我们先来介绍物联网的连接层，在介绍连接层基础上来介绍物联网感知层的相关技术，最后介绍传输和应用层的相关技术。

物联网中有不同的连接技术，除了传统的有线连接技术之外，我们在这里主要介绍无线连接的技术，包括WiFi技术、蓝牙技术包括BLE技术等、ZigBee技术、LoRa、SigFox，每一种技术的适用场景和特点也是不一样的。

为了介绍清楚无线传输的基础以及在上面发展出来的各种技术，我们先想来介绍无线通信的基本原理。无线通信的基本思想就是利用无线电磁波或者无线信号来调制和传输信息。比如最简单的场景中，我们可以产生一个固定频率的无线信号，通过有信号来代表1，没有信号来代表0，来传输信号。由于产生和控制无线电磁型号需要比较复杂的过程，在几年前我在计划该课程的时候一直苦恼怎么让大家很直观的理解其中的技术，最好还能够实际动手试试。现在市面上有能够让我们进行信号处理的设备，但是我觉得还可以做得更直接更好，利用我们的智能手机就能完成这一系列工作。这一个课程中主要利用声波信号来模拟无线信号，声波信号可以很方便的产生，利用普通的智能手机就能够产生、发射和接收声波信号，同时声波信号也易于处理，很好的可视化，同时声波信号的处理和大部分无线信号的处理过程都比较类似，有利于我们理解无线处理的过程。本课程后面的材料中如果没有特殊声明，都讲会使用声波信号来模拟物联网连接过程中的各种技术，请同学们也自己动手来多尝试一下。

在介绍声波信号的处理之前，我们先来理解一下声波信号的基本特征。一般来说，人可以听到的声波范围大概为20Hz到20KHz中间，因此现在大部分手机声音采样率为44100，对应到能够处理的最大声音频率为22KHz左右。超过20KHz的声波我们经常叫做超声波，

声波的生成、发送和接收：

这一部分对应了无线数据发送的时候的生成、发送和接收。在无线数据发送的过程中，这里每一部分都对应了很多的操作。我们先介绍声波的一些基本知识，然后通过声波的基本知识介绍数据发送和接收的过程。

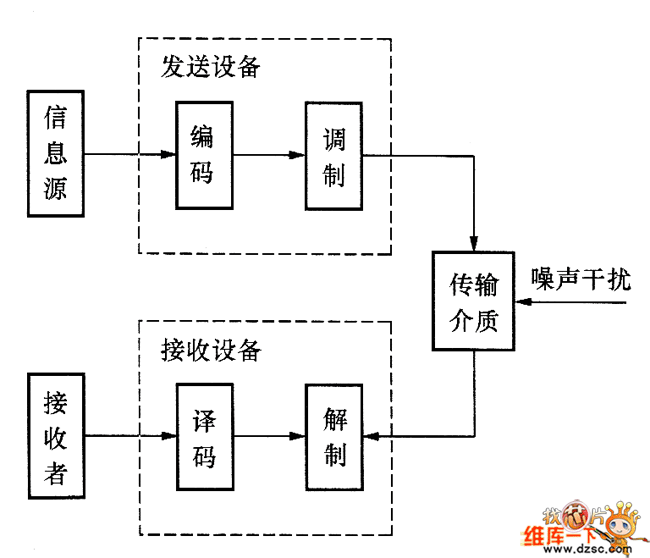
声音是一种机械波，有物体的震动引起了空气的震动，从而能够传播声音。因此出现了超声波洗龙虾、超声波洗牙。

关于Matlab的主要语法，有过任意编程语言经验的人应该都不会感觉到复杂，请大家自行学习。

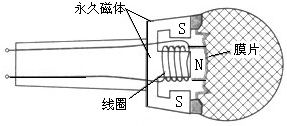
日常生活中我们能看到的信号大部分都是模拟信号，我们能看到的风雨雷电，我们听到的声音，看到的其他的景色等，要将这些信息存入到电脑中间，需要经过一系列的处理。电脑中的数据都是数字化的，也就是都可以用01bit来表示的。因此要讲模拟的数据存入到电脑中，其中重要的步骤就是采样和量化。

