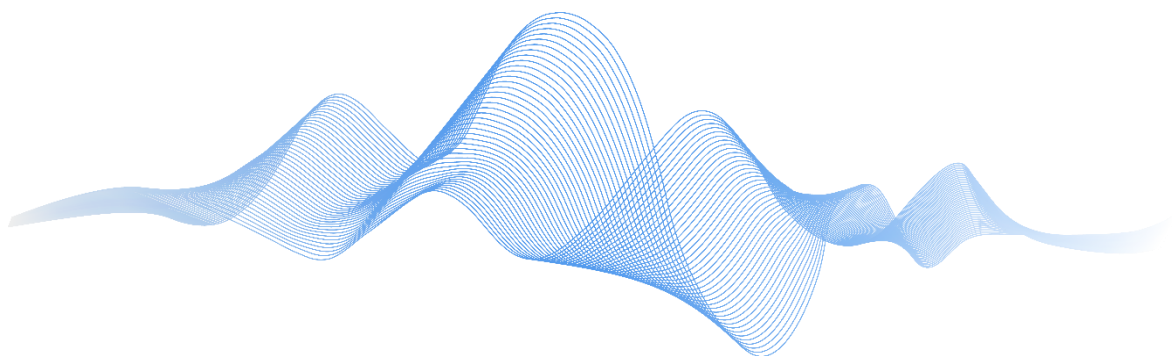


Ultra-S 系列 用户编程手册



产品名称	宽频任意波超快超声成像与检测平台
文件编号	LY202607002
文件状态	☐ 草稿 ☒ 正式发布 ☐ 正在修改
当前版本	V1.1
编制日期	2026-07-01

浙江利影医疗科技有限公司

目 录 Contents

1. 简介	- 1 -
2. Scripts 架构	- 2 -
2.1 整体架构说明	- 3 -
2.2 根目录顶层文件说明	- 3 -
2.3 子文件夹分类说明	- 3 -
2.3.1 成像算法模块	- 3 -
2.3.2 原始数据处理工具模块	- 4 -
2.3.3 仿真调试工具模块	- 4 -
2.3.4 底层公共支撑模块	- 4 -
3. LYconfig	- 4 -
3.1 LYConfig.AcqConfig.Probe	- 5 -
3.1.1 name	- 6 -
3.1.2 type	- 6 -
3.1.3 element_pitch	- 6 -
3.1.4 element_num	- 6 -
3.1.5 element_radius	- 6 -
3.1.6 element_pos	- 6 -
3.1.7 image_start	- 8 -
3.1.8 bias_vol	- 8 -
3.1.9 tx_ele_map	- 8 -
3.1.10 rx_ele_map	- 8 -
3.2 LYConfig.AcqConfig.Tx	- 10 -
3.2.1 channel	- 11 -
3.2.2 fs	- 11 -
3.2.3 sos	- 11 -
3.2.4 focus_depth	- 11 -
3.2.5 prf	- 12 -
3.2.6 steer	- 12 -
3.2.7 Pulse/任意波	- 12 -
3.2.8 aperture	- 21 -
3.2.9 sequence	- 22 -
3.2.10 AFE+ATGC	- 27 -
3.2.11 FirFilter+DownsampleM+DownsampleLoc	- 28 -

3.2.12 fs_win	- 31 -
3.2.13 trigger_mode	- 33 -
3.2.14 trigger_in	- 33 -
3.2.15 trig_out	- 34 -
3.2.16 sampling_delay_sequence+sampling_delay_time	- 35 -
3.2.17 mux + mux_sequence + mux_delay	- 35 -
3.3 ScanList	- 35 -
3.3.1 头信息	- 35 -
3.3.2 时序控制	- 36 -
3.3.3 波形调度	- 37 -
3.4 LYConfig.AcqConfig.Rx.....	- 38 -
3.4.1 fs	- 39 -
3.4.2 sos	- 39 -
3.4.3 ImageDepth / sample_num.....	- 39 -
3.4.4 Revdepth / Revpt.....	- 39 -
3.4.5 frameN/timeN	- 39 -
3.4.6 fetch_frequency/NumsPerFile	- 40 -
3.4.7 aperture	- 41 -
3.4.8 sequence	- 42 -
4 LYConfig.AcqConfig.PostConfig.....	- 44 -
4.1.1 demo_depth.....	- 45 -
4.1.2 demo_value.....	- 45 -
4.1.3 filterdepth.....	- 45 -
4.1.4 filtercoef.....	- 45 -
4.1.5 filtertype.....	- 45 -
4.1.6 filterorder	- 45 -
5 典型脚本编程示例	- 46 -
5.1 聚焦波示例	- 46 -
5.2 平面波示例	- 50 -
附录 1. Pinmap（引脚映射）说明	- 54 -
版权与法律声明	- 56 -
版权所有	- 56 -
使用限制与许可	- 56 -
编程免责与责任限制	- 56 -

1. 简介

本文档专为初次使用 Ultra-S 系列超声研究系统的开发者编写。阅读并使用本指南要求开发者具备一定的超声基础知识与 MATLAB 编程经验。如需补充相关基础，建议参考以下权威著作：

- 超声领域：K. Kirk Shung 著《Diagnostic Ultra-Sound Imaging and Blood Flow Measurements》（第二版）
- MATLAB 编程：Brian R. Hunt 著《A Guide to MATLAB, 3e: For Beginners and Experienced Users》

Ultra-S 研究系统主要由采集硬件与计算机端软件两大部分组成。该采集硬件包含用于多通道发射波形生成、模拟接收信号放大与滤波、数字信号处理以及扫描序列控制的电子模块；计算机则内置基于 Matlab 编程环境开发的软件模块，使用户能够在硬件和软件环境中对一系列操作（称为“事件”）进行编程与执行。事件序列通过编写 Matlab 脚本来实现，该脚本可生成一组可载入系统的对象；这些对象使用定义了属性集的 Matlab 结构体进行参数化配置。当设置脚本运行时，会生成包含所有对象结构及编程信息的二进制数据文件，命名为 LYConfig，并以 Matlab 中的 .mat 格式存储；

完成 LYConfig 配置后，用户需调用对应的 DAQ_RealTime_X 函数启动系统（其中 X 代表特定的成像功能，例如平面波成像对应 DAQ_RealTime_PlaneWave）。该函数在 Matlab 环境中作为常规脚本运行，并通过硬件抽象层（HAL）与硬件通信。DAQ_RealTime_X 函数从 LYConfig 文件中读取 Ultra-S 硬件所需的属性，整个操作流程会被同步载入硬件和软件序列控制器中，并启动执行过程，并根据设定情况同时打开图形用户界面和显示窗口。控制与处理任务使用 C 语言编写，为了保障实时性，底层的波束合成、图像处理及显示等高负载任务均由 C 语言和 CUDA 联合编写。

在序列执行过程中，DAQ_RealTime_X 函数不仅充当软件序列控制器，还负责处理所有与硬件的实时交互。对于无需在 MATLAB 环境中额外干预的操作，当前事件序列执行完毕后，系统会自动从头循环或从指定的返回点继续运行。事件采集结束后，Ultra-S 系统提供了专用的 MATLAB 解析脚本，方便用户对采集到的原始数据进行读取、查看或深度的离线分析。

为了帮助您快速上手，我们强烈建议遵循以下学习路径：

强烈建议新用户首先阅读位于文档文件夹中的《用户入门手册》。对 Ultra-S 中的一些概念、定义进行初步了解。

本《编程手册》目的是给用户 提供编程结构的详尽参考信息，该教程将引导您逐步完成

序列程序的构建，并详细介绍了相关编程技术背后的原理。

入门的最佳方式是研究并分析 **Scripts** 文件夹中的各类示例脚本，通过示例脚本熟悉系统的基本操作，从而掌握更高级的编程技巧。在需要开发新序列时，建议直接从现有示例库中寻找一个与目标功能相近的脚本，在此基础上进行修改与调试，这将大幅提升您的开发效率。

后续章节阐述了用于编程 Ultra-S 系统的各类结构和对象。要掌握编写序列程序所需的所有结构定义方法，最佳途径是研究对应功能的示例脚本，比如平面波 `PlaneWave_Acq.m`、聚焦波 `FocusWave_Acq.m`。通过分析这些示例脚本并了解对象的定义方式，可更深入地理解编程对象及其使用方法。

2. Scripts 架构

下面是整个 Scripts 文件夹架构

Scripts 主根目录/

└─ 顶层文件

| └─ version3 1.txt 版本日志

| 用户入门手册.docx

| 用户入门手册.pdf

1 成像算法模块

PlaneWave 平面波成像脚本

| | — FocusWave 聚焦波成像脚本

— DivergingWave 发散波成像脚本

| SyntheticAperture 合成孔径成像脚本

光声成像脚本

| $\vdash$ GUI GUI 程序

| — DeepLearn 深度学习脚本

1 原始数据处理工具模块

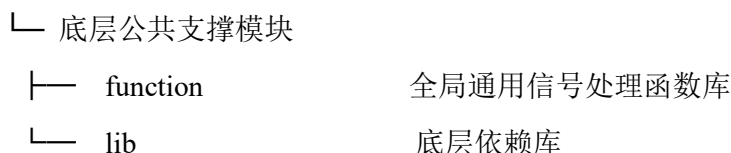
— RawDataConverter 原始数据格式转换工具

└─ RawDataViewer 原始波形查看分析工具

1 仿真调试工具模块

| — EmitTest 发射波形测试工具

| — Simulation 声场仿真工具



2.1 整体架构说明

Scripts 脚本采用模块化分层文件架构，按照“顶层说明文档、成像算法模块、数据处理工具、通用底层支撑”分层组织文件。根目录存放版本记录、项目用户入门手册文档；各类成像算法、原始数据处理、通用函数、底层依赖等功能均独立划分子文件夹，各模块职责解耦、公共资源统一复用，便于功能迭代、代码维护与工程部署。

2.2 根目录顶层文件说明

根目录仅保留版本记录与项目整体说明文档，软件交互、算法代码全部收纳于子文件夹内：

version3_1.txt：版本日志文件，记录当前版本更新内容、功能迭代记录、软硬件兼容限制、历史版本变更信息。

用户入门手册.docx / 用户入门手册.pdf 为整体概要文档，提供可编辑 Word 版本与只读 PDF 版本；内容涵盖系统整体架构、各模块功能概述、算法依赖关系、软硬件运行环境、整体使用流程。

2.3 子文件夹分类说明

根目录下共 13 个功能子文件夹，按用途划分为「成像算法模块、原始数据处理工具、仿真工具、底层公共支撑模块」四大类。

2.3.1 成像算法模块

存放各类超声 / 光声成像核心重建算法，按成像模块独立分区：

PlaneWave：平面波成像脚本，包含线阵、凸阵、RCA 采集、成像、分析脚本；

FocusWave：聚焦波成像脚本，包含线阵、凸阵、相控阵采集、成像、分析脚本；

DivergingWave：发散波成像脚本，包含线阵、相控阵采集、成像、分析脚本；

SyntheticAperture：合成孔径成像脚本，用于适配 1024 阵元，支持多路复用功能；

PA：光声成像模块，包含凹阵采集、成像、分析脚本。

在上述每个子文件夹内，均包含对应功能的脚本、配置文件、说明文档，确保各模块独立可运行、可测试。

GUI: GUI 图形交互界面提供聚焦波、平面波参数设置、数据采集的功能。具体使用见《GUI 用户操作手册.pdf》。

DeepLearn: 深度学习成像模块 基于深度学习的超声波束合成，作为传统成像算法的 AI 扩展模块，在对应的 zip 文件解压后可以参考《深度学习+波束合成 脚本使用说明.docx》。

2.3.2 原始数据处理工具模块

用于超声射频原始数据转换、查看、分析的配套工具：

RawDataConverter: 原始通道 ADC 数据格式转换，按照需求获取对应的通道数据，并保存为对应格式，具体使用见《ADCConverter.pdf》。

RawDataViewer: 原始通道 ADC 数据查看工具，提供 ADC 原始回波数据加载、波形可视化、数据分析功能，用于探头通道功能校验、数据分析，具体使用见《ADCViewer_MapMode.pdf》和《ADCViewer_AMode.pdf》。

2.3.3 仿真调试工具模块

用于发射时序调试、通用声场仿真的辅助工具集：

EmitTest: 发射序列调试模块，支持脉冲、Chirp 等多种发射波形参数调试。

Simulation: 支持平面波、聚焦波超声发射声场仿真，点散射体模拟、波束形成与图像重建，具体使用见《仿真说明书.pdf》

2.3.4 底层公共支撑模块

全项目共用基础函数、依赖库模块：

function: 全局通用函数库 存放全模块复用的信号处理子函数，包含前端处理、图像后处理等通用接口，所有成像算法统一调用该文件夹工具，避免代码冗余。

lib: 底层依赖库目录，存放程序运行必需的第三方信号处理库、GPU 加速 CUDA 内核、静态链接库等底层依赖文件，为软件运行基础依赖。

3. LYconfig

LYConfig =

AcqConfig	struct	与前端采集相关参数（Probe/Tx/Rx...）
FullPath	char	数据保存路径（SaveDataflag =0 时为[]）
ImgShowflag	double	0 = 关闭实时显示（default），1 = 开启

PostConfig struct 图像后处理参数 (ImgShowflag = 0 时为[])

SaveDataflag double 0 = 不保存数据, 1 = 保存 (default)...

LYconfig 文件主要由以下 5 个核心变量组成, 用于控制系统的前端采集、数据存储与图像显示:

- AcqConfig (struct)

用于存储前端采集相关的核心硬件参数, 具体包含探头 (Probe)、发射序列 (Tx)、接收序列 (Rx) 以及扫描序列 (ScanList) 配置信息。

- SaveDataflag (double)

ADC 数据保存模式控制标志:

0: 不保存数据。

1: 自动保存 (默认)。参数下发完成后, 系统会立即将获取到的 ADC 数据保存至指定文件夹。

2: 手动保存。仅当用户按下“保存”按钮时, 系统才会保存 ADC 数据。

- FullPath (char)

数据保存的目标路径。注意: 此参数受 SaveDataflag 状态影响。当 SaveDataflag = 0 时, 此项为空 ([]), 即使强制设置路径也不会生效。

- ImgShowflag (double)

实时图像显示控制标志:

0: 关闭实时显示 (默认), 不显示重建图像。

1: 开启实时显示, 在界面上展示重建后的图像。

在脚本运行允许的情况下, 强烈建议采用 ImgShowflag = 1 配合 SaveDataflag = 2 的组合。这样不仅可以实时观测图像, 还能按需手动保存重要数据。

- PostConfig (struct)

图像后处理参数集合。用户可通过调节该结构体内的参数来优化最终的成像质量。

注意: 当 ImgShowflag = 0 (即关闭图像显示) 时, 此项会被置空 ([]).

