收机的句柄变量
    salHandle sensorHandle;
    //连接接收机
    salErrorType error = salConnectSensor2(&sensorHandle, NULL, "172.141.11.202", "", 0);
    if(error == SAL_ERR_NONE)
    {
        cout << "已连接\n";
    }
    else
    {
        cout << "连接失败\n";
        //连接失败退出
        return 0;
    }
    salHandle measHandle; //测量任务句柄
    salSweepParms sp; //本次扫描的通用参数
    salFrequencySegment *pSegmentTable; //指向频段参数数组的指针
    double centerFrequency; //当前扫描频段的中心频率
    double span; //频率跨度
    double sampleRate; //采样率
    int numPoints; //每帧FFT的有效点数
    int firstPoint; //首个有效点索引
    float amplitudeData[1024]; //存放测量结果的缓冲区
    salSegmentData dataHdr; //接收机回送数据的描述头
}

```

图 4-26 频谱扫描所用变量

```

//设置通用参数
memset(&sp, 0, sizeof(sp));
sp.numSweeps = 3; //扫3遍
sp.numSegments = 5; //每遍扫5个频段

//申请频段参数数组的内存空间
pSegmentTable = (salFrequencySegment*)malloc(sp.numSegments * sizeof(salFrequencySegment));
memset(pSegmentTable, 0, sp.numSegments * sizeof(salFrequencySegment));

span = 20e6;
sampleRate = 28e6;
centerFrequency = 100e6 + (span / 2); //初始频段的中心频率
numPoints = static_cast<int>(1024 * span / sampleRate); //计算有效点数
firstPoint = (1024 - numPoints) / 2; //计算首个有效点索引
//依次设置每个频段的参数
for (int i = 0; i < sp.numSegments; i++)
{
    pSegmentTable[i].numFftPoints = 1024;
    pSegmentTable[i].numPoints = numPoints;
    pSegmentTable[i].firstPoint = firstPoint;
    pSegmentTable[i].centerFrequency = centerFrequency;
    pSegmentTable[i].sampleRate = sampleRate;
    centerFrequency += span; //将centerFrequency更新为下一个频段的中心频率
}

//启动频域测量
error = salStartSweep(&measHandle, sensorHandle, &sp, pSegmentTable, NULL);
if (error != SAL_ERR_NONE)
{
    salClose(sensorHandle);
    free(pSegmentTable);
    cout << "启动测量失败\n";
    return 0;
}

```

图 4-27 设置变量并启动测量

```

//获取测量结果
salUInt32 sweepCounter = 0;
do
{
    // 获取一段测量数据
    error = salGetSegmentData(measHandle, &dataHdr, amplitudeData, sizeof(amplitudeData));
    if (error != SAL_ERR_NONE)
    {
        cout << "获取数据失败\n";
        break;
    }
    else
    {
        //打印数据结果
        cout << "起始频率" << dataHdr.startFrequency << " 采集时刻"
              << dataHdr.timestampSec << "秒" << dataHdr.timestampNSec << "纳秒" << endl;
        for(int i = 0; i < 5; i++)
        {
            for(int j = 0; j < 5; j++)
            {
                cout << amplitudeData[i * 5 + j] << "dBm ";
            }
            cout << endl;
        }
        cout << endl;
    }
    if (dataHdr.segmentIndex >= sp.numSegments - 1)
    {
        sweepCounter++;
    }
} while (sweepCounter < sp.numSweeps);
// 关闭测量handle
salClose(measHandle);
// 关闭传感器handle
salClose(sensorHandle);
// 释放内存
free(pSegmentTable);
char input;
cin >> input;
return 0;
}

```

图 4-28 读取测量结果

首先来看图4-26中的代码，前半部分与第一个例子基本一样，只是在正常连接时不再关闭句柄，因为程序后面要用到这个句柄。如果连接失败时就直接退出程序。后半部分定义了一些新的变量，其中新面孔salSweepParms、salFrequencySegment和salSegmentData结构体类型均定义在salFrequency.h头文件中。所有这些新加入的变量都是为了频谱扫描的那两个API的参数所准备的。

- measHandle是一个测量任务句柄，代表这次频谱扫描的测量任务；
- sp是一个salSweepParms类型的结构体，它记录了本次扫描用到的一些参数，在本例中，我们只需要填入扫描段数和扫描次数字段，其它字段设为0即可；
- pSegmentTable是一个段参数结构体的指针，在本例中它指向了一个动态申请的内存区域，该内存区域稍后会被初始化成一个段参数的数组，至于段参数的含义及计算方法随后会有说明；
- centerFrequency保存当前段的中心频率，在后面的迭代中，该值会每次递增一个扫描宽度并依次填入到每个扫描段的段参数中；
- span是本次扫描子频段的有效带宽；
- sampleRate是采样率；
- numPoints和firstPoints是每一帧FFT要取的有效点数和首个有效点的索引；
- amplitudeData是一个浮点数组，用来存放测量结果。

有了这些变量，我们就可以按照我们的测量要求，把数据对应着填入变量中，如图4-27

所示。

在此要说明一下为什么从100MHz扫到200MHz要设置5段的段扫描参数，以及段扫描参数设置成图中所示的值背后的原理。

对于接收机的硬件电路而言，它一次只能执行一帧FFT的数据采集和计算。计算完毕后需要接收机的CPU为它设置下一次的扫描参数并再次启动。而每一帧所覆盖的有效频率范围是有一定限度的（最大相当于接收机的分析带宽20MHz），所以如果要扫描一个较大的频率区间，接收机实际上是通过扫描一系列的子频段然后拼接而成的，如图4-29所示。

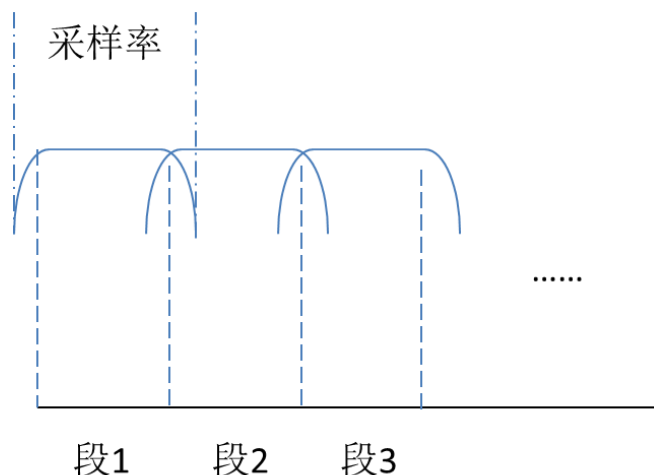


图 4-29 频段的拼接

为了防止混叠，每个频段的有效带宽通常小于设置的采样率，在28MHz采样率下，有效带宽是中间的20MHz，其余部分的测量数据是不准确的。这意味着在扫描时，我们应按照有效带宽来拼接子频段，并告知接收机只取FFT计算结果中的有效数据。

在本例中，我们要求接收机在28MHz采样率下从100MHz扫到200MHz，从以上说明我们得出，接收机扫100MHz的频率宽度，是通过扫5个20MHz拼接而成的。第一段的中心频率是 $(100\text{MHz} + 20\text{MHz}/2)$ ，第二段的中心频率是第一段的中心频率加20MHz，以此类推。每一段的有效数据点数等于：

$$\text{FFT点数} \times \frac{\text{有效带宽}}{\text{采样率}}$$

有效点对半分布在子频段中心频率的左右两侧，无效数据分布在子频段的两端。

配置完这些参数后，就可以把参数发送给接收机并启动测量了。我们调用salStartSweep()来为接收机设置扫描参数并启动本次测量，如果接收机正确响应，就会随即启动扫描，接收机的内部CPU会把段参数依次配置给扫描硬件。在完成一遍扫描之后，接收机启动网络传输功能把本次扫描每一子频段的数据打包发送到程控主机端。我们通过图4-28中的代码来获取和观察这些测量数据。

图4-28中的代码主体是一个do-while循环，在循环中我们调用salGetSegmentData()来获取每一子频段的测量结果并打印之。salGetSegmentData()函数每次返回一个子频段的测量数据和该段的信息描述。测量数据放在用户申请的缓冲区中，信息描述放在一个salSegmentData的结构体中，信息包括了接收机扫描该段时的状态如频段索引、扫描次数、起始频率、采集起始时刻等。我们可以使用cout对象打印这些测量结果。在采集完成后，我们应关闭句柄并释放动态申请的内存，然后退出。

在本例中，我们使用一个嵌套循环来打印每段的前25个幅度数据，采用5行5列的形式，该程序的运行结果如图4-30所示。

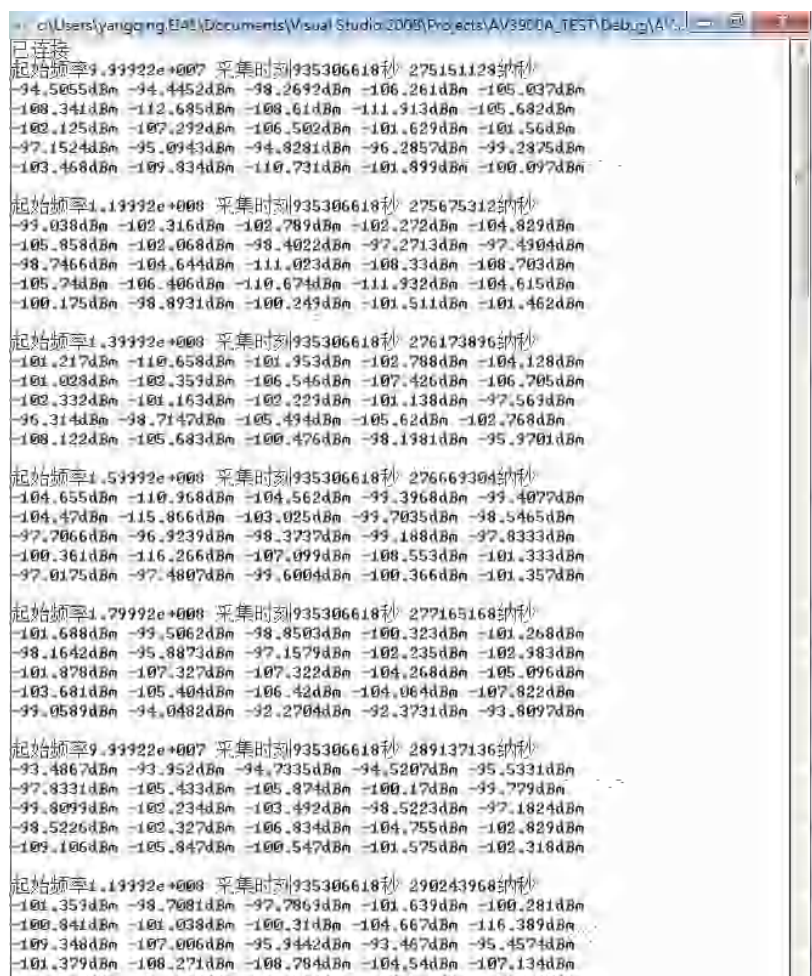


图 4-30 频谱扫描测量结果

■ salStartSweep

函数声明：