模拟信号到数字信号

这是信号处理中的关键一步，因为网络通信尤其是无线通信的时候，利用的都是物理信号（如电磁信号），因此通信过程中的重要一步就是将模拟信号转成数字信号。



如果真实存在着一个1KHz的声波信号，我们需要接收和处理这个信号的话，先要对信号进行采样和量化，这实际上也是所有计算机处理数据和信号的基础。我们先来看，如果有一个人在说话，这个说话显然是模拟的信号，我们如何将说话来保存到电脑中。计算机中将所有模拟信号转化为数字信号是通过采样和量化来实现的。具体来说，通过将模拟信号转化为电信号（通常为电压信号），然后通过将电压量化转成数字信号。在声音信号的处理中，我们首先依赖的是麦克风。麦克风能够将模拟的声音信号转化为电压信号，简单来说，麦克风上有一个可以随着声音振动而振动的薄膜，薄膜振动引起了产生电压的不断变化，因此电压的变化就代表了声音信号的变化。



不过大家要注意的是，这个时候电压在时间和大小上依然是连续的，也就是说任意的电压值都是有可能的，很显然，这样大数据在电脑上是无法保存的，保存的大小是无穷大。为了保存这样的数据，我们先要对数据进行处理。第一个处理就是连续的信号转化为离散的信号。这一步是通过采样来实现的。采样是指计算机设备定时的对信号电压进行采集，采样解决了连续信号到离散信号的问题，可是采样后的电压值仍然没有办法直接保存，因为电压值在可能的取值范围内是连续的，为了解决这个问题，就需要量化。量化简单的来说就是将无限的连续的电压取值转化为有限的离散的电压取值，这样采样后的结果就能够保存。我们平时说的模拟/数字转换器（Analog to Digital Converter，即ADC，与之对应的是数/模转换器,即DAC）其实也就是干的这样的事情。

具体这个事情是怎么来实现的呢，我们首先通过简单的仿真实验来进行情况说明，然后我们再通过实际的信号让大家体会这一过程。假设存在一个正弦的声音信号，频率为5Hz，理想情况下这个信号是一个正弦波，这个信号在matlab下可以如下产生和画图出来。注意，matlab下产生的已经是离散的信号了。

%% 信号的产生过程

%% 产生一个频率为1000、时长为1s的信号；

t = 0:1/200:1; % 1s内2000个采样点

f = 5; % 频率f=1000

y = sin(2\*pi\*f\*t);

plot(t, y);



注意这里面有几个小的问题，当我们将信号频率稍微做一些修改好，画出来的图将变成下面这个样子。

%% 信号的产生过程

%% 产生一个频率为100、时长为1s的信号；

t = 0:1/200:1; % 1s内2000个采样点

f = 100; % 频率f=1000

y = sin(2\*pi\*f\*t);

plot(t, y);



注意这个图已经完全不是我们想象中的正弦图像了，大家想想看看这个是什么原因。

这里面有几个很有趣的问题，是不是连续的信号就无法用离散的信号表示出来，连续的信号是不是就得用无穷多的离散信号来表达？这里牵涉到了信号采样中的一个重要问题，我们将这个问题更加具体化，假设信号中的最大频率为f，那么应该以多大的采样频率对信号进行采样合理呢，显然不可能无限大的采样频率，这样在计算机上无法处理，实际上也非常不经济。这个问题也就变为信号的频率为f，那么最少的采样频率为多少。

奈奎斯特采样定律告诉我们，对于这样的情形，需要的采样频率为2\*f。采样过程所应遵循的规律，又称[取样定理](https://baike.baidu.com/item/%E5%8F%96%E6%A0%B7%E5%AE%9A%E7%90%86)、抽样定理。采样定理说明采样频率与[信号频谱](https://baike.baidu.com/item/%E4%BF%A1%E5%8F%B7%E9%A2%91%E8%B0%B1)之间的关系，是[连续信号](https://baike.baidu.com/item/%E8%BF%9E%E7%BB%AD%E4%BF%A1%E5%8F%B7)[离散化](https://baike.baidu.com/item/%E7%A6%BB%E6%95%A3%E5%8C%96)的基本依据。

我们来看一个更加有意思的例子，如果信号的采样频率设置得不对会怎么样？

%% 信号的产生过程

%% 信号的产生过程

%% 产生一个频率为100、时长为1s的信号；

t = 0:1/10000:1; % 1s内2000个采样点

f = 401; % 频率f=1000

y = sin(2\*pi\*f\*t);

plot(t, y);

这个例子中，信号是400Hz的，理论上来说，信号是如下这个样子的。

%% 信号的产生过程

%% 产生一个频率为100、时长为1s的信号；

t = 0:1/402:1; % 1s内2000个采样点

f = 401; % 频率f=1000

y = sin(2\*pi\*f\*t);

plot(t, y);