3.1 LYConfig.AcqConfig.Probe

Probe 结构体用于规定换能器 (探头) 的各项物理及逻辑参数。用户需要在脚本 Probe_para.m 中配置相关字段, 以指定换能器的类型、阵元数量、间距等关键信息。

LYConfig.AcqConfig.Probe =

type	char	换能器类型 (linear/convex/phase...)
name	char	换能器名称
element_pitch	double	换能器 pitch, 单位 m

element_num	double	换能器阵元个数
element_pos	struct	换能器阵元位置 x、y、z、theta
element_radius	double	换能器曲率半径，单位 m
image_start	double	图像成像区域起始位置，单位 m
tx_ele_map	double	发射通道映射
rx_ele_map	double	接收通道映射
bias_vol	double	（可选）直流偏置电压使能。1 = 启用，0 或未配置 = 禁用系统内置偏置电压供电

3.1.1 name

换能器名称，由用户自定义。

3.1.2 type

换能器类型：支持线阵、凸阵、相控阵、矩阵、凹阵、RCA，输入形式：'linear'、'convex'、'phase'、'matrix'、'concave'、'rca'。

3.1.3 element_pitch

换能器阵元间距，由换能器厂家提供；

3.1.4 element_num

换能器阵元数量，由换能器厂家提供；

3.1.5 element_radius

凹、凸阵换能器曲率半径，单位 m。

3.1.6 element_pos

在完成上述基础参数配置后，Probe_para 函数会自动计算阵元的空间几何位置 element_pos，其包含以下 4 个长度为 N（1*N 维，即阵元总数）的数组：

element_pos.x: 换能器阵元 x 轴坐标（m）

element_pos.y: 换能器阵元 y 轴坐标（m）

element_pos.z: 换能器阵元 z 轴坐标（m）

element_pos.theta: 阵元法线夹角（rad）。对于凹/凸阵，代表阵元与阵列中心连线同 Z 轴的夹角；线阵和相控阵此处全置为 0。

对于 RCA 换能器有部分不同，对于 128+128 换能器，总共 256 个阵元与 256 通道数量一致，X 方向的 128 个阵元对应前 128 通道，Y 方向的 128 个阵元对应后 128 通道，在计算 X、Y 坐标时，对于 X、Y 方向维度为 1×256 ，X 坐标前 128 为实际的物理位置，后 128 置 0，Y 坐标前 128 置 0，后 128 为实际的物理位置；这里将阵元数量减少为 32+32 阵元换能器，那么一共对应 64 通道，X 方向的 32 个阵元对应 1~32 通道，Y 方向的 32 个阵元对应 33~64 通道，在计算 X、Y 坐标时，对于 X、Y 方向维度为 1×128 ，前 64 个值分别为 X、Y 方向阵元实际物理位置，后 192 个值为 0。

（注：各换能器默认坐标系示意图请参考图 1 - 图 4）

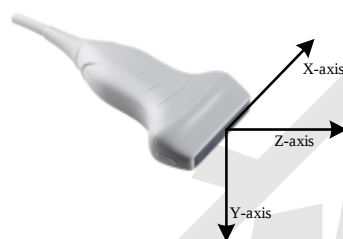


图 1 线阵、相控阵换能器坐标系

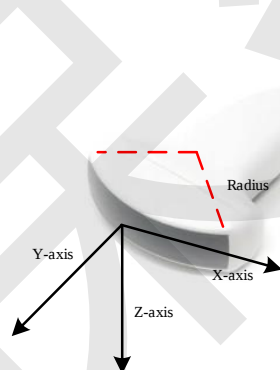


图 2 凸阵换能器坐标系

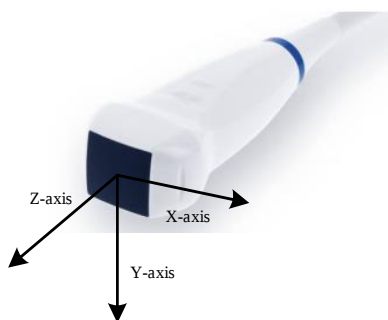


图 3 矩阵、RCA 换能器坐标系

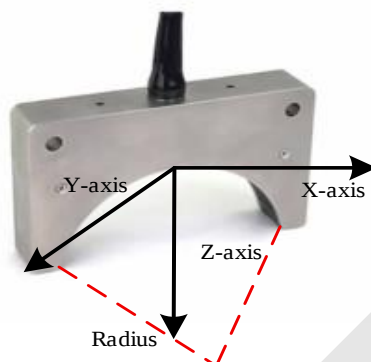


图 4 凹阵换能器坐标系

3.1.7 image_start

换能器成像起始深度，单位 m，与换能器匹配层相关，用于射频数据起始段截断裁剪，用户配置。

3.1.8 bias_vol

部分探头需借助系统自带偏置电压源，向换能器阵元施加直流偏置电压；本可选属性用于开启偏置电压输出，1 代表启用偏置电压，0 代表未启用，若未配置 Probe.bias_vol 属性代表系统内置偏置电压供电功能未启用。

3.1.9 tx_ele_map

发射通道映射表，用于定义换能器逻辑阵元与物理发射通道之间的映射关系。支持 256 阵元及以下换能器，数组长度固定为 256 且不可更改。索引代表逻辑阵元号（从 0 开始），值代表物理通道号。

对于 256 及以下阵元数量的换能器，默认映射为 0:255（即第 0 个阵元对应发射通道 0，第 1 个阵元对应发射通道 1……）。对于定制换能器，如果阵元与通道连接不是一一对应的，用户需要重新分配此映射表。

3.1.10 rx_ele_map

接收通道映射表，Ultra-S 存在默认的硬件接收通道错位，其默认序列为：

[0,2,1,3,4,6,5,7,8,10,9,11,12,14,13,15,16,18,17,19,20,22,...0,249,251,252,254,253,255]

由于 Ultra-S 存在默认的硬件接收通道错位（如 [0, 2, 1, 3, 4, 6, 5, 7..]，即第 1 列对应通道 2，第 2 列对应通道 1），假设获取的数据为 N*256（N 代表采样点数，256 对应通道数），这意味着第 0 列数据对应 0 通道，第 1 列数据对应 2 通道，第 2 列数据对应 1 通道……此参数在 Probe_para 中可以查到。

当用户对整个编程熟悉了之后可以根据实际需求自定义换能器参数。通过修改 Probe_para.m 中的相关字段，可灵活调整阵元数量、间距及曲率半径等关键指标，从而适配不同几何结构的超声探头。

以一款相控阵换能器为例，假设换能器名称为 P4-1，pitch 为 0.295 mm，96 阵元，使用 260 接口，Pinmap 与附录 2 中不匹配，前 96 个阵元与物理通道的实际连接关系（记为数组 A）如下：

A = [22, 21, 20, 40, 23, 41, 45, 42, 44, 43, 46, 17, 16, 47, 12, 19, 18, 13, 14, 15, 48, 49, 53, 50, 52, 51, 8, 54, 55, 10, 11, 4, 5, 6, 7, 9, 3, 2, 1, 0, 39, 38, 37, 36, 32, 33, 34, 35, 92, 93, 94, 95, 91, 90, 89, 88, 127, 126, 125, 124, 118, 120, 121, 122, 123, 116, 117, 72, 73, 119, 76, 75, 77, 74, 78, 79, 112, 113, 114, 109, 108, 115, 80, 111, 110, 81, 84, 83, 85, 82, 86, 104, 87, 107, 106, 105]

索引从 0 开始，第 0 个阵元对应 22 物理通道，第 1 个阵元对应 21 物理通道……

对于此换能器需要重新调整映射表。

● 配置 Probe.tx_ele_map

换能器有效阵元为 0~95，A 中阵元数依次递增，默认发射通道映射为 0:255，因此前 96 个发射通道可直接映射为数组 A。由于映射表长度必须为 256，我们将剩余的 160 个闲置通道（排除 A 中已使用的数字）提取出来，按升序排列（顺序其实不影响，但升序便于管理）填充到 97~255 的位置。

最终 Probe.tx_ele_map 形式为：

[22, 21, 20, 40, 23 ... (A 的元素) ... 105, 24, 25, 26 ... (剩余 160 个元素) ... 254, 255]

● 配置 Probe.rx_ele_map

由于 Ultra-S 默认的接收通道映射存在错位规则（如 1 和 2 互换），需要基于默认的 Probe.rx_ele_map 进行逆向查表转换：

第 0 个阵元连接物理通道 22。查默认接收通道映射表可知，物理通道 22 的数据在第 21 列，故 rx_ele_map[0] = 21。

第 1 个阵元连接物理通道 21。查默认接收通道映射表可知，物理通道 21 的数据在第 22 列，故 `rx_ele_map[1] = 22`。

以此类推，对 `tx_ele_map` 中的所有 256 个通道进行错位转换。

最终需要用的配置如下：

```
Probe.name= 'P4-1';
Probe.type = 'phase';
Probe.element_pitch = 0.000295;
Probe.element_num = 96;
Probe.element_radius = 0;
Probe.element_num = 96;
Probe.image_start = 0;
Probe.tx_ele_map =
[22,21,20,40,23,41,45,42,44,43,46,17,233,234,235,236,.....246,247,248,249,2
50,251,252,253,254,255]
Probe.rx_ele_map =
[21,22,20,40,23,42,46,41,44,43,.....43,244,246,245,247,248,250,249,251,252,
254,253,255]
```

其余参数会自动计算；

不同的换能器在连接设备时，需要搭配相应的转接板。Ultra-S 系列为您提供常用的 260/408 接口转接板，260 接口转接板支持 128 阵元及以下阵元数量的换能器，408 接口支持 256 阵元及以下数量的换能器；单/少阵元换能器常用的 SMB 转接板，如有需要或者有其他转接板需求，请联系售后技术支持。

对于 260/408 接口转接板的 Pinmap 见附录 1。

3.2 LYConfig.AcqConfig.Tx

LYConfig.AcqConfig.Tx =

channel	double	通道数量
fs	double	采样率，单位 Hz
sos	double	声速，单位 m/s
focus_depth	double	聚焦深度
prf	double	脉冲重复频率，单位 Hz
steer	double	发射偏转角度，单位 °
Pulse	struct	发射波形相关参数
FirFilter	struct	滤波器相关参数

DownsampleM	double	降采样位置
DownsampleLoc	double	降采样位置
aperture	double	发射孔径个数
fs_win	cell	发射变迹窗函数
sequence	cell	发射序列事件
ATGC	struct	模拟增益相关参数
AFE	struct	模拟前端 AFE 参数
trigger_mode	double	内控/外控标志位
trigger_in	double	外控触发输入端口
trig_out	struct	内控触发输出参数
sampling_delay_sequence	double	采样延迟相关序列
sampling_delay_time	double	采样延迟序列时间，单位 us
mux	double	多路复用相关参数
mux_sequence	cell	多路复用发射序列事件
mux_delay	double	多路复用延迟

LYConfig.AcqConfig.Tx 包含了与信号发射和数据采集相关的核心配置。在实际编程中，部分参数可根据具体的应用场景按需省略。例如：当不使用外部触发模式时，无需配置触发输入端口（trigger_in）；在执行平面波成像时，无需设定聚焦深度（focus_depth）等。关于各参数的具体适用条件与配置方法，对应功能的示例脚本中均有详细说明。

3.2.1 channel

机器物理通道数，根据机器实际物理通道去配置；

3.2.2 fs

机器采样率，配置为 125000000，单位 Hz，固定不变；

3.2.3 sos

声速，单位 m/s；

3.2.4 focus_depth

聚焦深度，单位 m，由用户设定；

3.2.5 prf

脉冲重复频率，其取值范围为 10 至 62500，单位 Hz。

理想的脉冲重复频率根据探测深度来计算。在实际应用中，考虑到参数下发与机器响应需要耗时，系统在理论周期的基础上人为增加了 20 μ s 的时间余量。此外，PRF 还会受到系统传输带宽的严格限制。传输带宽与采样点数直接相关，用户在设置图像深度后会调用 SampleInfo 函数计算采样点个数，SampleInfo.m 函数是为了方便用户获取采样点数而定义的函数，用户也可自定义采样点数，需要遵循以下规则：采样点数量必须是 64 的倍数-1。

系统在完成参数设置后调用 helper_CalculatePRF 函数对设定的 PRF 进行合法性校验。如果设定值超出了系统允许的最大值，该函数会自动将其限制为当前允许的最大 PRF，并在 MATLAB 命令行窗口输出如下警告提示：

```
The prf you selected = 10000000 has exceeded the limit,
automatically adjust to max prf = 7210
```

3.2.6 steer

聚焦波、平面波发射波束的偏转角，单位：度；

3.2.7 Pulse/任意波

发射波形字段，这里对发射波形进行详细描述。

目前发射波形支持矩形波、任意波。

矩形波支持三电平、五电平。

对于三电平提供 4A 模式：A、B 两组脉冲驱动器并联同步输出，两路电流叠加，提升阵元激励功率，开启 4A 后底层等效为 3 电平脉冲架构，一般用于发射频率大于 35 MHz 的换能器。

最小电压 ± 1.5 V，最高电压 ± 100 V。

任意波最高电压支持 ± 90 V。

对于矩形波各通道支持发射频率、发射极性、发射延时、周期数、占空比单独调节；

对于任意波各通道支持发射电压、发射极性、发射频率、发射延时、周期数、占空比单独调节，支持时空加窗。

Pulse 中各字段含义如下：

pulse_type

定义了发射波形种类。

矩形波：'rect'，自定义各通道矩形波'rect_user'

任意波：'sin'，'chirp'，自定义各通道任意波'linear_user'

pulse_level_type

定义了电平类型，代表选用的电平。

矩形波三电平可以配置 {'A'}、{'B'}、{'D'}，A 和 B 没有区别，D 代表 4A 模式。

对于五电平需要 AB 组合在一起使用，比如 {'A', 'B'}、{'B', 'A'}。

具体配置要配合占空比 pulse_duration、极性 pulse_polarity、电压 pulse_voltage 一块使用。

任意波规定配置为 {'C'}。

pulse_frequency

定义发射频率，单位：Hz，元素个数必须与 pulse_level_type 一致。

pulse_voltage

定义单边发射电压，单位：V

pulse_num

定义发射波形周期数，0.5 或 1 的倍数。

pulse_duration

定义一个发射周期内各输出阶段的持续时间比例，所有阶段时间比例之和必须等于 1。

对于任意波模式，无需配置 pulse_duration，各阶段持续时间由用户自定义波形数据决定。

pulse_polarity

定义一个发射周期内各输出阶段的输出极性。1 代表正高压（+HV），-1 代表负高压（-HV），0 代表接地（GND/RTZ）。元素个数必须与 pulse_duration 一致，且每个元素只能取 1、-1 或 0。

通过组合极性、占空比和周期数，可灵活生成多种脉冲序列。

pulse_duration	pulse_polarity	输出波形
[0.5, 0.5]	[1, -1]	双极性脉冲方波（Non-RTZ）
[0.5, 0.5]	[1, 0]	单极性脉冲方波
[0.25, 0.25, 0.25, 0.25]	[1, 0, 0, -1]	双极性 RTZ 波形

说明：

- 对于矩形波，pulse_duration、pulse_polarity、发射频率及发射电压共同决定最终输出波形。
- 当配置为 sin、chirp 时，波形极性定义为正波形在前，负波形在后；
- pulse_polarity 与 pulse_duration 的元素个数必须一致。
- 系统按数组顺序依次输出各阶段波形。例如，当 pulse_duration 为 [0.5, 0.5]，pulse_polarity 为 [1, -1] 时，系统先输出半个周期的正极性高压，再输出半个周期的负极性高压。

pulse_win

对于任意波进行时空加窗。支持的窗函数如表 1 所示。

窗函数类型	代码标识	附加参数（coef）及其作用
矩形窗	'rect'	无（即均匀加权，主瓣最窄，旁瓣最高）
汉明窗	'hamming'	无（常用的平滑窗，有效压低旁瓣，但主瓣略宽）
汉宁窗	'hanning'	无
高斯窗	'gauss'	需要，用于定义标准差（控制钟形曲线的宽窄）
图基窗 / 锥形余弦窗	'tukey'	需要，定义余弦平滑过渡区域的比例（Taper ratio）
布莱克曼窗	'blackman'	无
布莱克曼-哈里斯窗	'blackmanharris'	无（提供极高的旁瓣衰减率）
切比雪夫窗	'cheb'	需要，定义旁瓣的特定衰减分贝值
凯塞窗	'kaiser'	需要，用于灵活调节主瓣与旁瓣的平衡

表 1 窗函数列表

pulse_cw_flag

定义了是否连续发射，1 代表发射连续波，0 代表脉冲波，需要注意发射波形中只能配置连续波或脉冲波，两者不能共存。

pulse_emit_dis

定义是否发射，当为 1 时即使配置发射波形也不发射；当为 0 时正常发射。通过该字段可灵活控制发射波形发射状态，默认状态为 0；

下面介绍几种发射波形配置，方便用户熟悉使用。

- 发射电压单边 60 V、发射周期 2.5、发射频率 5MHz、双极型脉冲方波，电平为 A

```
Pulse.pulse_type = 'rect';
Pulse.pulse_level_type = {'A'};
Pulse.pulse_cw_flag = 0;
Pulse.pulse_voltage = [60];
Pulse.pulse_num = 2.5;
Pulse.pulse_duration = [0.5,0.5];
Pulse.pulse_polarity = [1,-1];
Pulse.pulse_frequency = 5e6;
Pulse.pulse_emit_dis = 0;
```