```
SAL_API salStartSweep(salHandle *measHandle,
                       salSensorHandle sensorHandle,
                       salSweepParms *parms,
                       salFrequencySegment *pSegmentTable,
                       SAL_SEGMENT_CALLBACK dataCallback)
```

函数功能：

启动频谱扫描。

参数列表：

参数类型	参数名	参数说明
salHandle*	measHandle	[out] 本命令创建的测量 handle
salSensorHandle	sensorHandle	[in] salConnectSensor2() 返回的接收机 handle
salSweepParms*	parms	[in] 扫描参数定义
salFrequencySegment*	pSegmentTable	[in] 频谱扫描段参数表

SAL_SEGMENT_CALLBACK dataCallback [in] 设置为 NULL

备注:

measHandle 是输出参数, 数据获取函数可以利用这个参数来取得测量结果; sensorHandle 就是 salConnectSensor2() 的输出参数; parms 是指向一个配置本次扫描的参数结构体的指针, 通常情况下, 除了扫描段数和扫描次数需要设置外, 该结构体的其它字段都可以默认为零; pSegmentTable 是一个结构体的数组, 结构体记录了扫描其中一段的配置参数。

■ salGetSegmentData

函数声明:

```
SAL_API salGetSegmentData (salHandle measHandle,
                             salSegmentData *dataHdr,
                             salFloat32 *pAmplitude,
                             salUInt32 userDataBufferBytes)
```

函数功能:

调用此函数请求获取扫描测量的一段数据结果。调用函数之前, 需要先分配内存以便存储数据结果。

参数列表:

参数类型	参数名	参数说明
salHandle	measHandle	[in] salStartSweep() 取得的测量 handle
salSegmentData*	dataHdr	[out] salSegmentData 信息头
salFloat32*	pAmplitude	[out] 用户分配的存储空间保存测量幅度数据
salUInt32	userDataBufferBytes	[in] 用户分配存储空间大小

备注:

调用该函数一次, 会得到一个扫描段的测量结果, 通常情况下, 需要在一个循环中调用该函数, 以便得到所有扫描段的测量结果。measHandle 是 salStartSweep() 的输出参数; dataHdr 记录了该段测量结果的状态数据; pAmplitude 指向用户申请的数据缓冲区; userDataBufferBytes 标明 pAmplitude 所占的字节数; 测试数据正常时, 函数返回 SAL_ERR_NONE。其他返回值, 表示测试不正常。

注意

当 FFT 点数过多时, 不宜使用局部数组进行数据获取和处理, 以免造成堆栈溢出, 应使用动态分配的内存或静态数组。

时域采集例程

采集 IQ 序列

除了频谱数据，接收机还可以输出IQ序列，我们通过下面的例程，看接收机如何获取1GHz频点上的IQ数据。

◆ IQ序列长度：512

◆ 采样率：14MHz

我们要求程序每次扫描结束后打印前50点的IQ值。该程序的源代码如图4-31、图4-32所示。

```
int _tmain(int argc, _TCHAR* argv[])
{
    //定义一个代表接收机的句柄变量
    salHandle sensorHandle;
    //连接接收机
    salErrorType error = salConnectSensor2(&sensorHandle, NULL, "172.141.11.202", "", 0);
    if(error == SAL_ERR_NONE)
    {
        cout << "已连接\n";
    }
    else
    {
        cout << "连接失败\n";
        //连接失败退出
        return 0;
    }

    salHandle measHandle;           //测量任务句柄
    salTunerParms tunerParms;       //调谐电路参数
    salTimeDataParms timeDataParms; //数字中频电路参数
    salTimeData dataHdr;           //信息头
    salInt32 iqData[1024];         //结果存放缓冲区，每个IQ点占2个salInt32

    //调谐电路参数
    memset(&tunerParms, 0, sizeof(tunerParms));
    tunerParms.centerFrequency = 1e9;
    tunerParms.sampleRate = 14e6;
    //数字中频电路参数
    memset(&timeDataParms, 0, sizeof(timeDataParms));
    timeDataParms.dataType = salCOMPLEX_32;
    timeDataParms.numSamples = 512;
    timeDataParms.numTransferSamples = 512;
}
```

图 4-31 时域采集所用变量

```

//设置接收机调谐参数
salSetTuner(sensorHandle, &tunerParms);
//启动时域数据采集
error = salRequestTimeData(&measHandle, sensorHandle, &timeDataParms, NULL);
if (error != SAL_ERR_NONE)
{
    cout << "启动测量失败\n";
    salClose(sensorHandle);
    return 0;
}
error = salGetTimeData(measHandle, &dataHdr, iqData, sizeof(iqData));
if (error == SAL_ERR_NONE)
{
    //打印信息头
    cout << "中心频率" << dataHdr.centerFrequency << endl;
    cout << "采集时刻" << dataHdr.timestampSeconds
        << "秒" << dataHdr.timestampNSecods << "纳秒" << endl;
    cout << "I" << "\t\t" << "Q" << endl;
    //字符串缓冲区
    char buf[100];
    //打印IQ数据
    for(int i = 0; i < 50; i++)
    {
        sprintf(buf, "0x%.8x\t0x%.8x\n", iqData[i * 2], iqData[i * 2 + 1]);
        cout << buf << endl;
    }
}
else
{
    cout << "获取数据失败\n";
}
// 关闭测量handle
salClose(measHandle);
// 关闭传感器handle
salClose(sensorHandle);
char input;
cin >> input;
return 0;
}

```

图 4-32 启动采集并打印结果

在图4-31中，连接接收机的部分与前面的例子一样，连接成功后定义了时域数据采集要用到的变量，其中salTunerParms、salTimeDataParms、salTimeData结构体类型均定义在sal-TimeData.h头文件中。

- measHandle是一个测量任务句柄，代表这次时域采集的测量任务；
- tunerParms是一个salTunerParms类型的结构体，它记录了本次测量调谐电路的设置参数；
- timeDataParms是一个salTimeDataParms类型的结构体，它记录了本次测量数字中频电路的设置参数；
- dataHdr是一个salTimeData类型的结构体，它记录了接收机回发数据的信息描述；
- iqData是一个整型数组，用来存放测量结果。IQ序列每一点是一个复数对，所以数组中使用相邻的两个位置记录一个IQ序列点。

配置完这些参数后，就可以把参数发送给接收机并启动测量了。

在图4-32中，我们先调用salSetTuner()来为接收机设置调谐参数，然后调用salRequestTimeData()请求获取时域数据。如果接收机正确响应，就可以使用salGetTimeData()获得测量结果了。测量结果的信息描述部分放在dataHdr中，数据放在iqData数组中，数组中的每一对数分别代表一个IQ测量点的实部和虚部，其类型均是有符号的整数，采用补码的编码形式。在采集完成后，我们应关闭句柄，然后退出。

在本例中，我们使用一个循环打印前 50 对 IQ 数据，其测量结果如图 4-33 所示：



图 4-33 时域采集的测量结果

■ salSetTuner

函数声明：