大家可以先来想象一下这个采样出来的结果会是怎么样的。



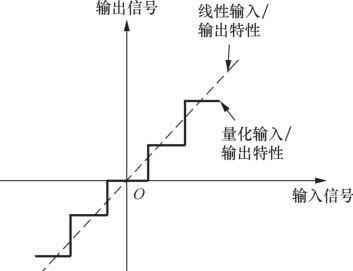
思考：

1. 为什么会出现这样的结果。
2. 如果频率为f，采样频率为fs，那么混叠后的频率将会是多少，这样会对通信产生什么影响。
3. 用于演示混叠现象的经典例子之一是所谓的“车轮效应”。在影片里当马车越走越快时，马车车轮似乎越走越慢，然后甚至朝反方向运转。为什么？

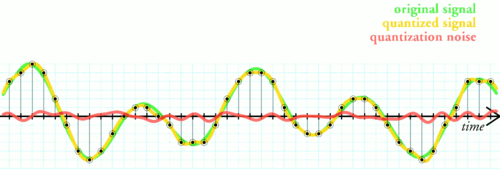
**量化**

在数字信号处理领域，[量化](https://baike.baidu.com/item/%E9%87%8F%E5%8C%96/8959421)指将信号的连续取值（或者大量可能的离散取值）近似为有限多个（或较少的）离散值的过程。量化主要应用于从连续信号到数字信号的转换中。连续信号经过采样成为离散信号，离散信号经过量化即成为数字信号。注意离散信号通常情况下并不需要经过量化的过程，但可能在值域上并不离散，还是需要经过量化的过程 。信号的采样和量化通常都是由ADC实现的。

例如8位的ADC可以将标称[输入电压范围](https://baike.baidu.com/item/%E8%BE%93%E5%85%A5%E7%94%B5%E5%8E%8B%E8%8C%83%E5%9B%B4)内的模拟电压信号转换为8位的[数字信号](https://baike.baidu.com/item/%E6%95%B0%E5%AD%97%E4%BF%A1%E5%8F%B7)。思考：多少位不同的电压值？



量化效果图

[](https://en.wikipedia.org/wiki/File:Quantization_error.png)

The simplest way to quantize a signal is to choose the digital amplitude value closest to the original analog amplitude. This example shows the original analog signal (green), the quantized signal (black dots), the [signal reconstructed](https://en.wikipedia.org/wiki/Signal_reconstruction) from the quantized signal (yellow) and the difference between the original signal and the reconstructed signal (red). The difference between the original signal and the reconstructed signal is the quantization error and, in this simple quantization scheme, is a deterministic function of the input signal.

为了进一步说明这种四舍五入方法，这里以3 位(3 bit)的二进制数码进行量化的过程为例，如图4-9所示。

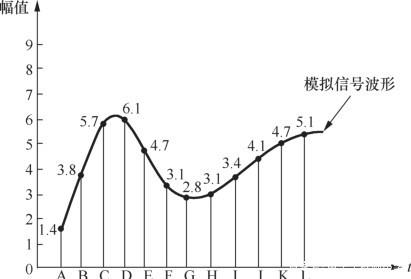
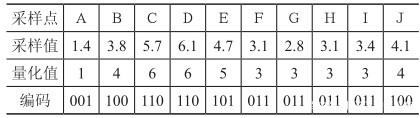
图4-9 示意图

图4-9中，输入信号是一个模拟信号，采样点为A～L，幅值范围为0～9，采样值、量化和编码的结果如表4-3所示。

表4-3 采样值、 量化和编码的结果

<https://baijiahao.baidu.com/s?id=1614627981206121506&wfr=spider&for=pc>

可见量化的过程中，输出的信号和原信号是有一定的区别的，这个区别就是量化噪声。我们通常说的高保真，其中有一个重要步骤采取足够多的量化位数，保证量化噪声足够小，量化出来的信号足够接近原始信号，当然其中带来的代价就是存储和处理的开销变大。大家可以去网易音乐上看看高保真音乐的，他后面以后一个码率，这一个码率跟量化是直接相关的。思考：大家想想那个码率都跟什么因素相关。

按照[量化级](https://baike.baidu.com/item/%E9%87%8F%E5%8C%96%E7%BA%A7)的划分方式分，有均匀量化和非均匀量化。思考：为什么有这两种量化的方式？

[均匀量化](https://baike.baidu.com/item/%E5%9D%87%E5%8C%80%E9%87%8F%E5%8C%96)：ADC输入动态范围被均匀地划分为2^n份。