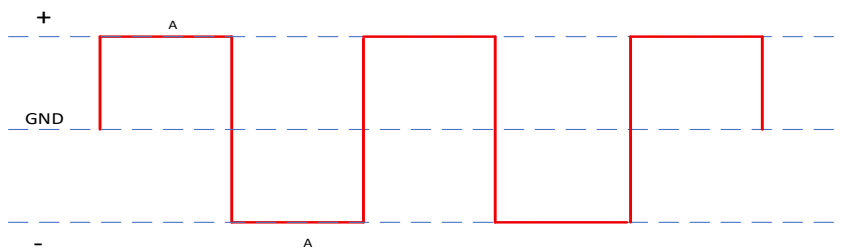


图 5 2.5 个周期、发射频率 5MHz 的双极型脉冲方波

如果想使用 4A 模式: `Pulse.pulse_level_type = {'D'};`

- 单极性脉冲方波、单边 60 V、负极性、1 个周期、电平 B

```
Pulse.pulse_type = 'rect';
Pulse.pulse_level_type = {'B'};
Pulse.pulse_cw_flag = 0;
Pulse.pulse_voltage = [60];
Pulse.pulse_num = 1;
Pulse.pulse_duration = [0.5,0.5];
Pulse.pulse_polarity = [-1,0];
Pulse.pulse_frequency = 5e6;
Pulse.pulse_emit_dis = 0;
```

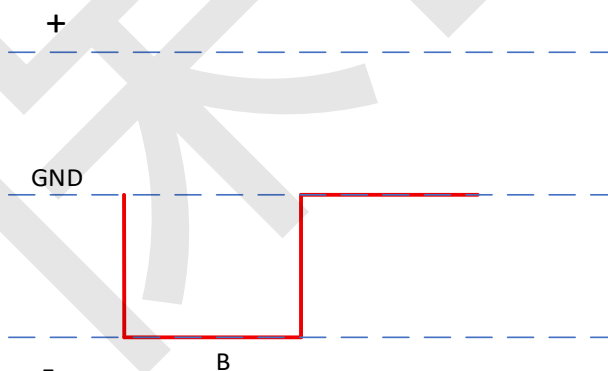


图 6 单极性脉冲方波

- 发射电压单边 60 V、发射周期 2.5、发射频率 5MHz、双极型脉冲方波，电平为 A，先发射 0.25 个周期的正极性，再发射 0.5 个周期接地，发射 0.25 个周期的负极性

```
Pulse.pulse_type = 'rect';
Pulse.pulse_level_type = {'A'};
Pulse.pulse_cw_flag = 0;
```

```
Pulse.pulse_voltage = [60];
Pulse.pulse_num = 2.5;
Pulse.pulse_duration = [0.25,0.25,0.25,0.25];
Pulse.pulse_polarity = [1,0,0,-1];
Pulse.pulse_frequency = 5e6;
Pulse.pulse_emit_dis = 0;
```

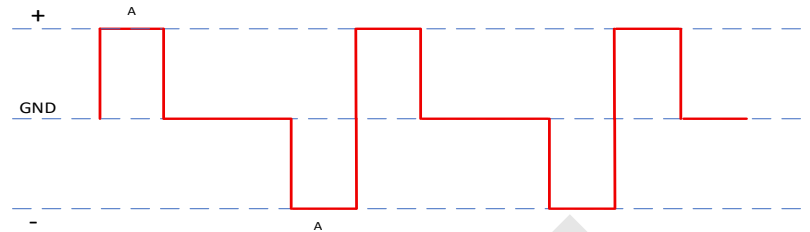


图 7 调节占空比的脉冲方波

- 5 电平模式：对于电平 A 发射电压单边 40 V，对于电平 B 发射电压单边 15 V，发射周期 3，一个周期内由 A 电平先发半个周期正负脉冲，再由 B 电平发半个周期正负脉冲，发射频率 5 MHz

```
Pulse.pulse_type = 'rect';
Pulse.pulse_level_type = {'A','A','B','B'};
Pulse.pulse_voltage = [30,30,15,15];
Pulse.pulse_num = 1;
Pulse.pulse_duration = [0.25,0.25,0.25,0.25];
Pulse.pulse_polarity = [1,-1,1,-1];
Pulse.pulse_frequency = 5e6;
Pulse.pulse_cw_flag = 0;
```

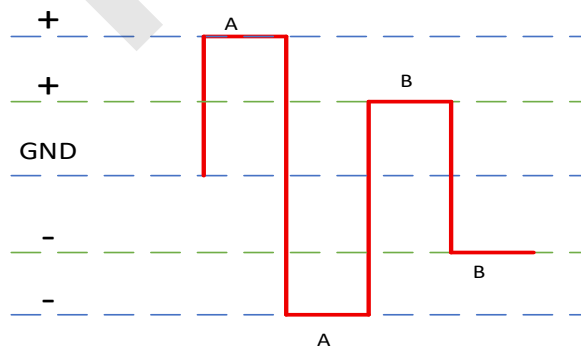


图 8 5 电平脉冲方波

- 各通道发射波形任意配置

对于各通道任意配置中的参数与前面使用方法一样，只需要对每一个通道单独进行配置；

这里需要说明由于电平只有两个，因此电压最多只能配置两种，当 A、B 两个电平的电压首次配置完成后，再次配置 A、B 电平电压不会生效，只会以第一次为准；

1.电平与 4A 模式互斥；即当系统已配置为 5 电平模式时，则无法启用 4A 模式，反之亦然。这一互斥规则确保了发射电路的工作状态唯一且稳定，避免因模式冲突导致电压输出异常或硬件损坏。

```
Pulse.pulse_type = 'rect_user';
% 配置各通道的发射波形频率
Pulse.pulse_frequency{1,1} = 5e6;
Pulse.pulse_frequency{2,1} = 7.5e6;
Pulse.pulse_frequency{3,1} = 8e6;
Pulse.pulse_frequency{4,1} = 5e6;
for i = 5:channel
    Pulse.pulse_frequency{i,1} = 5e6;
end
% 配置各通道的发射波形电平
Pulse.pulse_level_type{1,1} = {'A','A','B','B'};
Pulse.pulse_level_type{2,1} = {'B'};
Pulse.pulse_level_type{3,1} = {'A'};
Pulse.pulse_level_type{4,1} = {'A','B','A','B'};
for i = 5:channel
    Pulse.pulse_level_type{i,1} = {'A'};
end
% 配置各通道的发射波形电压
Pulse.pulse_voltage{1,1} = [40,40,20,20];
Pulse.pulse_voltage{2,1} = [20];
Pulse.pulse_voltage{3,1} = [40];
Pulse.pulse_voltage{4,1} = [40,20,40,20];
for i = 5:channel
    Pulse.pulse_voltage{i,1} = [40];
end
% 配置各通道的发射波形周期
Pulse.pulse_num{1,1} = 1.5;
Pulse.pulse_num{2,1} = 2;
Pulse.pulse_num{3,1} = 1;
```

```

for i = 4:channel
    Pulse.pulse_num{i,1} = 1.5;
end
% 配置各通道的发射波形持续时间
Pulse.pulse_duration{1,1} = [0.25,0.25,0.25,0.25];
Pulse.pulse_duration{2,1} = [0.5,0.5];
Pulse.pulse_duration{3,1} = [0.5,0.5];
Pulse.pulse_duration{4,1} = [0.25,0.25,0.25,0.25];
for i = 4:channel
    Pulse.pulse_duration{i,1} = [0.5,0.5];
end
% 配置各通道的发射波形极性
Pulse.pulse_polarity{1,1} = [1,-1,1,-1];
Pulse.pulse_polarity{2,1} = [1,-1];
Pulse.pulse_polarity{3,1} = [-1,1];
Pulse.pulse_polarity{4,1} = [1,-1,1,-1];
for i = 4:channel
    Pulse.pulse_polarity{i,1} = [-1,1];
end

```

● Sin 波

发射电压单边 20 V、发射周期 2、发射频率 5MHz、电平为 C，这里正波形在前，负波形在后

```

Pulse.pulse_type = 'sin';
Pulse.pulse_level_type = {'C'};
Pulse.pulse_voltage = [20];
Pulse.pulse_num = 2;
Pulse.pulse_frequency = 5e6;

```

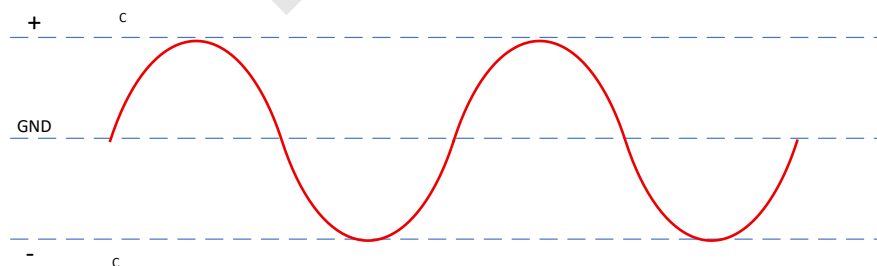


图 9 2 个周期的正弦波

发射电压单边 50 V、发射周期 10、发射频率 5MHz、电平为 C、gauss 窗系数为 2.8，这里正波形在前负波形在后

```
Pulse.pulse_type = 'sin';
Pulse.pulse_level_type = {'C'};
Pulse.pulse_voltage = [50];
Pulse.pulse_num = 10;
Pulse.pulse_frequency = 5e6;
Pulse.pulse_win = {'gauss', '2.8'};
```

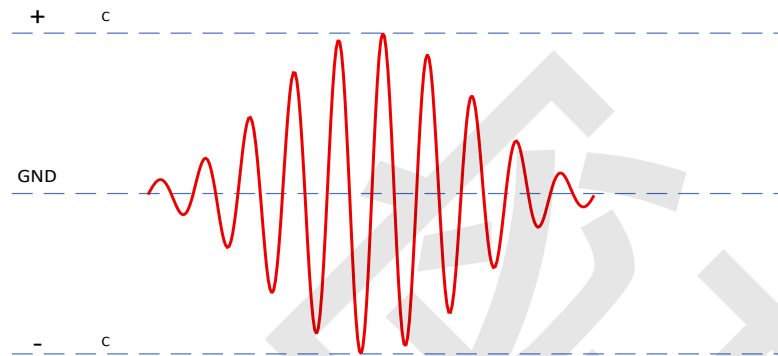


图 10 10 个周期加高斯窗的正弦波

● Chirp 波

发射电压单边 50 V、发射周期 25、发射频率 2 MHz~ 10MHz、电平为 C、hamming，这里默认正波形在前负波形在后

```
Pulse.pulse_type = 'chirp';
Pulse.pulse_voltage = [50];
Pulse.pulse_num = 25 ;
Pulse.pulse_frequency = [2e6, 10e6];
Pulse.pulse_win = {'hamming'};
```

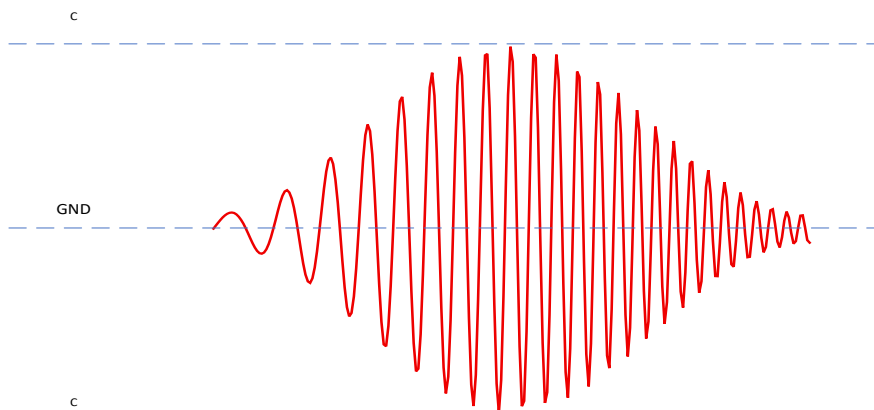


图 11 25 个周期加汉明窗的 chirp 波

对于 chirp 发射频率必须配置为两个频率，其余数量不支持；

- 各通道发射波形任意配置

对于各通道任意配置中的参数与前面使用方法一样，只需要对每一个通道单独进行配置。

这里需要说明，对于各通道发射波形周期数，自定义波形只能为整数，带有加窗的任意

波 Pulse.pulse num 只能为 1。

% 采样率

```
tx_wave fs = 250e6;
```

% 自定义任意波，各通道都需要配置

```
Pulse.pulse_type = 'linear_user';
```

%各通道发射电压

[illegible]

%各通道发射频率

```
Pulse.pulse frequency{1,1} = 7.5e6;
```

```
Pulse.pulse frequency{2,1} = 5e6;
```

```
Pulse.pulse frequency{3,1} = 4e6;
```

```
Pulse.pulse frequency{4,1} = 7.5e6;
```

```
for i = 5:channel
```

```
Pulse.pulse_frequency{i,1} = 2.5e6;
```

end

%各通道发射波形

```

for i = 1:2
    t = 0:1/tx_wave_fs:1/Pulse.pulse_frequency{i};
    Pulse.wave_shape{i,1} = sin(2*pi*Pulse.pulse_frequency{i}*t);
end
t = 0:1/tx_wave_fs:1/Pulse.pulse_frequency{3};
Pulse.wave_shape{3,1} = sin(2*pi*Pulse.pulse_frequency{3}*t);
Pulse.wave_shape{4,1} = Pulse.wave_shape{1,1};
for i = 5:channel-1
    t = 0:1/tx_wave_fs:1/Pulse.pulse_frequency{i}*5;
    Pulse.wave_shape{i,1}=
        sin(2*pi*Pulse.pulse_frequency{i}*t).*hamming(length(t))';
end
t = 0:1/tx_wave_fs:1/Pulse.pulse_frequency{channel}*3;
Pulse.wave_shape{channel,1} =
    sin(2*pi*Pulse.pulse_frequency{channel}*t).*hamming(length(t))';
%各通道发射周期，自定义波形只能为整数，带有加窗的任意波 Pulse.pulse_num 只能为 1
Pulse.pulse_num{1,1} = 3;
Pulse.pulse_num{2,1} = 2;
Pulse.pulse_num{3,1} = 1;
Pulse.pulse_num{4,1} = 3;
for i = 5:channel
    Pulse.pulse_num{i,1} = 1;
end

```

3.2.8 aperture

发射孔径（Transmit Aperture）是指在单次声波发射事件中，系统实际激活并参与发声的阵元集合及其总的物理跨度。它不仅代表了“当前有几个阵元在工作”，更是决定超声波束空间形态（聚焦能力、主瓣宽度、穿透深度）的最核心物理参数之一。此参数由用户分配。

发射孔径的大小影响超声波的传播特性和图像质量。它决定了波束的焦距、分辨率以及声束的扩展程度。

大的发射孔径能产生较窄的声束，可以获得更高的横向分辨率、发射更强的信号，超声波在传播过程中会受到物质的散射和衍射影响。较大的孔径可能导致声波在组织中的散射增加，尤其在浅表区域，可能会使得图像变得模糊，降低分辨率。