```
SAL_API salSetTuner(salSensorHandle sensorHandle,  
                    salTunerParms *parmsIn)
```

函数功能：

设置接收机的调谐电路参数。

参数列表：

参数类型	参数名	参数说明
salSensorHandle	sensorHandle	[in]salConnectSensor() 返回的 handle

`salTunerParms*` `parmsIn` `[in]` 调谐和输入设置参数

备注:

此函数用来设置调谐电路参数，包括中心频率、采样率、天线选择、衰减、前置放大器等。

■ `salRequestTimeData`

函数声明:

```
SAL_API salRequestTimeData(salHandle *measHandle,
                             salSensorHandle sensorHandle,
                             salTimeDataParms *parms)
```

函数功能:

向接收机请求获取时域数据。

参数列表:

参数类型	参数名	参数说明
<code>salHandle*</code>	<code>measHandle</code>	<code>[out]</code> 创建的测量 handle
<code>salSensorHandle</code>	<code>sensorHandle</code>	<code>[in]</code> <code>salConnectSensor()</code> 返回的接收机 handle
<code>salTimeDataParms*</code>	<code>parms</code>	<code>[in]</code> 请求测量定义的参数

备注:

该函数用来发起一次时域数据的采集。在调用该函数之前，应该配置好 `parms` 参数使用 `salGetTimeData()` 来获取测量结果。

■ `salGetTimeData`

函数声明:

```
SAL_API salGetTimeData(salHandle measHandle,
                          salTimeData *dataHdr,
                          void *userDataBuffer,
                          salUInt32 userDataBufferBytes)
```

函数功能:

获取时域测量结果。

参数列表:

参数类型	参数名	参数说明
<code>salHandle</code>	<code>measHandle</code>	<code>[in]</code> <code>salRequestTimeData()</code> 返回的测量 handle
<code>salTimeData*</code>	<code>dataHdr</code>	<code>[out]</code> <code>salTimeData</code> 数据头
<code>void*</code>	<code>userDataBuffer</code>	<code>[out]</code> 用户分配的内存空间保存测试数据
<code>salUInt32</code>	<code>userDataBufferBytes</code>	<code>[in]</code> <code>userDataBuffer</code> 所占字节数

备注:

salGetTimeData()用来获取测试数据。为了更好的实现性能,您应当一直读取测试数据,直到 stateEventIndicator 的数据项设置为 sal_STATE_LAST_BLOCK。如果您想终止数据项的传输,最好使用 salSendTimeDataCommand()停止或取消数据流。

第二篇 技术说明

第五章 工作原理

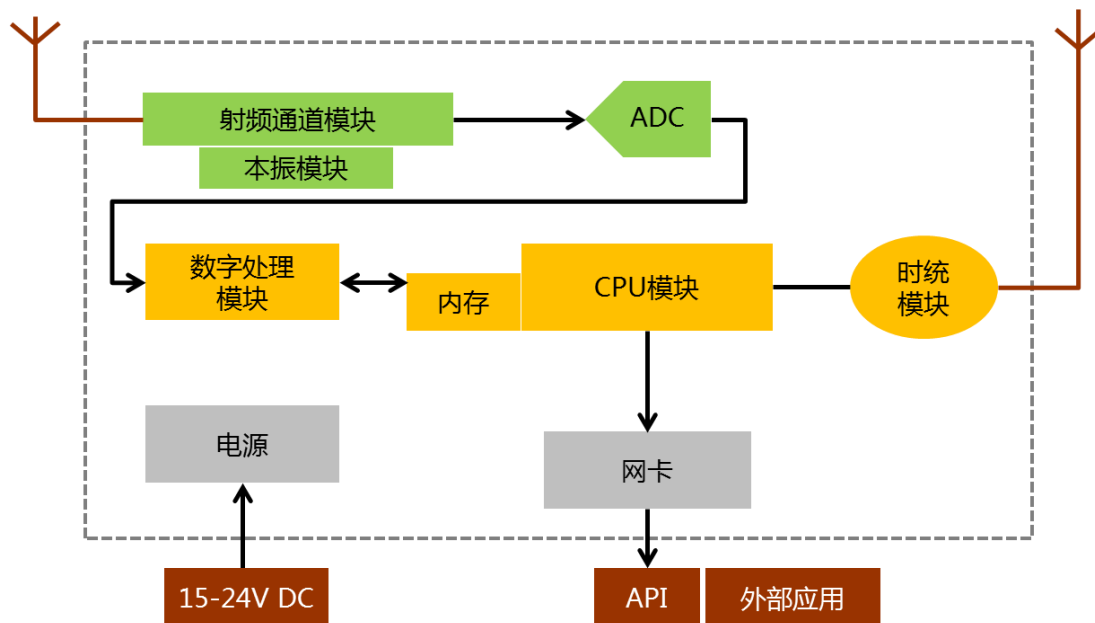


图 5-1 整机原理框图

接收机的内部模块组成如图 5-1 所示。我们可以跟踪信号在接收机内部的流向来理解接收机的工作原理。从接收天线接收到的无线电信号，经过天线电缆输入到射频处理板，经过信号预选、混频、滤波处理后，生成模拟中频信号；该信号经过 A/D 转换、数字下变频处理后生成 I/Q 时间序列的数字化数据，该数据可直接输出给 CPU，也可通过 FPGA 的 FFT 变换形成频谱数据再输出给 CPU；CPU 读取 FPGA 的处理结果，进行一些数据的预处理，然后根据通信协议将数据打包发送给远程的服务器。

射频处理板主要是对接收天线输入的射频信号进行处理，经过信号预选、混频、滤波、A/D 采样输出数字中频信号。本振模块为下变频器提供本振，通过扫描本振，实现整机对设定范围内信号的获取处理。变频通道部分实现信号下变频，并通过滤波器，实现镜频抑制、杂散抑制、中频抑制。ADC 部分对模拟中频采样，为中频数字板提供数字中频信号。

中频数字电路主要是对数字中频信号进行数字下变频、FFT 变换、数据缓存等处理。射频输入信号经过微波变频模块下变频到中频后，经由 AD 转换为数字中频信号送入 FPGA，在 FPGA 内部首先对中频信号进行数字下变频，得到零中频信号，同时将信号分解为同相分量 I 和正交分量 Q 两路，IQ 信号随后经过由 CIC 和 FIR 组成的滤波器组进行抽取滤波，以满足不同分辨率带宽测试需求。在 FFT 频谱分析模式下，经过滤波器组调整采样频率后的 IQ 数据按帧送入 FFT 运算模块进行 FFT 变换，FFT 输出的幅度信息经过峰值、取样、均值等方式的检波后，存入内存模块以备 CPU 读取。

CPU 及其上层软件用于实现整机的协调控制，主要包括扫描控制、数据读取、远程交互、时间同步、整机状态监控等功能。

第六章 性能特性测试



声明：

下列各个指标测试时的具体操作步骤是根据图示中的测试仪器编写的，当采用同等性能特性的其它测试仪器时，具体操作方法应参照该仪器的使用说明书进行。测试步骤中提到的复位仪器，均指厂家复位模式，如设备处于用户定义复位状态，应改为厂家复位状态并进行再次复位，以保证设备初始状态处于已知状态。

表6-1. 测试用仪器设备

序号	仪器名称	主要技术指标	推荐型号
1	合成信号发生器	频率范围：20MHz~6GHz 功率输出：-70dBm ~+15dBm 频率准确度：±0.02% 功率电平可校准、存储	AV1461 或 E4438
2	功率计	功率范围：-70dBm~+44dBm 校准源频率：50MHz 校准源幅度：0dBm	E4419B 或 AV2434
3	功率探头	频率范围：20MHz~ 6GHz 功率范围：-30dBm~ +20dBm	Agilent8487A 或 AV71711
4	矢量网络分析仪	频率范围:20MHz ~ 6GHz	AV3656B
5	射频定向耦合器	频率范围：20MHz ~ 6GHz 方向性：35dB VSWR：<1.45	自制
6	低通滤波器	截止频率：1.0GHz 插入损耗：<0.9dB 带外抑制：>65dB	SLP-1200
7	低通滤波器	截止频率：2.0GHz 插入损耗：<1.5dB 带外抑制：>65dB	
8	负载	阻抗：50Ω	HP85054-60046
9	转接器	3.5mm(f)到 3.5mm(f)	自制
10	转接器	3.5mm(m)到 3.5mm(m)	自制
11	转接器	3.5mm(m)到 N(f)	自制
12	转接器	3.5mm(f)到 N(m)	自制
13	转接器	3.5mm(f)到 N(f)	自制
14	转接器	N(m)到 N(m)	自制
15	转接器	BNC(m)到 N(f)	自制
16	转接器	BNC(m)到 BNC(m)	自制
17	计算机	Win XP 或 Win 7 平台	

频率范围

分别输入频率为 20MHz 和 6GHz 的信号来测试无线电监测接收机的频率测量范围。由

于整机与外部不能共时基，综合多种影响因素，测试结果在 $\pm 2\text{ppm}$ 以内满足要求。

- a) 测试设备
 - 合成信号发生器 AV1461
 - 外控计算机 联想电脑
- b) 转接器
 - 3.5mm(f)到N(f)转接器 1个
- c) 电缆
 - N(m-m)电缆 1根

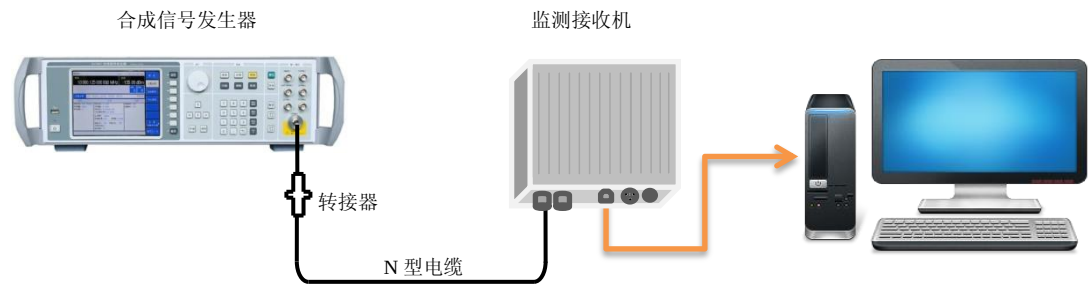


图6-1. 频率范围测试

- d) 测试步骤
 - 1) 按照上图连接测试设备，合成信号发生器的输出端口接到被测接收机的信号输入端 1 口；
 - 2) 设置合成信号发生器的输出频率为 20MHz，幅度为-10dBm，射频输出打开；
 - 3) 设置被测接收机的中心频率为 20MHz，扫宽 10kHz，参考电平 0dBm，分辨率带宽等自动。在显示界面上打开光标菜单选峰值，读出峰值光标的频率值并记录；
 - 4) 设置合成信号发生器的输出频率改为被测接收机最大频率，其他不变；
 - 5) 设置被测接收机的中心频率为其最大频率，扫宽 100kHz，参考电平 0dBm，分辨率带宽等自动。在显示界面上打开光标菜单选峰值，读出峰值光标的频率值并记录；