[非均匀量化](https://baike.baidu.com/item/%E9%9D%9E%E5%9D%87%E5%8C%80%E9%87%8F%E5%8C%96)：ADC输入动态范围的划分不均匀，一般用类似指数的[曲线](https://baike.baidu.com/item/%E6%9B%B2%E7%BA%BF)进行量化。

非均匀量化是针对均匀量化提出的，因为一般的语音信号中，绝大部分是小幅度的信号，且人耳听觉遵循指数规律。为了保证关心的信号能够被更精确的还原，我们应该将更多的bit用于表示小信号。

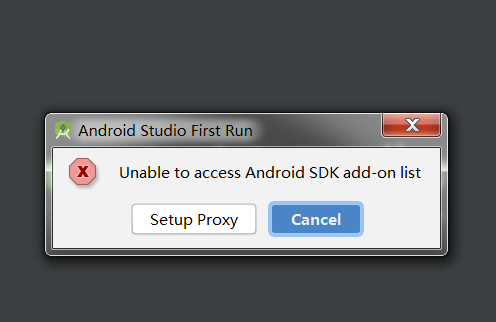
假设信号采用的是均匀量化的方法，量化噪声带来的信噪比变化，

思考：

假设要将最大频率为8KHz的唱歌的信号转成数字信号，量化bit为8bit，则产生的信号码率至少是多少？1min的信号大小是多少？

到了这里，我们来看一个真实的采样和量化的例子，这也是我们整个数据处理实验的开始。这个实验在手机上就能实现，我们来看看手机上如何进行声音的采样和量化。

在做实验之前，你需要准备好一个手机（安卓和苹果手机都可以），并配置好基本的开发环境，如果你还不太熟悉，参见？。



安装好了之后，可以启动第一个声音记录的小程序。仔细看这个小程序，跟大家学习过的网络传输的过程是类似的，前面有一个ADC不断的将声音信号转化成为数字信号，这个数字信号是存储在一个buffer里面的，需要我们讲buffer中的数据不断的读取出来然后保存成声音信号。类似于网络中的数据处理，网卡有一个很重要的功能就是将物理信号转化为数字信号，然后也是需要我们将网络收到的数据从buffer中读取出来。事实上这一个例子也展示了大多数需要用到ADC转化的功能，数据被转换并且保存在一个buffer中，然后对buffer中的数据进行处理。当然进一步来看的话，这个buffer中的数据如何处理也是有不同的手段的，有一些是需要轮询，不断的取看buffer中是否有数据，有一些是基于中断，一旦有数据就会产生一个中断通知进行数据处理，这两种处理方法都可以实现非阻塞模式，即用户无需在这个地方等待buffer内容。还有一种是阻塞模式，用户需要等待buffer内容，当buffer中有数据时才能够继续往下处理。几种模式各有优缺点，这也是数据处理中，尤其是通信和与硬件打交道的数据处理中经常要面临的问题。如果数据处理不及时，数据就会被覆盖导致丢失，这个问题在网络传输中也会遇到。

这里面通过这一个小的程序我们想介绍几个概念，这也是平时大家在网络学习的过程中容易忽视的概念，因为网络学习中我们很难有机会从最底层开始接触到基础的信号处理。在启动录音的过程中，需要设置几个参数，第一个就是采样率，目前设置的是44100Hz，基本能够将人能听到的声音录下来。一个是量化的bit位数，这里设置的是16位，即每一个采样点用16bit来量化。另外一个是声道数量，目前选择的是双声道，即有两路数据。声音编码的格式采用的是PCM格式，这也是一种最基础的声音编码方法，与我们常见的mp3等格式不同，那些格式需要将声音文件进行进一步的编码。这个数据文件也是我们后面会要用到的数据的基础格式。

傅里叶变换

为了更加便于来分析一个信号，我们在介绍信号的更多属性前，我们先来介绍一下傅里叶变换。给定一个采样的信号，如何知道信号中的所有频率呢？傅里叶变化的一个最基本的作用就是建立数据时域和频域之间的一个转换关系。那么傅里叶变换是怎么得到的呢。我们先不看这么复杂的过程，先来看一个简单的例子。

**那么2.4GHz信号是否需要4.8GHz的采样频率，也就是意味着1s之内有4.8GHz的数据？**

**线性时不变系统**

傅里叶变化。

前面的研究我们可以看出，正弦信号是具备很好的属性，我们都很熟悉，也很好处理的。可是实际中有一个问题，不是每一个信号都是正弦信号，那如何处理这些信号呢。研究中我们有一个方法，可以把一个未知的问题转化为已知的问题，那么一个很自然的想法就是，能不能将一个未知的不好处理的信号转化为已知的好处理的信号。