对于平面波、扩散波，一般激励所有阵元，发射孔径数等于阵元数。

对于合成孔径，依次只激励一个阵元，发射孔径数等于 1。

对于聚焦波，近场使用小孔径，远场使用大孔径。这里通过 F#控制发射孔径大小，下面对其介绍。

发射孔径配置参数有：聚焦深度（单位：m）以及对应的 F# (F number)，形式为：

深度：0 0.01 0.02

F#:4 3.8 3.6

数值之间必须有空格。

发射孔径计算公式为：

$$\text{发射孔径} = \frac{\text{聚焦深度}}{\text{F\#}}$$

最大孔径：发射孔径最大不能超过机器通道数；

最小孔径值必须 ≥ 1 。

根据深度和 F#的对应关系插值出用户设置的聚焦深度对应的 F#，根据发射孔径计算公式计算出 aperture 大小，这里提供了 Aperature 函数用于计算发射孔径大小。

代码配置示例：

```
tx_aper.max_Aper = 128; % 最大孔径
tx_aper.min_Aper = 32; % 最小孔径
tx_aper.fn_depth = [0 0.04]; %深度范围
tx_aper.fn_value = [4 4]; % 深度范围下对应的 F#
% 调用函数计算发射孔径大小
aperture = Aperature(tx_aper,focus_depth,probe,channel);
用户也可以根据需求自定义发射孔径大小;
```

3.2.9 sequence

sequence 是一个 cell 变量，每个 cell 内部存放 1×1 struct，对应一次发射序列事件。

对于聚焦波，图像是由一系列线拼成的图像，对于每一条线定义为一个发射事件。在这 N 次发射事件中需要提供每条线的发射起始位置、偏转角度、聚焦位置、发射孔径参数，根据上述参数确定发射延时。

对于平面波成像一般所有阵元全部发射，发射一次或 N 个角度，最后相干复合后形成一张图像，在这 N 次发射事件中，通过每次发射的偏转角度控制发射延时。

对于扩散波成像一般所有阵元全部发射，根据虚拟阵元位置发射一次或 N 次事件，最后相干复合后形成一张图像，在这 N 次发射事件中，通过每次虚拟阵元位置确定每次发射的延时。

这里依次以聚焦波、平面波、扩散波进行讲解。

● 聚焦波

以第 64 个事件为例：

LYConfig.AcqConfig.Tx.sequence{1,64}=

start_x	double	发射线起点 x 轴位置，单位 m
start_y	double	发射线起点 y 轴位置，单位 m
start_z	double	发射线起点 z 轴位置，单位 m
theta	double	发射线偏转角度，单位 rad
focus_x	double	聚焦点 x 轴位置，单位 m
focus_y	double	聚焦点 y 轴位置，单位 m
focus_z	double	聚焦点 z 轴位置，单位 m
aperture_center	double	发射孔径中心
aperture_length	double	发射孔径长度
active	double	激活的阵元位置
delay	double	发射延时

其中 start_x、start_y、start_z 为发射起始位置，单位 m，focus_x、focus_y、focus_z 为聚焦点起始位置，theta 为发射线偏转角度，单位 rad，theta 决定了发射波束偏转角，theta 是从 3.2.6 章节 steer 中获取的。

当用户配置完成探头、线密度、发射波束偏转角信息，Ultra-S 提供 FsJsLineLoc.m 函数计算发射、接收线起点位置。

对于线阵聚焦波成像，发射线有 N 条，对应位置根据探头长度平均分成 N 份，凸阵根据角度分成 N 份，相控阵按照用户设定的角度以原点为中心平均分配；

对于聚焦波成像默认接收线与发射线起始位置相同；

在获取发射线起始点位置后配合 3.2.7 中发射孔径大小确定发射阵元位置，aperture_length 代表发射孔径长度，由 3.2.7 中的 aperture 定义，aperture_center 代表孔径中心，由发射线起始点位置转化为阵元对应的位置，随后通过下面伪代码计算出发射阵元位置：

```
half_ap = floor((aperture_length- 1) / 2);
```

```
active_ele = (aperture_center - half_ap):(aperture_center + half_ap);
```

active 定义激活的阵元位置，active 为 1*N 维行向量，N 为通道数，当为 1 时为激活阵元，0 时关闭，上节获取的 active_ele 对应 active 的位置索引，假设 active_ele = 33:64，则 active 中的 33:64 位置为 1 其余为 0；

delay 用于设置对应发射孔径的发射延时，以实现波束偏转或聚焦，仅对 active=1 的阵元有效。对于 active=0 阵元，它们对应的通道发射使能信号（Enable）会被直接关闭，处于静默状态，分配的任何发射延时都不起作用。

- 平面波

LYConfig.AcqConfig.Tx.sequence{1,1}=

active	double	激活的阵元位置
delay	double	发射延时
aperture_center	double	发射孔径中心
aperture_length	double	发射孔径长度
theta	double	发射偏转角度，单位 rad

对于平面波仅需配置上面 5 个参数，参数定义与聚焦波一样。

对于平面波，发射延时、孔径计算在代码 Tx_beamform_PlaneWave.m 中。

- 扩散波

LYConfig.AcqConfig.Tx.sequence{1,1}=

active	double	激活的阵元位置
delay	double	发射延时
vir_x	double	虚拟阵元 x 轴位置
vir_y	double	虚拟阵元 y 轴位置
vir_z	double	虚拟阵元 z 轴位置
aperture_center	double	发射孔径中心
aperture_length	double	发射孔径长度

对于扩散波定义虚拟阵元位置及发射孔径即可计算发射延时，vir_x、vir_y、vir_z 代表虚拟阵元的位置，其余参数定义与聚焦波一样。

对于上述参数发射延时、孔径配置，Ultra-S 提供对应的函数用于计算，见表 2。用户只需要按照输入参数要求输入即可获取，当用户对脚本熟悉之后可以自行修改；

函数	支持
Tx_beamform_FocusWave	聚焦波，支持线阵、凸阵、相控阵探头
Tx_beamform_PlaneWave	平面波，支持线阵、凸阵、RCA 探头
Tx_beamform_DivergingWave	扩散波，支持线阵、相控阵探头
Tx_beamform_USCT	合成孔径，仅支持配置多路复用板 1024 阵元换能器

表 2 发射延时计算函数

通过控制发射延时可以控制发射波束类型，聚焦波通过设置焦点位置实现空间能量汇聚，平面波以固定角度发射覆盖全场，扩散波则利用虚拟焦点形成发散波前。

Tx_beamform_X 中发射延时定义在 delay 变量，大小为 1*N 维行向量，N 为通道数。

这里以线阵为例，对上述三种发射波束的延时计算进行说明。

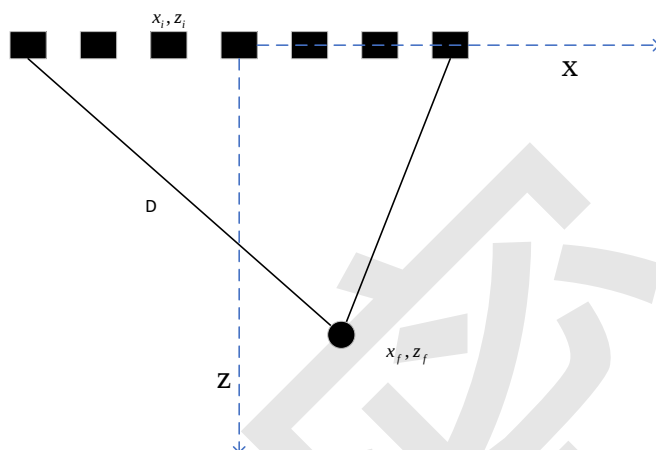


图 12 聚焦波发射延时

对于聚焦波，为确保各阵元 x_i, z_i 发射的声波能够在空间焦点 x_f, z_f 处同相叠加，需对各发射通道施加精确的时间延迟。第 i 个阵元的声波到达焦点所需的绝对飞行时间 (Time of Flight, TOF) t_i 取决于其与焦点的物理距离，计算公式为：

$$t_i = \frac{\sqrt{(x_i - x_f)^2 + (z_i - z_f)^2}}{c}$$

鉴于硬件因果性限制（延时必须 ≥ 0 ），为了实现同步聚焦，需以距离焦点最远、飞行时间最长的阵元作为零延迟参考基准。因此，各通道的实际发射延迟 $t_{\text{delay}}(i)$ 由下式确定：

$$t_{\text{delay}}(i) = \max(t_i) - t_i$$

该机制确保了距离焦点最远的阵元率先激发，而距离较近的阵元则根据其较短的声程依序延后激发，从而保证所有通道的声波波前能够完全同步抵达预设焦点。

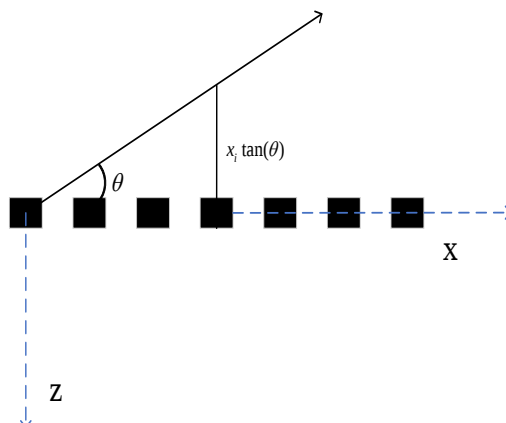


图 13 平面波发射延时

在平面波成像中，为使阵列发射的声波在空间中合成一个无焦点的平直波前，并以偏转角 θ （即声束传播方向与阵列法线 z 轴的夹角，或波前与 x 轴的夹角）进行传播，需要对各发射阵元施加呈线性分布的相对时间延迟。

对于位于坐标 x 处的第 i 个阵元，其为形成倾斜波前所需的理论发射延时 t_i 与其空间位置成正比，计算公式为：

$$t_i = \frac{x_i * \tan \theta}{c}$$

在实际工程实现中，由于偏转角 θ 可能为负，或者阵元坐标 x_i 跨越正负区间，理论延时 t_i 可能出现负值。鉴于硬件因果性限制（延时必须 ≥ 0 ），需对整体延时曲线进行平移补偿。通过选取所有通道中的最小理论延时作为零参考点，各通道的实际发射延时 $t_{\text{delay}}(i)$ 由下式确定：

$$t_{\text{delay}}(i) = t_i - \min(t_i)$$

该机制确保了实际延时值始终非负。这意味着“理论上需要最早激发的阵元（即 t_i 最小的阵元）”率先发射，其余阵元根据其空间位置依次延后激发，从而完美合成一个具有特定角度的连续平面波

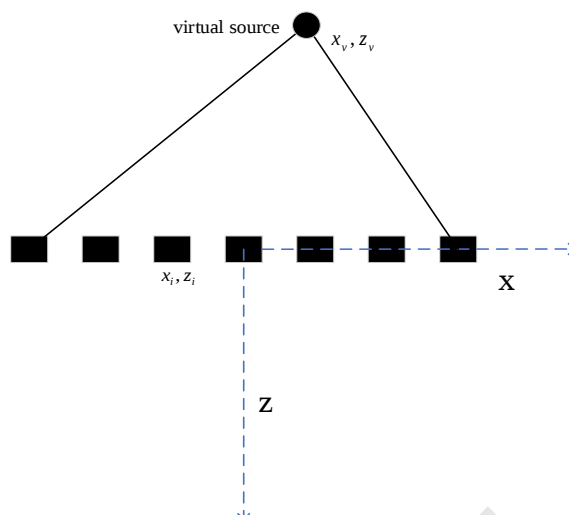


图 14 扩散波发射延时

对于扩散波为了在空间中产生一个向外不断扩散的宽视角波前，通常引入“虚拟点源（Virtual Source）”模型。通过精确控制各阵元的发射时序，使得阵列合成的波前在物理上表现得如同是从阵列后方某一点（即虚拟点源）辐射而出的球面波。

设虚拟点源的空间坐标为 x_v, z_v ，第 i 个阵元的坐标为 x_i, z_i 。声波从该虚拟点源传播至各阵元所需的理论绝对飞行时间 t_i 与它们之间的空间直线距离成正比，计算公式为：

$$t_i = \frac{\sqrt{(x_i - x_v)^2 + (z_i - z_v)^2}}{c}$$

扩散波要求距离虚拟点源最近的阵元率先激发，较远的阵元随波前的扩展延后激发。

基于物理硬件的因果性限制（实际发射延时必须 ≥ 0 ），需对理论飞行时间进行平移补偿。通过选取所有通道中的最小理论飞行时间 $t_{\min}$ （即距离虚拟点源最近的阵元）作为零延迟参考基准，各通道的实际发射延时 $t_{\text{delay}}(i)$ 由下式确定：

$$t_{\text{delay}}(i) = t_i - \min(t_i)$$

该补偿机制确保了延时序列的非负性。在工程执行上，它保证了物理距离虚拟点源最近的中心区域阵元在0时刻首发发射，边缘阵元则依据其空间声程差依次延后激发。最终，各通道激发的子波在空间中完美包络，形成一个向外发散的宽场球面波。

3.2.10 AFE+ATGC

AFE的核心作用是作为探头（模拟世界）与FPGA/DSP（数字世界）之间的桥梁，将探头接收到的极其微弱、动态范围极大的模拟回波信号，经过放大、调理与滤波后，高保真地转换为数字信号。AFE通常包含低噪声放大器（LNA）、压控衰减器（VGA）、可编程增益

放大器（PGA）、抗混叠低通滤波器（LPF - Low Pass Filter）、输入阻抗（Input resistance）配置。

在 Ultra-S 中用户可以自由对这些参数进行编程，以匹配不同应用场景下的成像需求。

LYConfig.AcqConfig.Tx.AFE=

lna_gain	double	低噪声放大器，单位 dB
pga_gain	double	可编程增益放大器，单位 dB
aaf	double	抗混叠低通滤波器，单位 MHz
ipt_res	double	输入阻抗，单位 Ω

LYConfig.AcqConfig.Tx.ATGC=

depth	cell	深度，单位 dB
value	cell	对应深度下的 ATGC 值，单位 dB

LNA 通过 lna_gain 配置，第一级前置放大，决定系统全局信噪比（SNR）。在极低噪声系数下，对微伏级的原始微弱回波进行初步放大。可配置范围为 15dB 到 21dB，步进为 3dB，不配置此值时默认为 21dB

VCAT 通过 LYConfig.AcqConfig.Tx.ATGC 中的 depth、value 配置，depth 代表深度，单位 m，value 代表深度下对应的值，单位 dB，范围为-36dB~0dB，Ultra-S 中会根据用户设置的参数，自动把不同深度下的 ATGC 插值出来；超声回波信号随深度增大而不断衰减，通过 VCAT 调节不同深度下的增益，使超声回波信号强度在不同深度下保持一致，不配置此值时默认为 0dB

PGA 通过 pga_gain 配置，可配置范围为 18dB 到 27dB，步进为 3dB。PGA 负责将信号进一步放大至最佳水平，防止信号失真或过载。不配置此值时默认为 18 dB

PGA 内置了三阶抗混叠低通滤波器（LPF），抗混叠滤波器用于在模数转换前滤除高于奈奎斯特频率的高频噪声与杂散分量，防止采样后产生频谱混叠该 LPF 采用巴特沃斯特性设计，为差分型有源三阶滤波器，其-3 dB 截止频率为 10、15、20、30、40、50 或 60 MHz，不配置此值时默认为 30。

输入阻抗通过 ipt_res 配置，可选项为 50 Ω 、100 Ω 、200 Ω 、400 Ω 或 1500 Ω ，用于匹配换能器与前端电路的阻抗特性，推荐的最小设置值为：低噪声模式下输入阻抗为 50 Ω ，低功率模式下为 100 Ω ，不配置此值时默认为 50 Ω 。