 - 6) 测量结果和合成信号发生器设置的频率相比较，其差值应小于信号发生器和被测接收机内部时基允许的最大偏差。将比较结果记录在“AV3900A 无线电监测接收机性能测试记录表”（以下简称“测试记录附表”）对应测试项中，比较结果相符记录为“√”，比较结果不相符记录为“×”。

边带噪声

从偏离载波 10kHz、100kHz 处测量 1.0GHz、0dBm 信号的边带噪声，打开平均功能对每个频率偏离点上的边带噪声进行平均。如果在设定频偏处有寄生响应，应该将光标偏离寄生响应，保证测量的准确度。

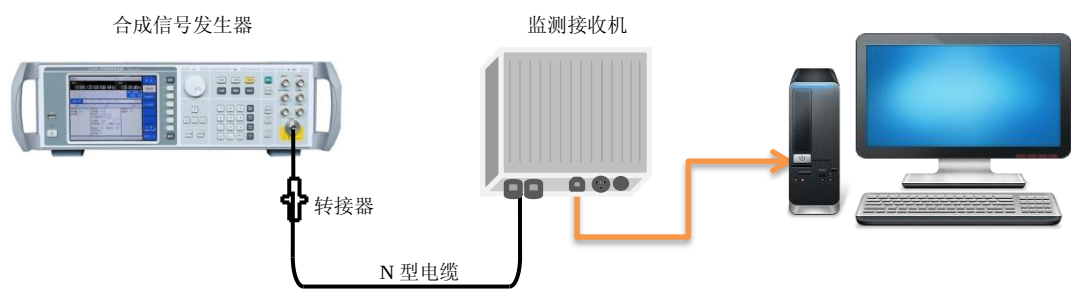


图6-2. 边带噪声测试

- a) 测试设备
- 合成信号发生器AV1461
 - 外控计算机 联想电脑
- b) 转接器
- 3.5mm(f)到N (f) 转接器 1个
- c) 电缆
- N(m-m)电缆 1根
- d) 测试步骤
- 1) 如图连接仪器;
 - 2) 设置信号发生器输出频率为 1GHz, 输出功率为 0dBm;
 - 3) 在外控软件上设置[中心频率] 1GHz, [扫宽] 30kHz, [参考电平] 0dBm, FFT 点数 8192, 采样率 54.6875kHz;
 - 4) 在频谱显示界面上打开平均功能, 并设置平均次数为 10;
 - 5) 在频谱显示界面上打开光标, 搜索峰值, 打开差值功能, 然后等待平均 10 次;
 - 6) 在频谱显示界面上设置光标频差为+10kHz。在附录测试记录表对应测试项中测量值栏记录差值光标幅度值, 作为+10kHz 偏离处的边带噪声;
 - 7) 在频谱显示界面上设置光标频差为-10kHz。在附录测试记录表对应测试项中测量值栏记录差值光标幅度值, 作为-10kHz 偏离处的边带噪声;
 - 8) 在外控软件上设置[扫宽] 300kHz, 重复步骤 4-7, 测量频偏+100kHz 和-100kHz 处的边带噪声;
 - 9) 按照附录测试记录表对应测试项中其他测量频率点依次设置接收机, 重复上述步骤, 将测量结果记录在附录测试记录表对应测试项中。

显示平均噪声电平

测量无线电监测接收机的显示平均噪声电平。接收机的输入端接 50Ω匹配器。用光标来定位具有最大响应的频率点, 然后读出窄扫宽下的平均噪声。如果在设定光标处有剩余响应, 应该将光标偏离剩余响应, 保证测量的准确度。噪声光标读数被归一化为 1Hz 噪声带宽, 用来对选取频率处的峰值进行平均和归一化修正。

- a) 测试设备
- 匹配器 50Ω匹配负载

外控计算机 联想电脑

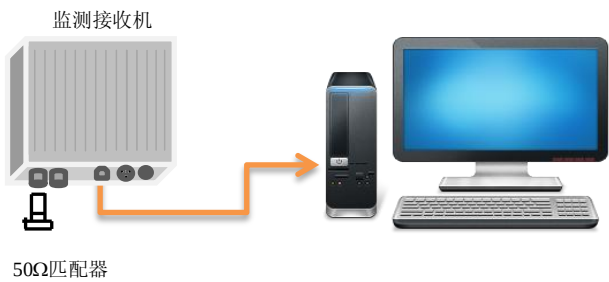


图6-3. 显示平均噪声电平测试

- b) 测试步骤
- 1) 按图所示把 50Ω匹配器连在被测接收机的输入端口上;
 - 2) 如下设置被测接收机: 前放关闭; 参考电平-60dBm; 所有光标关; 采样率 28MHz; FFT 点数 1024;
 - 3) 设置被测接收机起始频率 20MHz,终止频率 2.7GHz;
 - 4) 找到噪声电平最大值对应的频点, 并将其设为中心频率;
 - 5) 逐渐减小扫宽设置, 适当修正中心频率设置, 保证噪声峰值频点落在当前测量频率范围内, 直到扫宽为 5kHz, 并将参考电平设为-80dBm, 采样率设为 6.835kHz, FFT 点数设为 1024;
 - 6) 设置平均次数为 20, 并等待测量完成;
 - 7) 按【光标】、【峰值】, 打开噪声光标功能, 读出当前值, 作为前放关闭时该频段的显示平均噪声电平, 并记录在测试记录表对应测试项中;
 - 8) 按指标要求, 中频段划分重新设置被测接收机的起始频率和终止频率, 重复 4-7 步;
 - 9) 将被测接收机的前放打开, 按照指标要求, 中频段划分重新设置被测接收机的起始频率和终止频率, 重复 4-7 步。注意: 测试过程中根据需要可适当减小参考电平以保证测量结果曲线能正常显示在屏幕上。

二次谐波失真

合成信号发生器经低通滤波器为接收机测量二次谐波失真提供信号。低通滤波器消除来自于信号源的本身携带的高次谐波失真。

- a) 测试设备
- 合成信号发生器AV1461
 - 外控计算机 联想电脑
 - 1.0GHz低通滤波器 1个
- b) 转接器
- 3.5mm(f)到N(f)转接器 1个
 - 3.5mm(m)到N(f)转接器..... 1个
- c) 电缆
- N(m-m)电缆 1根

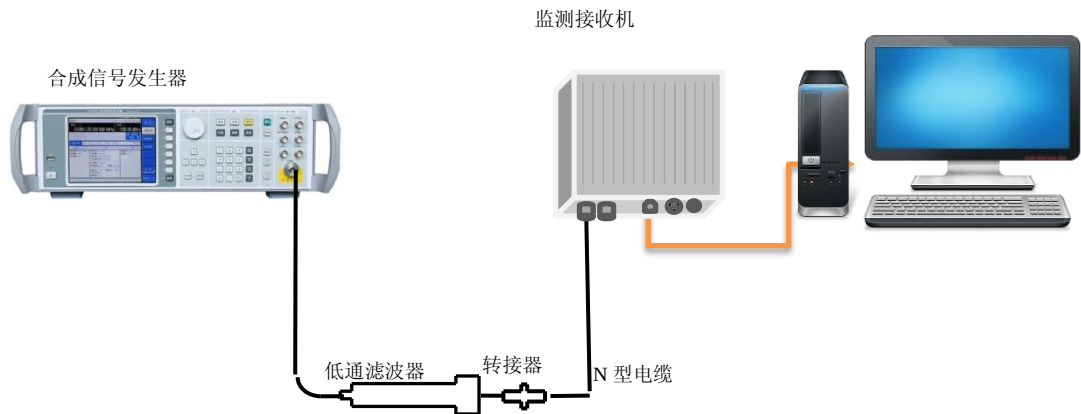


图6-4. 二次谐波失真测试

- d) 测试步骤
- 1) 用 1.0GHz 低通滤波器、射频转接器和射频电缆如图连接仪器，若信号发生器本身输出功率-30dBm 时二次谐波指标优于-60dB，可不连接低通滤波器；
 - 2) 将 AV1461 设置频率为 1000MHz，幅度为-30dBm，射频输出打开；
 - 3) 如下设置接收机：中心频率 1000MHz，扫宽 100kHz，参考电平-30dBm；
 - 4) 在外控软件上按【峰值】，调整 AV1461 功率电平使光标读数为-30dBm±0.1dB；
 - 5) 按【光标】[Δ 模式]，设置中心频率为 2.0GHz；
 - 6) 在完成一次新扫描后按【峰值】，把差值光标读数记录在附录测试记录表对应测试项中作为二次谐波失真值；
 - 7) 按照附录测试频点更换相应的测试附件并重复上述步骤，直到所有频点测试完成并记录在对应测试表格内。

三阶交调失真

两台合成信号发生器为测量三阶交调失真提供所需的信号。滤波器用来滤除最接近正在测量的信号失真产物。

注：如信号发生器#1 和信号发生器#2 的输出受负载牵引影响测试结果时，可在信号发生器#1、#2 的输出端各加一对应测量频段的隔离器，以减小对三阶交调失真测试结果的影响。

由于在混频器电平为-30dBm 时对应的交调产物幅度低，为免受接收机本底噪声的影响，在指标测试中测量混频器电平为-20dBm 的失真产物，然后修正 20dB（输入混频器电平-20dBm 的失真产物比在输入-30dBm 失真产物将比测量的失真产物高 20dB）。

- a) 测试设备
- | | |
|-------------------|---------|
| 功率计 | AV2434 |
| 功率探头 | AV71711 |
| 合成信号发生器#1 | AV1461 |
| 合成信号发生器#2 | AV1461 |
| 外控计算机 | 联想电脑 |
| 射频定向耦合器 | 1个 |
| 1.0GHz低通滤波器 | 2个 |
- b) 转接器
- | | |
|------------------------|----|
| 3.5mm(m)到N(m)转接器 | 1个 |
| N(f)到N(f)转接器 | 1个 |

	BNC(f)到N(m)转接器.....	1个
c) 电缆	BNC(m-m)电缆.....	1根
	3.5mm(f-m)电缆.....	2根
	3.5mm(m-m)电缆.....	2根

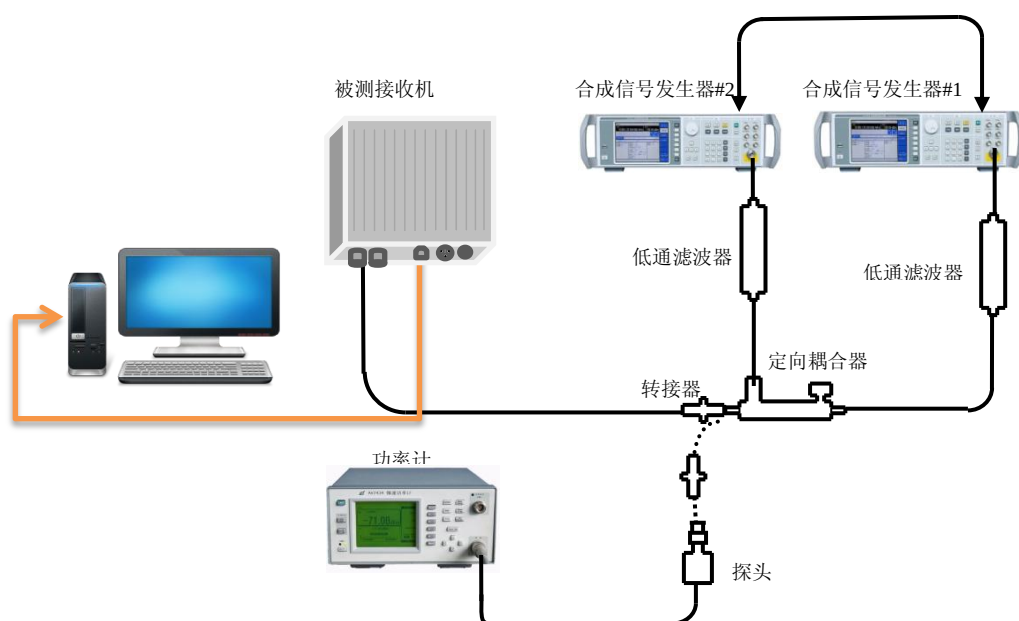