上述参数共同构成了超声前端模拟链路的完整配置体系。

3.2.11 FirFilter+DownsampleM+DownsampleLoc

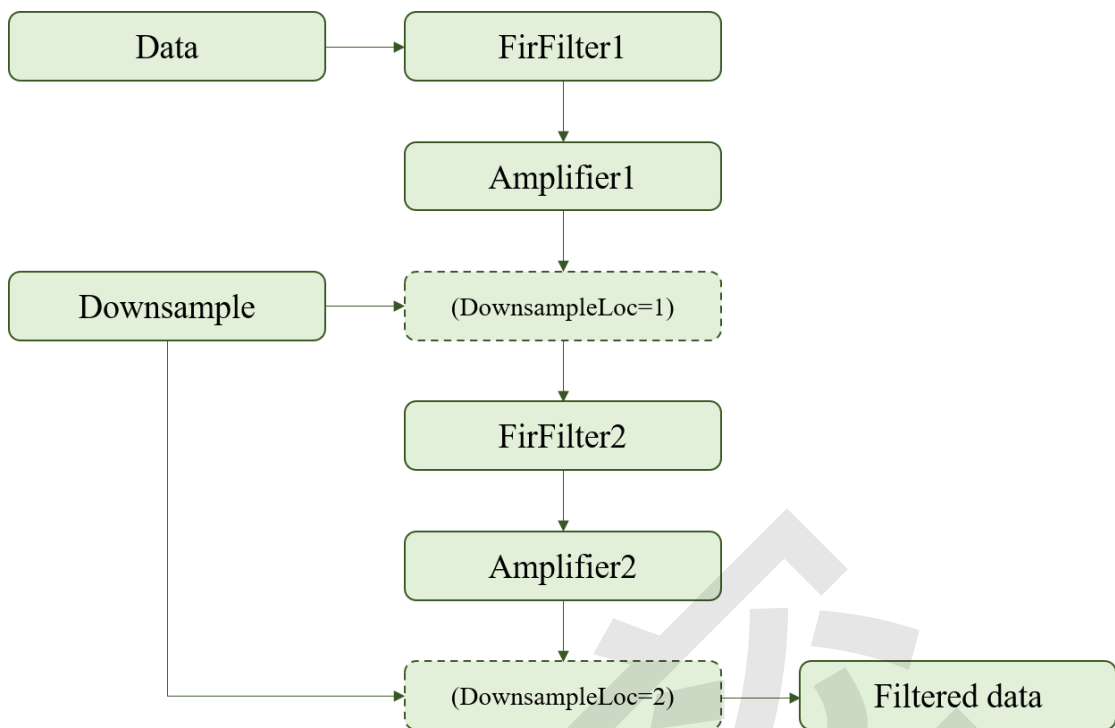


图 15 滤波器与降采样控制信号处理链路

Ultra-S 内部集成的高性能数字信号处理链路，由两级可编程 FIR 滤波器与整数降采样模块构成。该链路专为 16-bit 高动态范围数据流设计，通过精密的定点数乘加运算与动态截位策略，在重塑频率响应、抑制带外噪声的同时，有效控制数据带宽并确保信号不失真，整体信号处理链路见图 15。

LYConfig.AcqConfig.Tx.FirFilter=

Fir_coef1	double	第一级 Fir 滤波器
Fir_amp1	double	第一级 Fir 滤波器后放大倍数
Fir_coef2	double	第二级 Fir 滤波器
Fir_amp2	double	第二级 Fir 滤波器后放大倍数
DownsampleM	double	降采样倍数
DownsampleLoc	double	降采样位置

1. 两级 FIR 滤波器硬件架构与旁路控制

系统采用 16-bit 数据通路，两级滤波器内部均采用有符号 14-bit (Signed 14-bit) 系数量化精度，确保了优异的阻带衰减与通带平坦度。为满足不同应用场景，系统支持滤波器的灵活旁路 (Bypass) 机制：若用户仅需开启单级滤波或完全不进行滤波处理，无需为不使用的层级下发 Fir_coefX 系数，系统将自动对该级链路旁路。

第一级滤波器 (Fir_coef1)：65 抽头 (Taps)。可配置为低通 (LPF)、高通 (HPF)、带通 (BPF) 或带阻 (BRF) 类型。

第二级滤波器 (Fir_coef2)：64 抽头。可配置为低通 (LPF) 或带通 (BPF) 类型。

降采样与抗混叠路由控制 (Decimation & Anti-aliasing)

系统支持 1~16 倍整数降采样 (DownsampleM)。基于数字信号抗混叠原则，降采样节点前必须存在低通滤波器。DownsampleM 必须配置。系统提供 DownsampleLoc 控制字以灵活配置降采样的级联路由：

DownsampleLoc = 0 (旁路降采样)：强制 DownsampleM = 1，系统采样率为 125M。

DownsampleLoc = 1 (级联于一级滤波后)：第一级滤波器必须配置为低通类型以执行抗混叠，DownsampleM 设定范围为 1~16；

DownsampleLoc = 2 (级联于二级滤波后)：第二级滤波器必须配置为低通类型以执行抗混叠，DownsampleM 设定范围为 1~16。

建议奈奎斯特截止频率设计规范：降采样后的等效系统采样率 $f_{S_{new}}$ 必须满足 $f_{S_{new}} \geq 2 * (f_{tx} + BW)$ ，其中 f_{tx} 为发射中心频率， BW 为用户期望的带宽。

当使用 DownsampleM 时 SampleInfo 中的采样点数量会根据降采样倍数进行降低；

2. 乘加位宽扩展与动态截位机制 (Bit-growth & Dynamic Truncation)

这是滤波器配置的核心环节。在定点数运算逻辑中，16-bit 输入数据与有符号 14-bit 系数相乘，单次乘积位宽将扩展为 30-bit。经过 60+ 抽头的累加运算，内部累加器 (Accumulator) 将维持极宽的物理位宽。

为适配下一级的 16-bit 接口，系统通过参数 Fir_ampX (X=1,2，分别对应第一、第二级滤波器的输出) 提供动态截位窗口。若用户未显示配置该参数，系统默认设定 Fir_ampX = 0。

Fir_ampX (配置范围：0~16) 本质上控制的是 16-bit 数据截取窗口向低位的偏移量 (Bit Slicing Offset)。在算法等效上，此偏移操作等同于对信号施加 $2^{F_{ir_ampX}}$ 倍的数字增益：

Fir_ampX = 0 (系统默认 / 基准截位 / 无额外增益)：

系统直接截取滤波器内部宽位宽累加器的高有效位区间 (MSB，如内部设定的基准 [30:15])。此时输出数据未发生额外缩放。

Fir_ampX = 1 (增益 2 倍)：

截位窗口整体向低位滑动 1 个比特 (如移至 [29:14])。在算法上等效于将原始输出结果逻辑左移 1 位，即幅度放大 2 倍。

Fir_ampX = N (增益 2^N 倍)：

截位窗口向低位滑动 N 个比特，N 取 0~16；

不同类型的滤波器及系数归一化方式会导致滤波后的信号能量产生显著波动。合理配置 Fir_ampX，可将有效信号区间完美“套”入 16-bit 输出窗口：

若截位窗口过高（Fir_ampX 偏小），微弱回波信号将掉入低位被丢弃，导致量化噪声增加、精度损失。

若截位窗口过低（Fir_ampX 偏大），强回波信号将突破 16-bit 窗口上限，导致数值溢出。

在实际参数整定过程中，建议根据实际图像配置 Fir_ampX 截位窗口位置，以最大化系统的动态范围（Dynamic Range）。

下面代码是降采样 2 倍，先对信号进行高通滤波然后再进行低通滤波降采样示例

```
Sysrate = 125e6;
DownsampleM = 2;
DownsampleLoc = 2;
FirFilter.Fir_coef1 = fir1(64, 1/(sysrate/2/1e6), 'high', hamming(65));
FirFilter.Fir_amp1 = 8;
FirFilter.Fir_coef2 = fir1(63, 20/(sysrate/2/1e6), 'low', hamming(64));
FirFilter.Fir_amp2 = 7;
```

3.2.12 fs_win

在超声成像系统中，发射变迹（即在发射端进行孔径加权）是一种优化波束形态、降低图像旁瓣的关键技术。抑制旁瓣可以有效减少周边强反射体对目标区域的干扰，从而显著提升图像的对比度分辨率。

Ultra-S 系统内置了发射变迹功能，根据发射硬件的实现机制，提供以下两种标准实现方式以及一种自定义模式：

1. 电压幅度变迹 (Amplitude Weighting)

- 适用条件：仅支持任意波，底层参数 pulse_type 必须设置为 'sin' 或 'chirp'
- 实现原理：所有通道保持相同的脉冲波形，但独立调节输出的峰值电压。直接通过控制每个通道的激励电压幅度来实现加权。

配置逻辑：通道实际电压计算公式为

$$vol_i = w_i * vol$$

其中 vol_i 为各通道实际电压， vol 为系统设定基准电压， w_i 为该通道窗函数系数

2. 脉宽变迹 (Pulse Width Weighting)

- 适用条件：支持脉冲方波，txwave.pulse_type 必须设置为 'rect'（不支持 5 电平）。

- 实现原理：在保持各通道峰值电压不变的前提下，通过改变数字发射序列的占空比（Duty Cycle / 脉冲宽度）来等效控制输出功率。
- 配置逻辑：由于换能器主要响应激励信号的基波能量，脉宽参数 $pulse_width_i$ 与理论窗函数系数 w_i 之间需要遵循反正弦映射关系，以严格匹配第一谐波的振幅：

$$pulse_width_i = \frac{2}{\pi} * \sin^{-1} w_i$$

3. 自定义发射变迹 (Custom Apodization)

如果标准窗函数无法满足特定研发需求，Ultra-S 系统允许用户在发射波形配置中启用完全自定义的变迹方案：

- 自定义脉宽：将 `txwave.pulse_type` 设为 'rect_user'。启用后，系统将放权于用户，允许逐个通道自定义方波的脉宽参数。
- 自定义电压：将 `txwave.pulse_type` 设为 'linear_user'。启用后，用户可以逐个通道自定义任意波的输出电压幅度，实现高度灵活的孔径加权控制。

支持的窗函数 (Window Functions)

系统内置了多种经典窗函数，以满足不同探头和临床应用对“主瓣宽度（横向分辨率）”与“旁瓣电平（对比度）”的折中考量需求。

窗函数类型	代码标识	附加参数 (coef) 及其作用
矩形窗	'rect'	无（即均匀加权，主瓣最窄，旁瓣最高）
汉明窗	'hamming'	无（常用的平滑窗，有效压低旁瓣，但主瓣略宽）
汉宁窗	'hanning'	无
高斯窗	'gauss'	需要，用于定义标准差（控制钟形曲线的宽窄）
图基窗 / 锥形余弦窗	'tukey'	需要，定义余弦平滑过渡区域的比例 (Taper ratio)
布莱克曼窗	'blackman'	无
布莱克曼-哈里斯窗	'blackmanharris'	无（提供极高的旁瓣衰减率）
切比雪夫窗	'cheb'	需要，定义旁瓣的特定衰减分贝值
凯塞窗	'kaiser'	需要，用于灵活调节主瓣与旁瓣的平衡

代码配置示例

Matlab

% 示例：配置发射孔径使用高斯窗进行加权

```
fs_win = {'gauss', '1.2'};
```

```
LYConfig.AcqConfig.Tx.fs_win = fs_win;
```

配置解析：

- 上述代码指定系统当前使用的是高斯窗（Gaussian Window）进行发射孔径的变迹处理。
- 参数 '1.2' 即为该高斯窗的调节系数（coef），通常代表分布的标准差等缩放因子。该数值越小，高斯曲线越陡峭，中心阵元能量越集中；数值越大，能量分布在整个孔径上越平缓。

3.2.13 trigger_mode

trigger_mode 参数代表触发模式，用于控制系统的数据采集触发方式。

trigger_mode=0, 当设备运行时自行发射与接收数据，称为内控/内触发模式；trigger_mode=1 当设备根据外部其它设备进行发射与接收数据，称为外控/外触发模式。

当用户需要设备与其它设备进行联动，可使用内控+触发输出或者使用外控模式。

trigger_mode 参数用于控制系统数据采集的触发机制：

0（默认）：内控 / 内触发模式。设备按照内部时钟和设定的时序，自主循环执行发射与接收任务。

1：外控 / 外触发模式。设备处于从机状态，依赖外部设备输入的硬件电平信号来触发发射与接收操作。

应用提示：当需要将 Ultra-S 系统与其他外部设备进行时序联动时，可以灵活配置为“外控模式（接收外部指令被动工作）”或“内控模式 + 触发输出（系统主动工作并同步唤醒外部设备）”。

3.2.14 trigger_in

trigger_in 为触发输入端口，当启用外控模式（trigger_mode = 1）时，必须配置此参数。

- 端口映射：设备背部面板提供 5 个物理端口（取值范围 1 至 5），可任选其一作为触发输入源。系统在同一时刻仅响应 1 个端口的输入信号。
- 信号要求：系统接收的标准触发信号为 5V TTL 电平，硬件通过上升沿进行识别，要求输入脉宽为 1 μ s。

程序下发发射序列后，系统会挂起并持续等待。每检测到 1 个有效触发输入，系统便执行 1 条发射序列（即完成一次完整的收发事件）。

注意：在外控模式下，脚本中配置的 PRF（脉冲重复频率）或 PRT（脉冲重复时间）将失效。系统的实际采集频率完全由外部触发信号的输入频率决定（例如：外部每秒输入 1000 个触发信号，系统即每秒采集 1000 次）。但外部触发频率不得超过系统底层带宽所允许的最大物理发射频率。

3.2.15 trig_out

LYConfig.AcqConfig.Tx.trig_out=

trigger_out	double	激活的触发输出端口
trigger_out_width	double	激活的触发输出端口对应的输出脉宽，单位 μs
trigger_out_delay	double	激活的触发输出端口输出的相对延迟
trigger_out_sequence	double	触发输出的发射序列

LYConfig.AcqConfig.Tx.trig_out 结构体用于配置系统的主动触发输出参数，该功能仅在内控模式下支持，输出的触发信号同样为 5V TTL 电平。其包含以下核心字段：

trigger_out (double): 激活的触发输出端口号，取值范围为 1 至 5，对应设备背部的 5 个物理端口。

trigger_out_width (double): 输出信号的脉冲宽度（单位： μs ）。限制条件：设定的脉宽不得超过相邻两次发射事件的时间间隔。

trigger_out_delay (double): 触发输出信号的相对延迟时间（单位： μs ）。取值范围为 -32000 至 32000。

- 0：与超声发射时刻严格同步。
- 正数：滞后于超声发射时刻输出。
- 负数：提前于超声发射时刻输出。

trigger_out_sequence (double): 指定在哪些具体的发射序列事件中激活触发输出功能。

配置规则：系统默认不开启触发输出。如需启用，必须至少同步配置 trigger_out、trigger_out_width 和 trigger_out_delay 三个参数，且这三个数组的长度（即激活的端口数量）必须保持一致。

代码配置示例：

```
Trig.trigger_out = [1 2];
Trig.trigger_out_width = [20 30];
Trig.trigger_out_delay = [0 10];
Trig.trigger_out_sequence = [1 2];
LYConfig.AcqConfig.Tx = Trig;
```

配置解析：上述代码同时激活了设备的第 1 和第 2 端口。其中，第 1 端口的输出脉宽为 20 μs ，无延迟（0 μs ，与超声同步）；第 2 端口的输出脉宽为 30 μs ，延迟 10 μs 后输出。同时，trigger_out_sequence=[1,2] 意味着仅在系统执行第 1 条和第 2 条发射序列时，这两个端口才会产生对应的触发输出。

3.2.16 sampling_delay_sequence+sampling_delay_time

这两个参数一般用于光声设置，由于信号产生于激光发射时（Q 开关），所以采集信号的延迟也需要与 Q 的延迟一致(或接近,可能激光器有些许误差),具体为 sampling_delay_time, 单位为微秒。sampling_delay_time 对应的序列通过 sampling_delay_sequence 进行配置。

此组参数通常专用于光声成像 (PA, Photoacoustic Imaging) 场景。在光声应用中，超声信号实质上是由外部激光器发射（通常对应激光器的调 Q 开关 / Q-switch）激发的。为了精准捕获有效回波，超声接收端（ADC）的数据采集延迟必须与 Q 开关的实际出光延迟保持高度一致（也可用于补偿激光器自身固有的微小出光误差）。

sampling_delay_time: 具体设定的数据采集延迟时间（单位：μs）。

sampling_delay_sequence: 指定上述延迟时间生效的特定序列，确保系统在正确的序列事件启动 ADC 采样。如用户配置了 1 条序列用于光声和 N 条序列用于超声，则仅需在光声序列中调整数据采集延迟时间，此时此参数为光声序列的序号。