图6-5.三阶交调失真测试

d) 测试步骤

- 1) 如图连接仪器，但不把定向耦合器接到接收机上；
- 2) 设置合成信号发生器#1 AV1461 的频率为 901MHz，幅度为-9dBm；
- 3) 将合成信号发生器#2 AV1461 复位后，设置频率为 901.1MHz，功率为-110dBm，射频输出关闭，外参考；
- 4) 连接 AV2434 与 AV71711 并校零。设置 **【FREQ】** 901MHz；
- 5) 通过外控软件设置被测接收机：中心频率 901.05MHz，扫宽 500kHz，采样率 875kHz，FFT 点数 8192，参考电平 0dBm，衰减 40dB（射频衰减 10dB），前放关闭；
- 6) 把功率探头连接到射频定向耦合器的输出，调整#1 合成信号发生器的输出幅度使功率计的显示读数在-10dBm $\pm$ 0.1dB；
- 7) 从射频定向耦合器卸掉功率探头，改为连接到被测接收机输入上；
- 8) 等待新扫描完成后在外控软件上按**【峰值】**，并将光标设置为差值模式；
- 9) 关闭合成信号发生器#1 的射频输出，并设置#2 合成信号发生器的射频输出打开，功率置为 0dBm；
- 10) 在外控软件上按**【峰值】**，调整合成信号发生器#2 AV1461 功率电平，使接收机外控软件上的差值光标读数在 0dB $\pm$ 0.17dB；
- 11) 同时打开两路信号发生器的输出；
- 12) 在外控软件上打开平均功能，并设置平均次数为 5，等待新扫描完成，按**【峰值】**，次峰值分别测试三阶交调信号 900.9MHz 和 901.2MHz 处的幅值大小；

- 13) 按照附表中其他测量频点相应设置信号发生器和被测接收机，完成数据测量；
- 14) 上述差值光标读数为未修正的交调产物，是输入混频器电平为-20dBm 的失真产物。在输入-30dBm 时失真产物将比测量的失真产物低 20dB，从未修正的三阶交调失真中减去 20dB。在附录测试记录表对应测试项中记录最终测量结果。

镜频抑制

在所有频率波段上测试镜像频率响应。信号加在被测接收机的输入端口上，进行参考幅度测量。然后调整信号发生器输出信号等于当前被测接收机的镜像频率，测量并记录显示在接收机上的幅度。

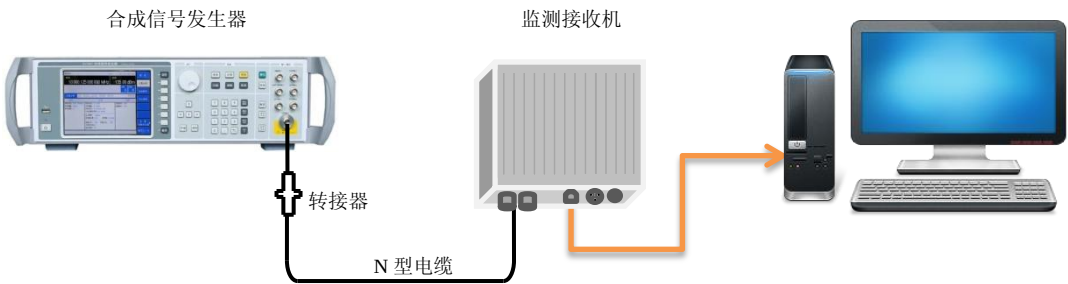


图6-6. 镜像抑制测试设置

- a) 测试设备
 - 合成信号发生器AV1461
 - 外控计算机 联想电脑
- b) 转接器
 - 3.5mm(f)到N(f)转接器 1个
- c) 电缆
 - N(m-m)电缆 1根
- d) 测试步骤
 - 1) 如图连接仪器；
 - 2) 如下设置被测接收机：按照附表测试频率设置中心频率；扫宽 500kHz，采样率 875kHz，FFT 点数 8192，参考电平-10dBm（默认关联衰减：30dB），前放关闭，打开平均功能，平均次数 10；
 - 3) 按照附表测试频点设置 AV1461 输出频率，功率设为-10dBm，逐步调整 AV1461 功率电平使信号峰值接近接收机的参考电平；
 - 4) 在接收机外控程序测量界面上按【峰值】【光标】[Δ 模式]；
 - 5) 设置 AV1461 为附录测试记录表对应测试项中列出的对应射频输入频率的镜像频率点，设置接收机的参考电平为-40dBm（默认关联衰减：0dB），按【峰值】；
 - 6) 在附录测试记录表对应测试项中记录当前差值光标幅度值作为该频点的镜频抑制；
 - 7) 重复 2) 到 6) 步，直到附表中所有频点测试完毕。

中频抑制

测试被测接收机对输入中频信号的抑制能力。信号加在被测接收机的输入端口上，进行参考幅度测量。然后分别调整信号发生器输出信号等于被测接收机的一、二、三中频频率，测量并记录显示在接收机上的幅度值。

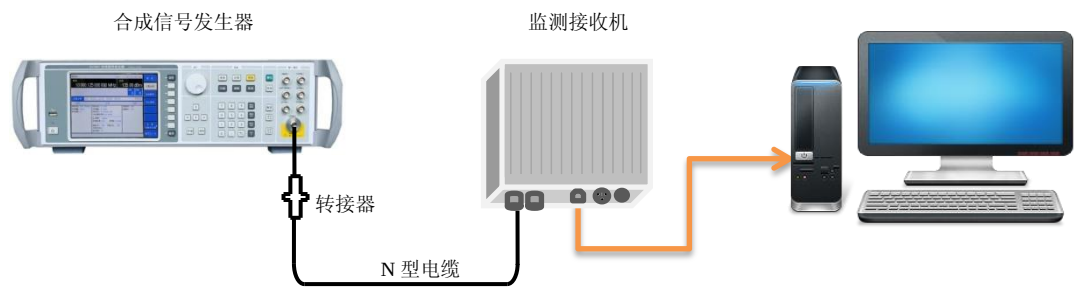


图6-7. 中频抑制测试设置

- a) 测试设备
- | | |
|---------------|--------|
| 合成信号发生器 | AV1461 |
| 外控计算机 | 联想电脑 |
- b) 转接器
- | | |
|-------------------------------|----|
| 3.5mm (f) 到N(f) 转接器 | 1个 |
|-------------------------------|----|
- c) 电缆
- | | |
|----------------|----|
| N(m-m)电缆 | 1根 |
|----------------|----|
- d) 测试步骤
- 1) 如图连接仪器；
 - 2) 分别设置 AV1461 输出信号频率为附表中的中频频率，功率为-10dBm；
 - 3) 相应的设置被测接收机中心频率为附表中的中频频率，扫宽 10kHz；参考电平 -10dBm；分辨率带宽 100Hz；
 - 4) 在接收机外控测量界面上按【峰值】【光标】；
 - 5) 在附录测试记录表对应测试项中记录当前光标幅度值和输入信号幅度的差值。

剩余响应

测试监测接收机的剩余响应，信号输入端口接 50Ω匹配负载。

- a) 测试设备
- | | |
|-------------|---------|
| 匹配器 | 50Ω匹配负载 |
| 外控计算机 | 联想电脑 |



图6-8. 剩余响应测试设置

- b) 测试步骤

- 1) 按照上图连接仪器。在被测接收机的输入端口上连接 50Ω匹配负载。设置接收机参考电平-60dBm（衰减 0dB，前置放大器打开）；分辨率带宽：30kHz（或采样率：28MHz，FFT 点数：4096）；
- 2) 设置接收机按如下 6 个波段分别扫描：20MHz~1GHz、1GHz~2GHz、2GHz~3GHz、3GHz~4GHz、4GHz~5GHz、5GHz~6GHz；并观察噪声基线上是否有剩余响应信号，如果存在剩余响应信号，用光标读出剩余响应幅度，并记录测量结果。测试时应使剩余响应信号的幅度大于显示平均噪声电平 10dB 以上，如果剩余响应信号幅度较小，应进一步减小分辨率带宽以降低显示平均噪声电平；
- 3) 设置被测接收机参考电平-40dBm（衰减：0dB，前置放大器关闭）；
- 4) 重复步骤 2，将测得结果记录在附录测试记录表对应测试项中。

幅度准确度

功率计测试信号发生器输出的读数与被测接收机测试信号发生器输出的读数之差就是绝对幅度准确度。

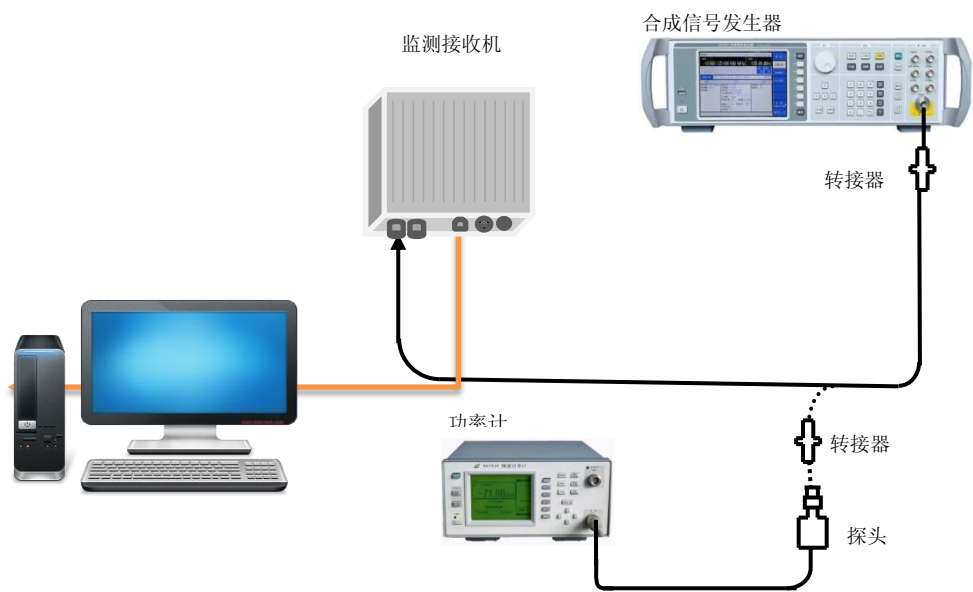


图6-9. 绝对幅度准确度测试设置

- a) 测试设备
 - 合成信号发生器AV1461
 - 功率计AV2434
 - 功率探头AV71711
 - 外控计算机 联想电脑
- b) 转接器
 - 3.5mm(f)到N(f)转接器 1个
 - N(f)到N(f)转接器..... 1个
- c) 电缆
 - N(m-m)电缆 1根
- d) 测量步骤
 - 1) 校准功率计；
 - 2) 如图连接测试设备；
 - 3) 将 AV1461 复位后频率设为 101MHz，幅度为 0dBm；
 - 4) 设置被测接收机中心频率 101MHz；扫宽 100kHz；参考电平 0dBm（衰减：40dB，

- 前置放大器关闭)，分辨率带宽 3kHz；
- 5) 将信号发生器输出连接至功率计探头，输入功率因子，记录功率计显示电平 $L_{\text{功率计}}(0\text{dBm})$ ，并将各个测试频点的 $L_{\text{功率计}}(0\text{dBm})$ 至 $L_{\text{功率计}}(-60\text{dBm})$ 都测试完毕，将信号发生器输出从功率探头处拆下，连接至被测接收机 RF 输入；
 - 6) 设置光标到峰值处，读出光标所示的电平值 L ，计算此时的绝对幅度准确度 $\Delta L = L - L_{\text{功率计}}(0\text{dBm})$ ；
 - 7) 根据附录测试记录表中“绝对幅度准确度”测试栏设置信号发生器的频率和输出功率电平，设置光标到峰值处，读出光标所示的电平值 L ，计算此时的绝对幅度准确度，直至信号发生器的功率电平为 -60dBm；
 - 8) 将通过计算得到的 ΔL 作为绝对幅度准确度的测试结果记录到附录测试记录表中。

衰减器切换不确定度

该测试在整个频段内针对射频和中频衰减器 60dB 范围测量衰减切换不确定度。切换不确定度参考 10dB 衰减器设置。信号发生器中的衰减器是测量标准。

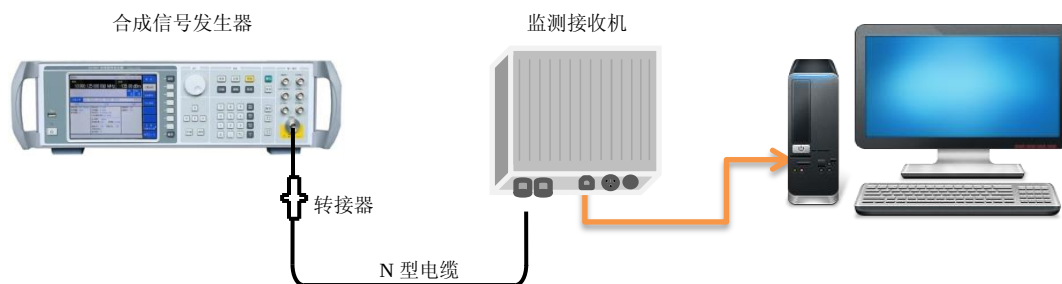


图6-10. 输入衰减器转换不确定度测试设置

- a) 测试设备

合成信号发生器	AV1461
外控计算机	联想电脑
- b) 转接器

3.5mm(f)到N(f)转接器	1个
------------------------	----
- c) 电缆

N(m-m)电缆	1根
----------------	----
- d) 测试步骤
 - 1) 按图连接测试设备；
 - 2) 设置信号发生器输出功率电平为 -60dBm，输出频率为 100MHz，幅度步进量为 10dB。
 - 3) 在接收机外控软件界面如下设置：中心频率 100MHz；扫宽 10kHz，参考电平 -10dBm（衰减 0dB，前置放大器关闭）；
 - 4) 等待新扫描完成，按【峰值】，【光标】，[Δ 模式]，此时以衰减器衰减值为 0dB 时作为参考；
 - 5) 改变 AV1461 幅度，按【幅度】[↑]，使信号源幅度以 -60dBm 为基准向上的变化量 = 衰减值；
 - 6) 在接收机外控软件界面，设置输入衰减器的衰减值；
 - 7) 等待扫描完成后，按【峰值】，计算当前输入衰减器误差 = Δ 光标的幅度差值 - 衰减

值；

- 8) 重复5)到7)步，直到表中所有衰减器衰减测试完毕，将测试结果记录到附录测试记录表中。

输入电压驻波比的测试

用矢量网络分析仪测试被测接收机输入端口的反射系数。

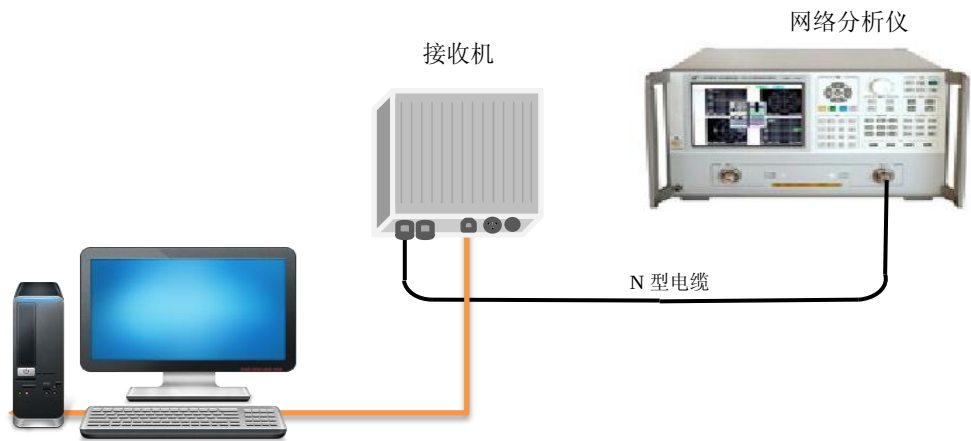


图6-11. 输入端口电压驻波比测试

- a) 测试设备
- | | |
|-------------|---------|
| 网络分析仪 | AV3656A |
| 外控计算机 | 联想电脑 |
- b) 电缆
- | | |
|----------------|----|
| N(m-m)电缆 | 1根 |
|----------------|----|
- c) 测试步骤
- 1) 如图连接测试设备；
 - 2) 设置矢量网络分析仪测量参数为 S11, 格式为驻波比, 频率范围为 20MHz~6GHz, 源功率为-10dBm；
 - 3) 在 N 型电缆连接矢量网络分析仪的端口进行单端口测量校准(包括开路、短路、负载校准)；
 - 4) 完成校准之后，从网络分析仪上取下校准件，将测试端口连接被测接收机的输入端口 1；
 - 5) 在外控软件上设置当前输入端口为端口 1；
 - 6) 在矢量网络分析仪上用光标功能读出最大电压驻波比（VSWR）；
 - 7) 将端口 1 换成端口 2，重复步骤 4-6；
 - 8) 在附录测试记录表对应测试项中记录测试结果。

最大输入电平

表征接收机能够准确测量的最大输入信号电平。

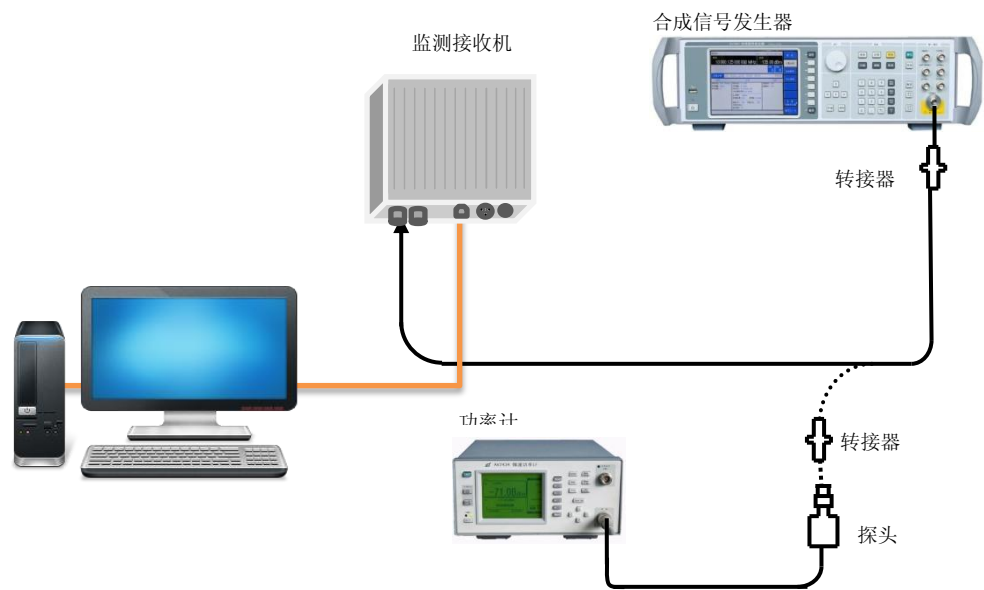


图6-12. 最大输入电平测试设置

- a) 测试设备
- | | |
|---------------|---------|
| 合成信号发生器 | AV1461 |
| 功率计 | AV2434 |
| 功率探头 | AV71711 |
| 外控计算机 | 联想电脑 |
- b) 转接器
- | | |
|---------------------------|----|
| 3.5mm (f) 到N(f) 转接器 | 1个 |
| N (f) 到N(f) 转接器..... | 1个 |
- c) 电缆
- | | |
|------------|----|
| N型电缆 | 1根 |
|------------|----|
- d) 测试步骤
- 1) 如图所示把测试设备连接好，校准功率计；
 - 2) 在外控软件界面设置被测接收机中心频率为 1 GHz, 采样率 28MHz, FFT 点数 4096, 频宽 20MHz, 参考电平+20dBm;
 - 3) 设置 AV1461 频率 1GHz, 幅度+20dBm, 射频输出打开;
 - 4) 将信号发生器输出连接至功率计探头, 输入功率因子, 微调信号发生器的输出功率值, 直到功率计显示电平为 20dBm $\pm$ 0.1dB, 将信号发生器输出从功率探头处拆下, 连接至被测接收机 RF 输入端口 1;
 - 5) 在外控软件界面按【峰值】;
 - 6) 将测得的结果计入测量数据记录表, 计算测量值和信号源实际输出功率的差值, 如果差值幅度误差整机幅度准确度指标内, 则记为合格, 否则不合格。

最大分析带宽

表征接收机单次处理的最大信号频宽, 在本振设置不变的情况下接收机能够分析处理的最大频率和最小频率信号的差值就是最大分析带宽。

- a) 测试设备
- | | |
|---------------|--------|
| 合成信号发生器 | AV1461 |
| 外控计算机 | 联想电脑 |

- b) 转接器
3.5mm (f) 到N型转接器 1个
- c) 电缆
N型电缆 1根

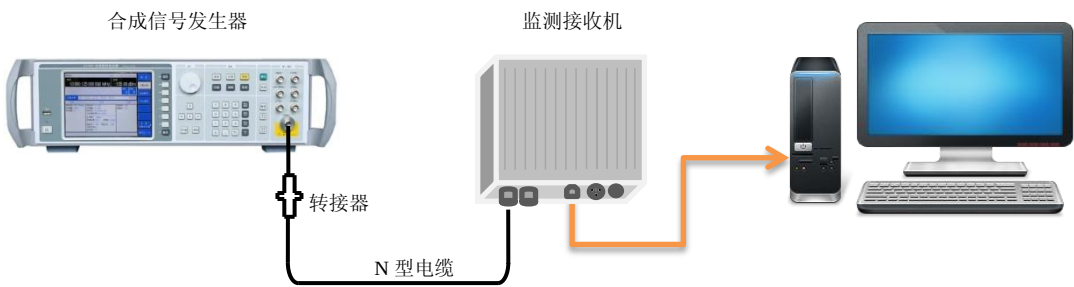


图6-13. 最大分析带宽测试设置

- d) 测试步骤
 - 1) 如图所示把测试设备连接好；
 - 2) 在外控软件界面设置被测接收机中心频率为 1GHz, 采样率 28MHz, FFT 点数 4096, 频宽 20MHz；
 - 3) 设置 AV1461 频率 1GHz, 幅度-2dBm；
 - 4) 在外控软件界面按【峰值】，打开频标差值功能；
 - 5) 设置 AV1461 频率 1.01GHz, 在外控软件界面按【峰值】，如果频标差值幅度值小于-3dB, 旋转旋轮使信号发生器频率减小, 每次设置按【峰值】，记录差值幅度值恰好为 3dB 时的频率值；否则记录为 1.01GHz；