3.2.17 mux + mux_sequence + mux_delay

超声探头阵元数量(如 512 或 1024)或超声探头数量有时远超系统物理通道数(如 256), 受限于成本、功耗及带宽，无法做到全并行连接；因此必须通过多路复用（MUX）将多个阵元时分复用到同一通道上，以硬件复杂度换取通道规模与系统成本的平衡。

这三个参数在《用户入门手册》中进行了说明，只有在使用多路复用板时才会启用，默认不需要配置。仅在使用多路复用转接板需要了解

3.3 ScanList

3.3.1 头信息

LYConfig.AcqConfig.ScanList									
LYConfig.AcqConfig.ScanList									
字段	Framestart	Frameend	Packsize	SIn	ModeType	Pulse_ptr	SampleN	Prt	
1	1	0	0	0	0	0	3711	125000	
2	0	0	0	1	0	0	3711	125000	
3	0	0	0	2	0	0	3711	125000	
4	0	0	0	3	0	0	3711	125000	
5	0	1	0	4	0	0	3711	125000	
6									

图 16 ScanList 字段组成

在 ADC 数据上传至上位机的过程中，系统必须对数据进行严格的校验，以确保上传数据的完整性与正确性。为了实现底层硬件与上层软件的精确同步，FPGA 会在每一次发射事

件产生的数据前端，自动增加一段专属的头信息（Header），这些字段在 ScanList 中进行设置，见图 16。包括：

Framestart（起始帧标志）：用于精准界定图像帧的起点。当该字段值为 1 时，代表当前数据属于该帧的第一次发射序列事件。

Frameend（结束帧标志）：用于界定图像帧的终点。当该字段值为 1 时，代表这是该帧的最后一次发射序列事件。

中间序列判定：如果 **Framestart** 和 **Frameend** 的值同时为 0，则系统明确判定当前数据属于一帧图像扫描过程中的中间序列事件。

Sln（线号 / 序列号）：代表当前数据对应的是第几次发射序列事件，通常与空间物理扫描线一一对应。

ModeType（成像类型）：用于指示当前的声学扫描模式。目前，除了 RCA 探头需要被特定设置为 4 之外，其余常规成像探头默认均配置为 0。

SampleN（采样点数量）：指明当前这条接收线上实际包含的有效采样点总数。

PackSize（脉冲反转谐波成像标志）：此字段主要服务于脉冲反转谐波成像（Pulse Inversion Harmonic Imaging, PIHI）这一高级算法应用。在谐波成像模式下，系统会在完全相同的空间物理位置，连续发射两次波形相位相反的激励脉冲，以抵消基波并增强谐波信号。当 **PackSize** 配置为 0 时，系统将按照预设的正常波形进行发射；当 **PackSize** 配置为 1 时，系统发射的波形将发生脉冲反转（相位翻转）。需要特别强调的是，由于这连续两次的波形发射是针对同一个目标位置进行的，因此它们所对应的线号（**Sln**）是完全相同的。图 17 是一个脉冲反转谐波成像的部分扫描序列：

字段	Framestart	Frameend	Packsize	Sln	ModeType	SampleN	Pulse_ptr	Prt
1	1	0	0	0	0	3647	0	41667
2	0	0	1	0	0	3647	0	41667
3	0	0	0	1	0	3647	0	41667
4	0	0	1	1	0	3647	0	41667
5	0	0	0	2	0	3647	0	41667
6	0	0	1	2	0	3647	0	41667
7	0	0	0	3	0	3647	0	41667
8	0	0	1	3	0	3647	0	41667
9	0	0	0	4	0	3647	0	41667
10	0	0	1	4	0	3647	0	41667
11	0	0	0	5	0	3647	0	41667

图 17 脉冲反转谐波成像 ScanList

3.3.2 时序控制

在 ScanList 中 **Prt** 字段决定了相邻两次发射事件之间的时间跨度。系统底层采用 125MHz 的高频时钟进行高精度量化。

在脚本中默认自动计算，根据配置的脉冲重复频率（prf），通过公式 $Prt = \text{round}(125e6 / \text{prf})$ 精确计算出对应的 Prt 值。

虽然在默认脚本生成的常规序列中，所有发射事件的 Prt 都是恒定相同的，但在高级研发和实际使用过程中，用户完全可以根据需求，在每一次扫描序列中将 Prt 灵活设置为不同的非等间隔值。Ultra-S 硬件系统在调度发射事件间隔时，一切都严格以 ScanList 列表中配置的 Prt 数值为绝对准则。

3.3.3 波形调度

Pulse_ptr（发射波形地址指针），Pulse_ptr 赋予了系统在单帧图像扫描中进行动态波形切换的能力。该字段代表发射波形的地址，在基础脚本中默认值为 0。

当用户需要针对不同的发射事件设置不同的发射波形参数时，可以通过修改 Pulse_ptr 来进行独立分配。

应用案例配置解析：假设系统当前需要执行 5 个发射事件。

默认配置组（Pulse_ptr = 0）：第 1、2、4、5 个事件的发射波形参数为 5MHz 的中心频率、2 个周期的方波，并使用矩形窗进行发射变迹。

特殊配置组（Pulse_ptr = 1）：第 3 个事件需要切换策略，其发射波形调整为 5MHz、1 个周期的方波，并将发射变迹替换为汉明窗（Hamming）。

通过在 LYConfig.AcqConfig.Tx.Pulse 和 LYConfig.AcqConfig.Tx.fs_win 结构体数组中提前定义好这两种波形，并在 ScanList 中将第 3 个事件的 Pulse_ptr 指向 1，即可实现底层的毫秒级无缝切换，极大提升了超声序列设计的自由度。

LYConfig.AcqConfig.ScanList									
字段	Framestart	Frameend	Packsize	SIn	ModeType	Pulse_ptr	SampleN	Prt	
1	1	0	0	0	0	0	3711	125000	
2	0	0	0	1	0	0	3711	125000	
3	0	0	0	2	0	1	3711	125000	
4	0	0	0	3	0	0	3711	125000	
5	0	1	0	4	0	0	3711	125000	

图 18 动态发射波形 ScanList

扫描序列及代码配置为：

```
txwave{1}.pulse_type = 'rect';
txwave{1}.pulse_level_type = {'A'};
txwave{1}.pulse_cw_flag = 0;
txwave{1}.pulse_voltage = [20];
txwave{1}.pulse_num = 2;
```

```
txwave{1}.pulse_duration = [0.5,0.5];
txwave{1}.pulse_polarity = [1,-1];
txwave{1}.pulse_frequency = 5e6;
txwave{1}.pulse_emit_dis = 0;
txwave{2}.pulse_type = 'rect';
txwave{2}.pulse_level_type = {'B'};
txwave{2}.pulse_cw_flag = 0;
txwave{2}.pulse_voltage = [30];
txwave{2}.pulse_num = 1;
txwave{2}.pulse_duration = [0.5,0.5];
txwave{2}.pulse_polarity = [1,-1];
txwave{2}.pulse_frequency = 4e6;
txwave{2}.pulse_emit_dis = 0;
fs_win{1} = { 'rect' };
fs_win{2} = { 'hamming' };
LYConfig.AcqConfig.Tx.Pulse = txwave;
LYConfig.AcqConfig.Tx.fs_win = fs_win;
```

这里需要注意，当存在不同发射波形时，对于方波有以下需要注意：

pulse_level_type 为同一个值但电压不同时，以最大值电压为准；

pulse_level_type 中设置了一个或多个 D 时，以 D 中的最大电压为准；

3.4 LYConfig.AcqConfig.Rx

LYConfig.AcqConfig.Rx=

fs	double	ADC 采样率，单位 Hz
ImageDepth	double	用户设定图像深度，单位 m
sample_num	double	用户设定采样点数量
sos	double	声速，单位 m/s
Revpt	double	实际采样点数量，单位 m
Revdepth	double	实际图像深度，单位 m
frameN/timeN	double	采集数据帧数/采集时长（单位 s）
aperture	struct	接收孔径参数
sequence	cell	接收事件相关参数
NumsPerFile	double	bin 文件中保存的序列个数

LYConfig.AcqConfig.Rx 参数包含与接收、采集相关的配置；其中：

3.4.1 fs

接收到的 ADC 数据采样率，单位 Hz，当用户启用降采样后脚本会自动根据用户设置的降采样数量进行计算；

前端模数转换器（ADC）对回波信号的采样频率，单位 Hz。当用户在系统中启用“降采样”功能后，底层脚本会自动根据设定的抽取因子重新计算并更新该值，以降低数据传输和处理压力。

3.4.2 sos

超声波在目标介质中的传播速度。此参数必须与发射配置(Tx)中的 sos 保持绝对一致，否则会导致波束合成延时计算错误，引发图像散焦或几何畸变。

3.4.3 ImageDepth / sample_num

用户期望观察的图像深度，单位 m，如果用户设定 ImgShowflag=1，会显示用户设定对应深度下的图像； sample_num 根据设定的成像深度（ImageDepth）、声速（sos）和采样率（fs）自动计算得出

$$\text{round}\left(\frac{\text{ImageDepth}}{\text{sos}/\text{fs}/2}\right)$$

3.4.4 Revdepth / Revpt

Revdepth 对应实际采集深度，Revpt 为 Revdepth 下的真实采样点数

在真实的超声硬件系统中，ADC 采集的初始阶段会受到 T/R（收发）开关切换的瞬态干扰、压电换能器的机械振铃效应以及接收模拟前端（AFE）恢复响应时间等影响。由于近场存在这段“盲区干扰”，如果直接按期望图像深度对应的采样点采集，会导致浅层图像被干扰信号覆盖，且无法对应真实的组织深度。因此，系统人为增加了 0.5 cm 的额外补偿深度。

Revpt 采样点数量小于 63 或者大于 8191 个点系统数据上传过程中会不稳定，建议采样点数量在 63~8191 的范围内；

Revpt 与 3.3 节 ScanList 中的 SampleN 大小相同。

3.4.5 frameN/timeN

用于定义连续采集模式下的数据规模

frameN（采集/显示帧数）：定义需采集或显示的完整图像帧数。如果指定帧数，总 ADC 帧数 = frameN × 发射序列长度；

timeN（采集/显示时长）：基于时间维度的采集或显示限制。根据 timeN 和脉冲重复频率（prf），计算出理论上所需的原始帧数。用理论帧数除以发射序列长度，若设定的时间结束时刚好不满足整数帧，系统为保证数据完整性，会强制补齐当前帧的数据。

3.4.6 fetch_frequency/NumsPerFile

fetch_frequency 和 NumsPerFile 都是可以控制数据接收和存储的参数，两者互斥。如果不需要特别指定，可以将二者注释，程序将采用默认参数计算。

以 prf=1000，发射序列长度=5 举例说明

指定 fetch_frequency 时：

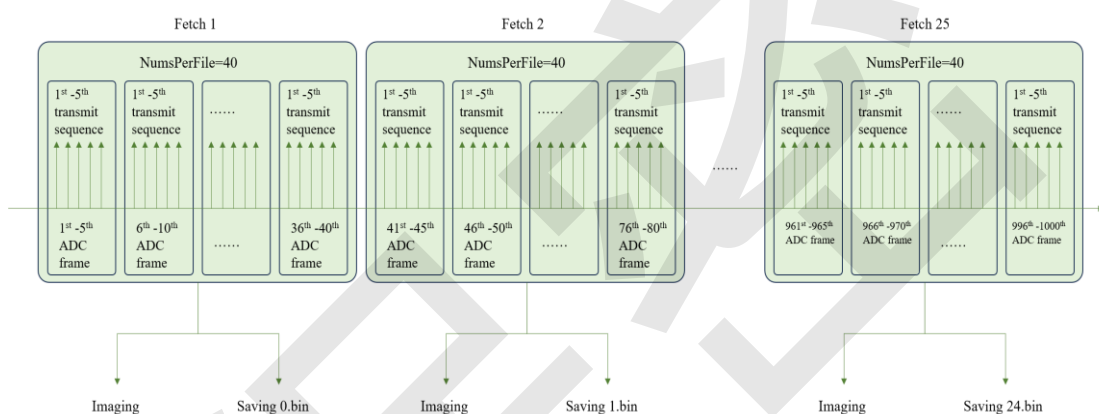


图 19 fetch_frequency 配置时数据接收和存储

如上图 19 所示，如果指定 fetch_frequency=25，则程序计算出每一次接收 ADC 数据帧数 = 1000/25=40，包含 40/5=8 个完整发射序列。如果开启了数据存储，则会相应被存储到 bin 文件中。特殊地，如果 prf/fetch_frequency 的结果不为发射序列长度的整倍数，则会取值相近的发射序列长度的整倍数，以便后续数据处理。

指定 NumsPerFile：

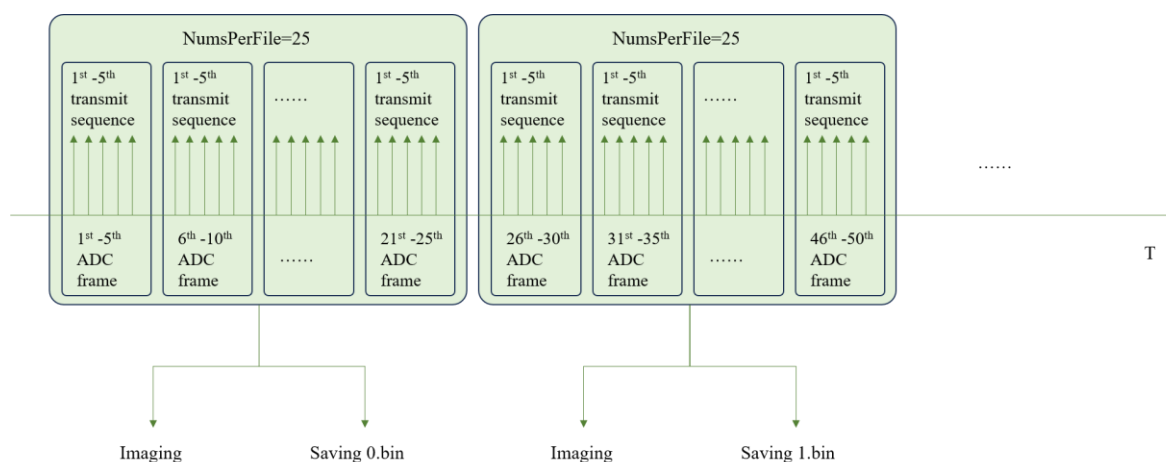


图 20 NumsPerFile 配置时数据接收和存储

如图 20 假定 NumsPerFile=25，则程序每一次接收 ADC 数据帧数=25，包含 25/5=5 个完整发射序列。如果开启了数据存储，则会相应被存储到 bin 文件中。在平面波和扩散波脚本中，如果 NumsPerFile 不为发射序列长度的整倍数，则会导致实时成像错误。

值得注意的是，由于程序读取的缓冲区的大小有限制，每一次取的数据量不能过大，如果发射序列长度较大，可能导致一次无法取到完整的一次发射序列的数据，此时程序会对 NumsPerFile 进行限制。

建议用户不要配置此值，由 Ultra-S 底层自动计算。

3.4.7 aperture

动态接收孔径的大小直接决定了探头在回波处理中的空间控制能力。

大孔径：聚焦范围变窄，横向分辨率显著提升，能接收更多回波能量（高信噪比）

小孔径：在浅层近场区域，为避免过大孔径引入离轴噪声和伪影，通常使用较小孔径进行限制。

提示：若用户仅需导出原始 RF 数据而不进行波束合成，此模块可不配置。

LYConfig.AcqConfig.Rx.aperture=

max_Aper	double	最大接收孔径
min_Aper	double	最小接收孔径
fn_depth	double	深度，单位 m
fn_value	double	深度下对应的 F#
js_win	double	接收窗函数系数，固定为 256

代码配置示例：

```
rev_aper.max_Aper = 128;
```

```
rev_aper.min_Aper = 48;
rev_aper.fn_depth = [0 0.04];
rev_aper.fn_value = [1.1 1];
rev_aper.js_win = hamming(256);
LYConfig.AcqConfig.Rx.aperture = rev_aper;
```

变迹窗 (js_win) 是为了对接收阵列的边缘通道进行平滑衰减，以此有效抑制图像的旁瓣伪影 (Sidelobe Artifacts)，提升图像对比度，仅在聚焦波中使用，支持矩形窗 rect、汉明窗 Hamming、汉宁窗 hanning、高斯窗 gauss。其余发射波束类型为矩形窗

3.4.8 sequence

sequence 是一个元胞数组 (cell)，内部存放的结构体数量与发射 (Tx) 事件数严格对应。它为后端的延迟累加 (DAS) 波束合成提供准确的空间坐标矩阵。

系统支持两种坐标系的波束合成参数映射：

对于笛卡尔坐标系包括：接收线起始位置、步进、cstartoffset

以第一条接收序列为例：

```
LYConfig.AcqConfig.Tx.sequence{1,1}=
```

start_x	double	接收线起点 x 轴位置，单位 m
start_y	double	接收线起点 y 轴位置，单位 m
start_z	double	接收线起点 z 轴位置，单位 m
step_x	double	接收线 x 方向步进距离，单位 m
step_y	double	接收线 y 方向步进距离，单位 m
step_z	double	接收线 z 方向步进距离，单位 m
x_num	double	接收线 x 方向波束合成点个数
y_num	double	接收线 y 方向波束合成点个数
z_num	double	接收线 z 方向波束合成点个数
theta_num	double	接收线角度方向波束合成点个数
cstartoffset	double	波束合成校正参数

其中 start_x、start_y、start_z 为接收线起始位置，单位 m，step_x、step_y、step_z 为接收线步进距离，x_num、y_num、z_num 为接收波束合成在各方向上的个数，上述参数是用于给接收波束合成提供位置信息。

以下图为例，蓝色线为接收波束合成方向，黑色点是波束合成点的位置、红色点是接收线起始位置，每一个波束合成点 i 的位置为

$$x_i = \text{start}_x + i * \text{step}_x$$

$$z_i = start_z + i * step_z$$

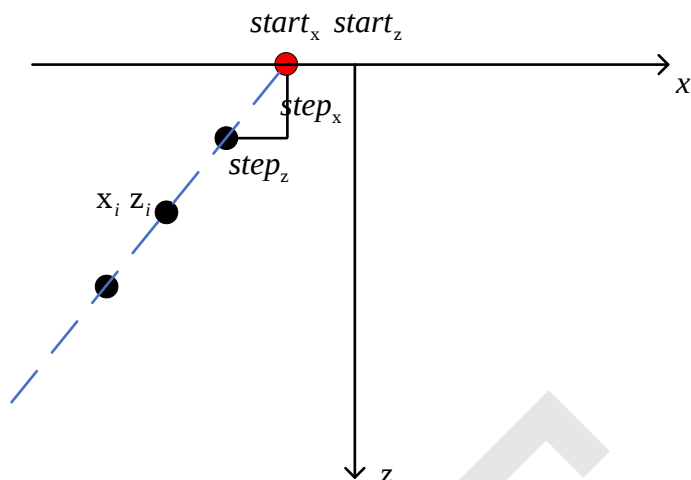


图 21 笛卡尔坐标系波束合成位置计算示意图

对于极坐标系包括：

LYConfig.AcqConfig.Tx.sequence{1,1}=

start_x	double	接收线起点 x 轴位置，单位 m
start_y	double	接收线起点 y 轴位置，单位 m
start_z	double	接收线起点 z 轴位置，单位 m
start_theta	double	接收线起始角度，单位 rad
x_num	double	接收线 x 方向波束合成点个数
y_num	double	接收线 y 方向波束合成点个数
z_num	double	接收线 z 方向波束合成点个数
theta_num	double	接收线角度方向波束合成点个数
step_x	double	接收线 x 方向步进距离，单位 m
step_y	double	接收线 y 方向步进距离，单位 m
step_z	double	接收线 z 方向步进距离，单位 m
step_theta	double	接收线角度步进，单位 rad
cstartoffset	double	波束合成校正参数

上述相同的参数信息与笛卡尔坐标系含义相同，对于极坐标成像额外增加了角度信息：起始角度 start_theta、角度步进 step_theta、角度方向波束合成点数 theta_num。

以下图为例，黑色框为波束合成区域，蓝色线为接收波束合成方向，黑色点是波束合成的位置、接收线位置的步进夹角为θ，每一个波束合成点i的位置为

$$\theta_i = \text{start}_\theta + i * \theta$$

$$z_i = \text{start}_z + i * \text{step}_z$$

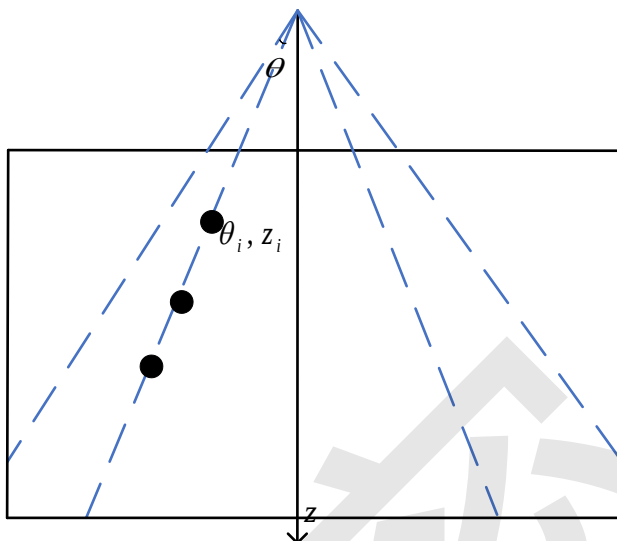


图 22 极坐标系波束合成位置计算示意图

cstartoffset: 这是一个针对发射孔径时间起点（Time Zero）的波束合成对齐校正参数。以聚焦波为例，边缘阵元先被激励（时刻 t_0 ），中心阵元最后被激励（时刻 t_1 ）以形成声学焦点。当系统开始统一接收数据时且接收线位于中间阵元时，于接收线上的所有点会存在一个相对延时 $t_1 - t_0$ 。**cstartoffset** 就是将这一时间差转化的对应点数偏移量。波束合成算法会在延时表中应用该偏移，以保证不同通道的数据在同一空间点上能够完美相位对齐。

4 LYConfig.AcqConfig.PostConfig

本配置模块主要用于定义系统中的信号后处理流程（包含 IQ 解调与低通滤波）。此处的参数调整仅用于优化实时成像的效果与图像质量，不会改变或污染已保存的原始射频数据。

LYConfig.PostConfig=

demo_depth	double	解调深度，单位 m
demo_value	double	深度对应下的解调频率，单位 Hz
filterdepth	double	滤波器深度，单位 m
filtercoef	double	深度对应下的截止频率归一化系数
filtertype	char	Fir 滤波器类型

filterorder double Fir 滤波器阶数

4.1.1 demo_depth

解调深度节点（单位：m），严格限 2 个元素。例如 [0, 0.04]，定义了动态解调变化的起始与终止深度。

4.1.2 demo_value

节点对应的解调频率（单位：Hz），严格限 2 个元素。例如 [7e6, 5e6]，表示在浅层(0m)使用 7MHz 解调以保分辨率，在深层(0.04m)降至 5MHz 以保信噪比。

4.1.3 filterdepth

滤波深度节点（单位：m），严格限 2 个元素。注意：目前系统仅第一个值生效，即当前为全深度统一滤波，尚未实现随深度动态变化的滤波。

4.1.4 filtercoef

归一化截止频率系数，严格限 2 个元素。注意：目前系统仅第一个值生效。该系数是按奈奎斯特频率（采样率的一半，fs/2）进行归一化的，取值范围为（0，1）。

4.1.5 filtertype

FIR 滤波器窗类型。指定滤波器设计的窗函数，决定了通带和阻带的波动特性。窗函数仅支持'hamming'、'hanning'、gaussian。

4.1.6 filterorder

FIR 滤波器阶数（抽头数）。阶数越大，滤波器的截止滚降越陡峭，但计算耗时和硬件资源消耗也随之增加。

● 解调机制与深度延长策略

组织成像（B-Mode）的动态解调：系统根据 demo_depth 和 demo_value 输入的两个点（如 0m->7MHz, 0.04m->5MHz）构建解调频率。如果用户设定的图像显示深度超过了第二个节点（例如深达 0.06m），系统不会截断，而是自动以 demo_value 的最后一个有效值（即 5MHz）向下做无限延长。

血流模式 (Color/Doppler)：如果是处理血流信号，解调频率绝对不能随深度动态变化！因为血流的速度信息强依赖于多普勒频移和严格的相位关系，动态改变解调频率会彻底破坏多普勒相位，导致血流速度计算错误。血流模式下，`demo_value` 必须保持为单一恒定值。

- IQ 解调与低通滤波的物理过程

原始 RF 信号与本地载波相乘（解调）后，信号中心被搬移到零频（基带），但同时会产生高频的负频谱镜像成分及夹杂的高频噪声。

此时配置的 FIR 低通滤波器介入。根据 `filtercoef` 的第一个值（按 $f_s/2$ 归一化）计算出实际截止频率，将带外高频成分滤除，最终输出纯净的正交 IQ 信号（I 路和 Q 路）。

- 架构差异：不同波束合成模式下的参数介入点

信号后处理的执行顺序强依赖于当前所使用的波束合成（Beamforming）模式：

聚焦波模式（Focused Wave）—— 先合成，后解调滤波

处理流：通道信号→延时叠加波束合成 (RF BF)→使用本参数进行 IQ 解调与低通滤波。

说明：传统的聚焦成像在射频域完成波束合成，合成后的单根线数据再进入后处理模块，此时执行动态解调和滤波可以有效优化最终图像。

平面波 / 扩散波模式（Plane/Diverging Wave）—— 先解调滤波，后合成

处理流：通道信号 → 使用本参数进行 IQ 解调与低通滤波→IQ 域波束合成 (IQ BF)。

说明：对 IQ 数据进行波束合成，一般要求发射频率与解调频率保持一致，以确保波束合成时的相位延迟计算准确无误。

下面是脚本中配置代码示例：

```
postpara.demo_depth = [0 0.06];
postpara.demo_value = [7e6 5e6];
postpara.filterdepth = [0.01 0.02];
postpara.filtercoef = [0.2,0.2];
postpara.filtertype = {'gauss','1.2'};
postpara.filterorder = 128;
LYConfig.PostConfig = postpara;
```

5 典型脚本编程示例

本章以聚焦波与平面波两个精简脚本为例，深入解读脚本编程规范及函数调用逻辑，旨在帮助初学者快速建立对系统配置的清晰认知。

5.1 聚焦波示例

%% 进入脚本目录，并清空当前工作区，加载所需函数、库并初始化

```
cd(fileparts(mfilename('fullpath')));
clear all
clc
close all
currentPath = pwd;
parentDir = fileparts(fileparts(currentPath));
addpath(genpath(parentDir));
init;
%% 运行过程中显示图像，当用户按下采集按钮开始采集数据
imgshowflag = 1;
savedataflag = 2;
%% 用户配置采集参数
% 探头名称
probe_name = 'L5-10';
% 显示的图像深度
imagedepth = 0.04;
% z 方向聚焦深度
focus_depth = 0.02;
% 发射线数量
scanLine = 200;
%发射波束偏转角
steers = 0;
%声速
sos = 1540;
%采样率
sysrate = 125e6;
%机器通道
channel = 128;
% 脉冲重复频率
prf = 3000;
% 采集时长
timeN = 2;
% 配置发射波形
txwave.pulse_type = 'rect';
txwave.pulse_level_type = {'A'};
txwave.pulse_cw_flag = 0;
txwave.pulse_voltage = [60];
```

```
txwave.pulse_num = 1;
txwave.pulse_duration = [0.5,0.5];
txwave.pulse_polarity = [1,-1];
txwave.pulse_frequency = 7.5e6;
% 配置降采样及低通滤波参数,
%第一级开启高通滤波,第二级为低通滤波器,降采样 2 倍,DownsampleLoc=2 在第二级滤波器
后进行降采样
DownsampleM = 2;
DownsampleLoc = 2;
FirFilter.Fir_coef1 = fir1(64, 1/(sysrate/2/1e6), 'high', hamming(65));
FirFilter.Fir_amp1 = 8;
FirFilter.Fir_coef2 = fir1(63, 15/(sysrate/2/1e6), 'low', hamming(64));
FirFilter.Fir_amp2 = 7;
% 发射孔径配置
tx_aper.max_Aper = 128;
tx_aper.min_Aper = 16;
tx_aper.fn_depth = [0 0.04];
tx_aper.fn_value = [4 4];
tx_aper.fs_win = { 'rect'};
% 接收孔径配置, 如果不显示图像可以不用配置
rev_aper.max_Aper = 128;
rev_aper.min_Aper = 48;
rev_aper.fn_depth = [0 0.04];
rev_aper.fn_value = [1.1 1];
rev_aper.js_win = hamming(256);

%%配置前端参数, 如果不配置程序使用默认参数运行
% 配置 ATGC
atgc_depth = [0, 0.005, 0.01, 0.02, 0.03, 0.04];
atgc_value = [-24, -20, -16, -8, -4, 0];
% 配置 AFE 相关参数
afe_value.lna_gain = 21;
afe_value.pga_gain = 21;
afe_value.aaf = 30;
afe_value.ipt_res = 50;

% 配置解调、低通滤波参数
postpara.demo_depth = [0 0.06];
```

```
postpara.demo_value = [txwave.pulse_frequency txwave.pulse_frequency];  
postpara.filterdepth = [0.01 0.02];  
postpara.filtercoef = [0.2,0.2];  
postpara.filtertype = 'hamming';  
postpara.filterorder = 128;
```

%% 当上述参数配置完成后，脚本会通过以下程序自动计算聚焦波所需参数

%加载探头、发射波形等参数

```
probe = Probe_para(probe_name);  
AcqConfig.Probe = probe;  
AcqConfig.Rx.timeN = timeN;  
AcqConfig.Tx.channel = channel;  
AcqConfig.Tx.fs = sysrate;  
AcqConfig.Tx.sos = sos;  
AcqConfig.Tx.focus_depth = focus_depth;  
AcqConfig.Tx.prf = prf;  
AcqConfig.Tx.steer = steers;  
AcqConfig.Tx.Pulse = txwave;  
AcqConfig.Tx.FirFilter = FirFilter;  
AcqConfig.Tx.DownsampleM = DownsampleM;  
AcqConfig.Tx.DownsampleLoc = DownsampleLoc;  
AcqConfig.Tx.ATGC.depth{1} = atgc_depth;  
AcqConfig.Tx.ATGC.value{1} = atgc_value;  
AcqConfig.Tx.AFE = afe_value;  
% 计算采样点  
[Rx,Revpt,Revdepth ]= SampleInfo(AcqConfig,imagedepth);  
AcqConfig.Rx = Rx;  
AcqConfig.Rx.Revpt = Revpt;  
AcqConfig.Rx.Revdepth = Revdepth;  
% 根据采样点、带宽确定是否重新分配 PRF  
AcqConfig = helper_CalculatePRF(AcqConfig);  
% 计算发射、接收线位置  
[emit_line,rev_line] = FsJsLineLoc(AcqConfig,scanLine);  
% 计算发射孔径大小、分配发射变迹窗  
AcqConfig.Tx.aperture = Aperature(tx_aper,  
AcqConfig.Tx.focus_depth,probe,channel);  
AcqConfig.Tx.fs_win = tx_aper.fs_win;  
% 根据发射线位置、聚焦深度等信息分配发射序列  
[tx_info,scanlist,cstartoffset ]= Tx_beamform_FocusWave(emit_line,AcqConfig);
```

```
AcqConfig.Tx.sequence = tx_info.sequence;
AcqConfig.ScanList = scanlist;
% 分配接收序列, 如果不显示图像可以不配置
rx_info = Rev_Info(rev_line,cstartoffset,AcqConfig);
AcqConfig.Rx.sequence = rx_info.sequence;
AcqConfig.Rx.aperture = rev_aper;
% 组装 LYConfig
LYConfig = buildLYConfig('AcqConfig', AcqConfig, ...
    'PostConfig', postpara, ...
    'Savedataflag', savedataflag, ...
    'Imgshowflag', imgshowflag);
% 下发、显示、采集
[AcqConfig,SucFlag] = DAQ_RealTime_FocusWave(LYConfig);
LYConfig.AcqConfig = AcqConfig;
% 保存参数
ia(LYConfig.SaveDataflag&&SucFlag)
filename_para = fullfile(LYConfig.FullPath, 'Param.mat');
save(filename_para, 'AcqConfig');
disp("Parameter file save path: " + filename_para)
end
```