 - 6) 设置 AV1461 频率 0.99GHz, 在外控软件界面按【峰值】，如果频标差值幅度值小于-3dB, 旋转旋轮使信号发生器频率增大, 每次设置按【峰值】，记录差值幅度值恰好为 3dB 时的频率值；否则记录为 0.99GHz；
 - 7) 计算 6、7 步得到的频率值之差, 在附录测试表中对应测试项中记录 20MHz 分析带宽值。

注：在采样率 28MHz, 频宽 20MHz 条件下接收机的信号分析带宽为 20MHz, 外控软件显示结果为单次 FFT 计算结果, 也可通过外控软件读取当前的 IQ 数据, 通过计算机相关处理软件自己计算中频频谱, 可以看到接收机本振频率没有变化, 说明这是一次运算能够测量的分析带宽。

输入端口隔离度的测试

被测接收机两个输入端口之间相互隔离的程度。

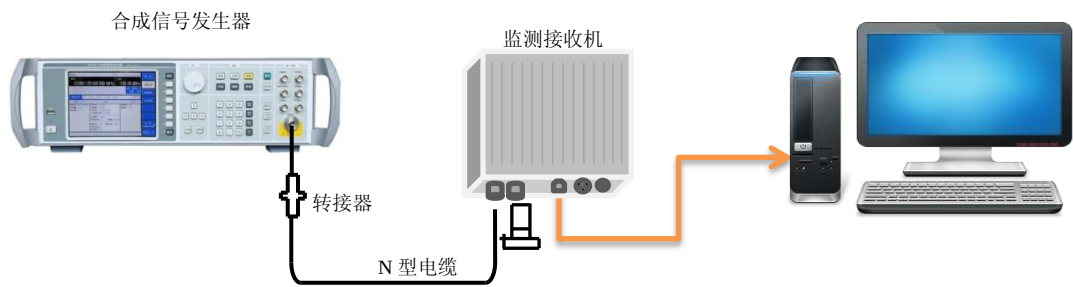


图6-14. 输入端口隔离度测试

- a) 测试设备
- | | |
|---------------|---------|
| 信号发生器 | AV1461 |
| 外控计算机 | 联想电脑 |
| 50Ω 匹配器 | AV70502 |
- b) 电缆
- | | |
|----------------|----|
| N(m-m)电缆 | 1根 |
|----------------|----|
- c) 测试步骤
- 1) 如图连接测试设备；
 - 2) 设置信号发生器输出频率为 20MHz，输出功率为-10dBm，射频输出关闭；
 - 3) 在 N 型电缆一端连接信号发生器的输出端口，另一端连接到接收机的输入端口 1，将接收机的输入端口 2 接上 50 欧姆匹配负载；
 - 4) 打开信号发生器的射频输出，在接收机的外控软件界面设置中心频率为当前信号发生器的输出频率，扫宽 1MHz，FFT 点数 4096；设置当前输入端口为端口 1，测试当前信号的幅度值为 a1，设置当前输入端口为端口 2，测试当前信号的幅度值为 a2，a1 和 a2 二者的差值即为端口 1 到端口 2 的隔离度；
 - 5) 将测试电缆连接到被测接收机的输入端口 2，端口 1 接上 50 欧姆匹配负载；
 - 6) 在外控软件上设置当前输入端口为端口 2，测试当前信号的幅度值为 b1，设置当前输入端口为端口 1，测试当前信号的幅度值为 b2，b1 和 b2 二者的差值即为端口 2 到端口 1 的隔离度；
 - 7) 按照附表将信号发生器输出频率改为其他频点，其他参数不变，重复步骤 3 到 6，测试当前频率下的端口隔离度。

第三篇 维修说明

第七章 故障信息说明及返修方法

本章将告诉您如何发现问题并接受售后服务。其中也包括对无线电监测接收机内部出错信息进行解释。

如果您购买了 AV3900A 无线电监测接收机，在操作过程中遇到一些问题，或您需要购买无线电监测接收机的相关部件、选件或附件，本所将提供完善的售后服务。

通常情况下，产生问题的原因来自硬件、软件或用户使用不当，一旦出现问题请您及时与我们联系。如果您所购买的无线电监测接收机尚处于保修期，我们将按照保修单上的承诺对您的无线电监测接收机进行免费维修；如果超过保修期，我们也只收取成本费。

第一节 正常性检查



声明： 本部分是指导您当 AV3900A 无线电监测接收机出现故障时如何进行简单的判断和处理，如果必要请尽可能准确地把问题返回厂家，以便我们尽快为您解决。

如果您所使用的 AV3900A 出现问题，您可按照以下功能检测过程自行对其进行检查。若问题仍未能排除，请与我们联系。

a) 如图连接被测接收机和外控计算机；

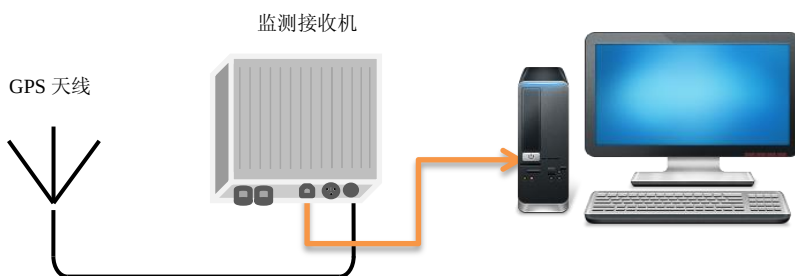


图7-1. 连接示意图

b) 用满足要求的电源供电，打开 AV3900A 无线电监测接收机和外控计算机；

c) 运行外控软件，应显示连接正常并无错误提示；

d) 频谱数据输出

在外控软件界面选择频谱测量功能，应能进入频谱测量界面，并有测量频谱迹线刷新显示，说明频谱数据输出功能正常；

e) IQ 数字中频输出

在外控软件界面选择 IQ 数字中频输出功能，应能进入 IQ 数字中频界面，并可控制 IQ 数字中频信号刷新输出，说明 IQ 数字中频输出功能正常；

f) 时间同步和标记

在外控软件界面选择 IQ 数字中频输出功能，并添加 GPS 同步选项，在 IQ 数字中频测试界面应能看到时间同步结果标示，并且每帧的 IQ 数字中频信号都有时间戳标记，说明时间同步和时间标记功能正常；

g) 程控操作

上述三种功能正常，说明该接收机支持 TCP/IP 网络操作和远程 API 控制，能正常实现程控操作。

第二节 故障说明

● AV3900A 加电后无反应

如果 AV3900A 加电后 20 秒左右没有听到大约 2 秒的启动正常指示音（“滴”音），说明开机运行不正常，请检查：

- 1) 电源接插件连接是否正常；
- 2) 查看适配器指示灯是否点亮，输出电压是否正确；
- 3) 市电连接是否良好；

以上都没有问题后重新接上电源运行，如果故障依旧，则为仪器故障，请联系返修。

● AV3900A 开机指示音正常，但无法跟电脑正常连接

能听到加电后的启动正常指示音，表明仪器启动基本正常，请确认以下设置是否正确：

- 1) 设置 PC 的 IP 地址为 172.141.11.1，子网掩码 255.255.255.0；
- 2) 使用直连网线连接 PC 和 AV3900A，AV3900A 接通电源，在 20 秒左右进入工作状态；
- 3) 运行随机的 AV3900A 外控文件。点击“IP 连接”，在弹出的对话框中显示 AV3900A 的 IP 地址为 172.141.11.202（此为默认），按确认建立连接；
- 4) 连接成功后，菜单条自动切到“扫描控制”，按启动扫描即可观看频谱曲线；
- 5) 切记：请勿随便更改 IP 地址，若更改请牢记，否则无法跟 PC 建立连接；
- 6) 若未更改 IP 地址，以上设置均正确仍无法正常连接，则为仪器故障，请联系返修。

● AV3900A 某性能指标不正常

AV3900A 加电运行连接均正常，但某项技术指标不合格，请安以下步骤检查核对重新连接测试：

- 1) 测量仪器和测试工具是否有效；
- 2) 测试环境是否符合要求；
- 3) 测试端口接头是否损坏，连接是否正常；
- 4) 测试过程及计算方法是否合理；
- 5) 若以上皆无问题，可能为仪器故障，请联系返修。

第三节 返修方法

当您的 AV3900A 出现难以解决的问题时，可通过电话或传真与我们联系。若确认仪器需要返修，请按下面的步骤对仪器进行包装：

- 1) 撰写一份描述仪器故障现象的纸质文档，与测试仪一同放入包装箱；
- 2) 用原包装材料将仪器包装好，以减少可能的损坏；
- 3) 在外包装纸箱四角摆放好衬垫，将仪器放入外包装箱；
- 4) 用胶带密封好包装箱口，并用尼龙带加固包装箱；
- 5) 在箱体上标明“易碎！勿碰！小心轻放！”字样；
- 6) 按精密仪器进行托运，并保留所有运输单据的副本。