5.2 平面波示例

```
%% 进入脚本目录, 并清空当前工作区, 加载所需函数、库并初始化
cd(fileparts(mfilename('fullpath')));
clear all
clc
close all

currentPath = pwd;
parentDir = fileparts(fileparts(currentPath));
addpath(genpath(parentDir));
init;

%% 运行过程中显示图像, 当用户按下采集按钮开始采集数据
imgshowflag = 1;
savedataflag = 2;
% 探头名称
probe_name = 'L5-10';
```

```
% 显示的图像深度
imagedepth = 0.04;
% 脉冲重复频率
prf = 1000; %
% 发射平面波角度
pw_tx_angle = [-8 -3 0 3 8];
%采样率
sysrate = 125e6;
%机器通道
channel = 128;
%声速
sos = 1540;
% 采集时长
timeN = 2;
%% 配置发射波形
txwave.pulse_type = 'rect';
txwave.pulse_level_type = {'A'};
txwave.pulse_cw_flag = 0;
txwave.pulse_voltage = [30];
txwave.pulse_num = 1;
txwave.pulse_duration = [0.5,0.5];
txwave.pulse_polarity = [1,-1];
txwave.pulse_frequency = 7.5e6;
% 配置降采样及低通滤波参数,
%第一级开启高通滤波, 第二级为低通滤波器, 降采样 2 倍, DownsampleLoc=2 在第二级滤波器
后进行降采样
DownsampleM = 2;
DownsampleLoc = 2;
FirFilter.Fir_coef1 = fir1(64, 1/(sysrate/2/1e6), 'high', hamming(65));
FirFilter.Fir_amp1 = 8;
FirFilter.Fir_coef2 = fir1(63, 15/(sysrate/2/1e6), 'low', hamming(64));
FirFilter.Fir_amp2 = 7;
% 接收孔径配置, 如果不显示图像可以不用配置
AcqConfig.Rx.aperture.fn_depth = [0 0.04];
AcqConfig.Rx.aperture.min_Aper = 16;
AcqConfig.Rx.aperture.max_Aper = 128;
AcqConfig.Rx.aperture.fn_value = [1.2 1];
% 配置解调、低通滤波参数
```

```

postpara.demo_value = txwave.pulse_frequency;
postpara.filtertype = 'hamming';
postpara.filterorder = 128;
postpara.filtercoef = 0.2;

%% 加载探头、发射波形等参数
probe = Probe_para(probe_name);
AcqConfig.Probe = probe;
AcqConfig.Tx.channel = channel;
AcqConfig.Tx.fs = sysrate;
AcqConfig.Tx.sos = sos;
AcqConfig.Tx.steer = pw_tx_angle;
AcqConfig.Tx.prf = prf;
AcqConfig.Tx.FirFilter = FirFilter;
AcqConfig.Tx.DownsampleM = DownsampleM;
AcqConfig.Tx.DownsampleLoc = DownsampleLoc;
AcqConfig.Tx.Pulse = txwave;
AcqConfig.Tx.fs_win = { 'rect' };
AcqConfig.Tx.aperture = channel;
% 计算采样点
[Rx, Revpt, Revdepth ]= SampleInfo(AcqConfig, imagedepth);
AcqConfig.Rx = Rx;
AcqConfig.Rx.Revpt = Revpt;
AcqConfig.Rx.Revdepth = Revdepth;
% 根据采样点、带宽确定是否重新分配 PRF
AcqConfig = helper_CalculatePRF(AcqConfig);
% 图像成像区域
sampt = round(imagedepth/(AcqConfig.Tx.sos/AcqConfig.Rx.fs/2));
scanLine = 128; % Number of scan lines
revloc.start_x = AcqConfig.Probe.element_pos.x(1);
revloc.start_y = 0;
revloc.start_z = AcqConfig.Probe.image_start ;
revloc.start_theta = 0;
revloc.step_x= (AcqConfig.Probe.element_pos.x(end)-
AcqConfig.Probe.element_pos.x(1))/(scanLine-1);
revloc.step_y = 0;
revloc.step_z = AcqConfig.Tx.sos/AcqConfig.Rx.fs/2;
revloc.step_theta = 0;
revloc.x_num = scanLine;

```

```
revloc.y_num = 0;
revloc.z_num = sampt;
revloc.theta_num = 0;
% 根据发射角度等信息分配发射序列
[tx_info,scanlist,rx_info]=
Tx_beamform_PlaneWave(pw_tx_angle,AcqConfig,revloc);
AcqConfig.Tx.sequence = tx_info.sequence;
AcqConfig.ScanList = scanlist;
% 分配接收序列, 如果不显示图像可以不配置
AcqConfig.Rx.sequence = rx_info.sequence;
% 组装 LYConfig
LYConfig = buildLYConfig('AcqConfig', AcqConfig, ...
    'PostConfig', postpara, ...
    'Savedataflag', savedataflag, ...
    'Imgshowflag', imgshowflag);

% 下发、显示、采集
[AcqConfig,SucFlag] = DAQ_RealTime_PlaneWave(LYConfig);
LYConfig.AcqConfig = AcqConfig;
% 保存参数

if(LYConfig.SaveDataflag&&SucFlag)
    filename_para = fullfile(LYConfig.FullPath, 'Param.mat');
    save(filename_para, 'AcqConfig');
    disp("Parameter file save path: " + filename_para)
end
```

附录 1. Pinmap（引脚映射）说明

	1	2	3	4	5	6	7	8	9	10	
A	GNDSENS	AGND	ID_PDAT	ID_PCLK	ID_PSEL	ID_PWEN	ID_PFWR	AGND	AGND	AGND	A
B	AGND		PRE_EN		KEY1	KEY2	LED1	LED2		AGND	B
C	AGND									AGND	C
D	AGND	D0	D1	D2	D3	D4	D5	D6	D7	AGND	D
E	AGND	VPP	VNN	VDD		MLE	MCLK		AGND	AGND	E
F	AGND	12V	12V							AGND	F
G	AGND									AGND	G
H	AGND									AGND	H
J	AGND									AGND	J
K	AGND	E93	E94	E95	E96	E97	E98	E99	E100	AGND	K
L	AGND	E92	E91	E90	E89	E104	E103	E102	E101	AGND	L
M	AGND	E85	E86	E87	E88	E105	E106	E107	E108	AGND	M
N	AGND	E84	E83	E82	E81	E112	E111	E110	E109	AGND	N
P	AGND	E77	E78	E79	E80	E113	E114	E115	E116	AGND	P
R	AGND	E76	E75	E74	E73	E120	E119	E118	E117	AGND	R
S	AGND	E69	E70	E71	E72	E121	E122	E123	E124	AGND	S
T	AGND	E68	E67	E66	E65	E128	E127	E126	E125	AGND	T
U	AGND	E4	E3	E2	E1	E64	E63	E62	E61	AGND	U
V	AGND	E5	E6	E7	E8	E57	E58	E59	E60	AGND	V
W	AGND	E12	E11	E10	E9	E56	E55	E54	E53	AGND	W
X	AGND	E13	E14	E15	E16	E49	E50	E51	E52	AGND	X
Y	AGND	E20	E19	E18	E17	E48	E47	E46	E45	AGND	Y
Z	AGND	E21	E22	E23	E24	E41	E42	E43	E44	AGND	Z
AA	AGND	E28	E27	E26	E25	E40	E39	E38	E37	AGND	AA
BB	AGND	E29	E30	E31	E32	E33	E34	E35	E36	AGND	BB
CC	AGND	AGND		RT1	RT2			RTGND	AGND	AGND	CC
	1	2	3	4	5	6	7	8	9	10	

图 21 260 接口 128 通道的 Pinmap

Pin	A	B	C	D	E	F	G	H	J	K	L	M	Pin
1	GND	GND	GND	GND	GND	GND	GND	GND	GND	GND	GND	RSVD	1
2	ES 160	GND	ES 223	GND	ES 256	ES 32	ES 63	ES 95	ES 96	ES 128	RSVD	GND SNS	2
3	ES 159	ES 191	ES 192	ES 224	ES 255	ES 31	GND	ES 64	GND	ES 127	RSVD	RSVD	3
4	ES 158	GND	ES 221	GND	ES 254	ES 30	ES 61	ES 93	ES 94	ES 126	RSVD	RSVD	4
5	ES 157	ES 189	ES 190	ES 222	ES 253	ES 29	GND	ES 62	GND	ES 125	RSVD	RSVD	5
6	ES 156	GND	ES 219	GND	ES 252	ES 28	ES 59	ES 91	ES 92	ES 124	RSVD	RSVD	6
7	ES 155	ES 187	ES 188	ES 220	ES 251	ES 27	GND	ES 60	GND	ES 123	RSVD	RSVD	7
8	ES 154	GND	ES 217	GND	ES 250	ES 26	ES 57	ES 89	ES 90	ES 122	RSVD	RSVD	8
9	ES 153	ES 185	ES 186	ES 218	ES 249	ES 25	GND	ES 58	GND	ES 121	RSVD	RSVD	9
10	ES 152	GND	ES 215	GND	ES 248	ES 24	ES 55	ES 87	ES 88	ES 120	RSVD	RSVD	10
11	ES 151	ES 183	ES 184	ES 216	ES 247	ES 23	GND	ES 56	GND	ES 119	RSVD	RSVD	11
12	ES 150	GND	ES 213	GND	ES 246	ES 22	ES 53	ES 85	ES 86	ES 118	RSVD	RSVD	12
13	ES 149	ES 181	ES 182	ES 214	ES 245	ES 21	GND	ES 54	GND	ES 117	RSVD	RSVD	13
14	ES 148	GND	ES 211	GND	ES 244	ES 20	ES 51	ES 83	ES 84	ES 116	RSVD	KEY2	14
15	ES 147	ES 179	ES 180	ES 212	ES 243	ES 19	GND	ES 52	GND	ES 115	RSVD	KEY1	15
16	ES 146	GND	ES 209	GND	ES 242	ES 18	ES 49	ES 81	ES 82	ES 114	RSVD	GND	16
17	ES 145	ES 177	ES 178	ES 210	ES 241	ES 17	GND	ES 50	GND	ES 113	RSVD	GND	17
Pin	A	B	C	D	E	F	G	H	J	K	L	M	Pin
18	ES 144	GND	ES 207	GND	ES 240	ES 16	ES 47	ES 79	ES 80	ES 112	RSVD	RSVD	18
19	ES 143	ES 175	ES 176	ES 208	ES 239	ES 15	GND	ES 48	GND	ES 111	RSVD	RSVD	19
20	ES 142	GND	ES 205	GND	ES 238	ES 14	ES 45	ES 77	ES 78	ES 110	SDA	RSVD	20
21	ES 141	ES 173	ES 174	ES 206	ES 237	ES 13	GND	ES 46	GND	ES 109	SCK	RSVD	21
22	ES 140	GND	ES 203	GND	ES 236	ES 12	ES 43	ES 75	ES 76	ES 108	GND	RSVD	22
23	ES 139	ES 171	ES 172	ES 204	ES 235	ES 11	GND	ES 44	GND	ES 107	GND	BIAS_5V	23
24	ES 138	GND	ES 201	GND	ES 234	ES 10	ES 41	ES 73	ES 74	ES 106	RSVD	BIAS_5V	24
25	ES 137	ES 169	ES 170	ES 202	ES 233	ES 9	GND	ES 42	GND	ES 105	RSVD	RSVD	25
26	ES 136	GND	ES 199	GND	ES 232	ES 8	ES 39	ES 71	ES 72	ES 104	RSVD	RSVD	26
27	ES 135	ES 167	ES 168	ES 200	ES 231	ES 7	GND	ES 40	GND	ES 103	RSVD	RSVD	27
28	ES 134	GND	ES 197	GND	ES 230	ES 6	ES 37	ES 69	ES 70	ES 102	RSVD	RSVD	28
29	ES 133	ES 165	ES 166	ES 198	ES 229	ES 5	GND	ES 38	GND	ES 101	RSVD	RSVD	29
30	ES 132	GND	ES 195	GND	ES 228	ES 4	ES 35	ES 67	ES 68	ES 100	A1	RSVD	30
31	ES 131	ES 163	ES 164	ES 196	ES 227	ES 3	GND	ES 36	GND	ES 99	A0	RSVD	31
32	ES 130	GND	ES 193	GND	ES 226	ES 2	ES 33	ES 65	ES 66	ES 98	RSVD	RSVD	32
33	ES 129	ES 161	ES 162	ES 194	ES 225	ES 1	GND	ES 34	GND	ES 97	RSVD	RSVD	33
34	GND	GND	GND	GND	GND	GND	GND	GND	GND	GND	GND	RSVD	34
Pin	A	B	C	D	E	F	G	H	J	K	L	M	Pin

图 22 408 接口 256 通道的 Pinmap

Pinmap（引脚映射表）是连接探头物理阵元与系统硬件通道的“地图”。它规定了系统上的每一个通道，具体对应到探头接口的哪一个物理引脚（Pin）上。

以附录 1 中图 1 为例，其中 E_i 为第 i 个通道，E1-E128 对应第 1 通道—第 128 通道。

要实现“第 i 个阵元”由“第 i 个系统通道”控制，硬件上的物理连接必须遵循以下路径：

探头第 i 阵元→经过探头的内部排线→连接到接口的 E_i 引脚→探头插入转接口→ E_i 引脚→经过系统的内部排线→连接到系统的第 i 通道。

因此，Pinmap 是提供给探头厂家的加工依据，要求厂家按照该映射关系进行内部排线，确保各阵元与引脚准确对应。

版权与法律声明

版权所有

本手册及相关软件开发包（SDK）/ 应用程序接口（API）的全部内容（包括但不限于文字、图表、代码示例、架构设计），版权均归**浙江利影医疗科技有限公司**所有。未经本公司事先书面许可，任何单位或个人不得以任何形式或手段翻印、复制、传播、修改或翻译本手册的任何内容。

使用限制与许可

本手册及配套 SDK/API 仅限用于配合 Ultra-S256 系列硬件设备的二次开发与系统集成。使用者不得对本硬件及软件系统进行反向工程（Reverse Engineering）、反编译（Decompile）、反汇编（Disassemble），或尝试获取系统底层源代码及未公开的硬件控制架构。

编程免责与责任限制

本手册中提供的代码示例、脚本指令及 API 参数仅供开发参考与集成示范。虽然本公司已尽力确保手册内容的准确性，但不保证 SDK/API 能完全满足特定用户的特殊业务需求，亦不保证代码运行毫无中断或无错。

对于因用户自行编写代码、执行自定义脚本、超范围修改参数或操作不当所导致的设备损坏、数据丢失、性能下降或任何间接损失，本公司概不承担任何法律与赔偿责任。

(注：本公司保留根据技术演进随时更新本手册及 API 接口的权利，恕不另行通知。)