

CH32 二维运行结构概览

版本：V1.1

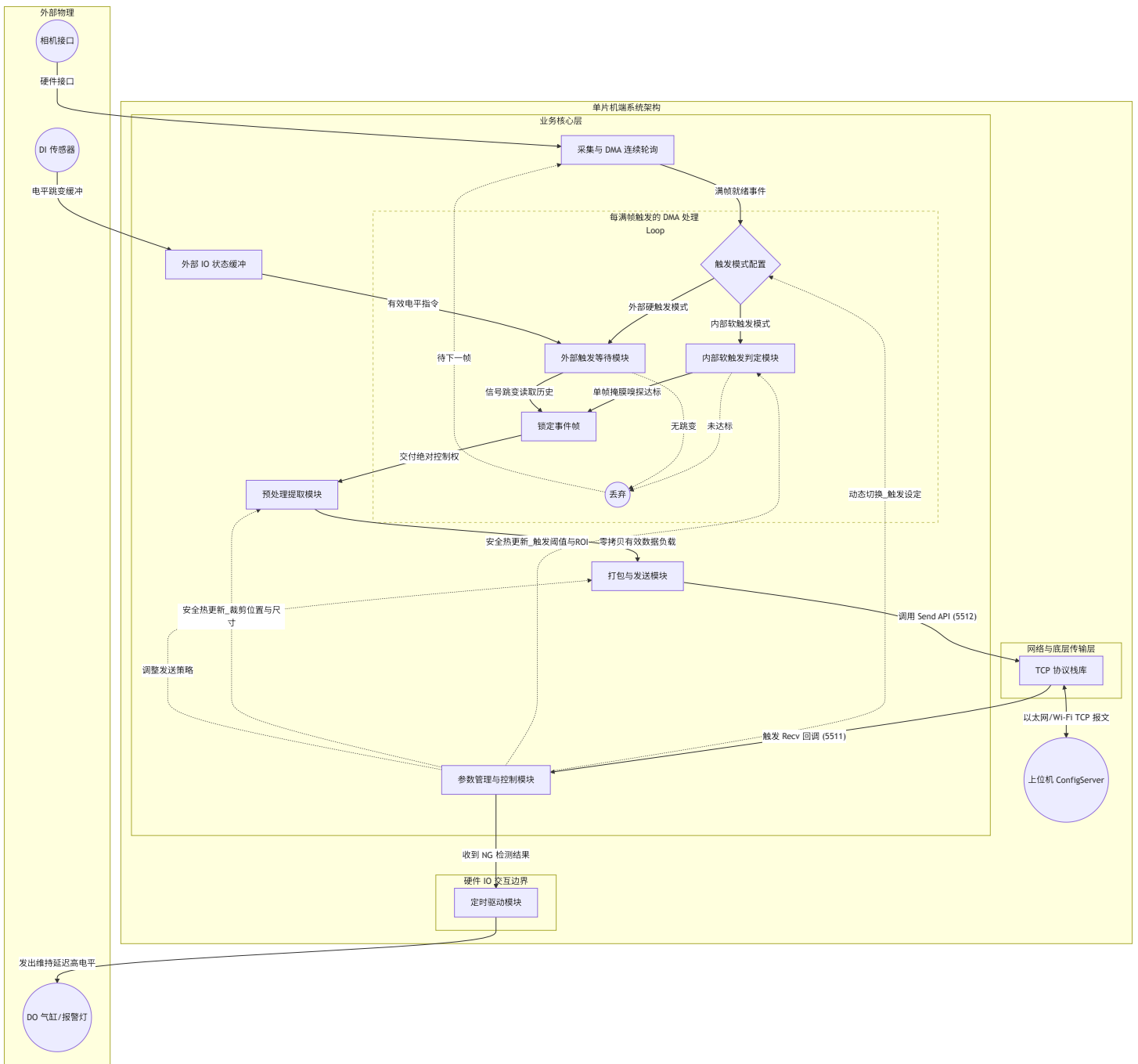
更新说明：更新系统架构图

1. 概述

本文档旨在描述运行于 CH32 单片机上的核心业务逻辑和软件架构。系统屏蔽了底层的硬件驱动细节，主要聚焦于单片机端的**业务流转**和**数据处理**。整个系统由采集、处理、发送三大核心模块组成，并辅以 TCP 协议栈库和动态参数配置功能，以确保系统在高性能要求下稳定高效地运行。

2. 系统核心架构

系统整体基于数据流驱动，涵盖了从图像帧获取到最终网络传输的完整生命周期。



3. 核心功能模块详细设计

3.1 采集模块

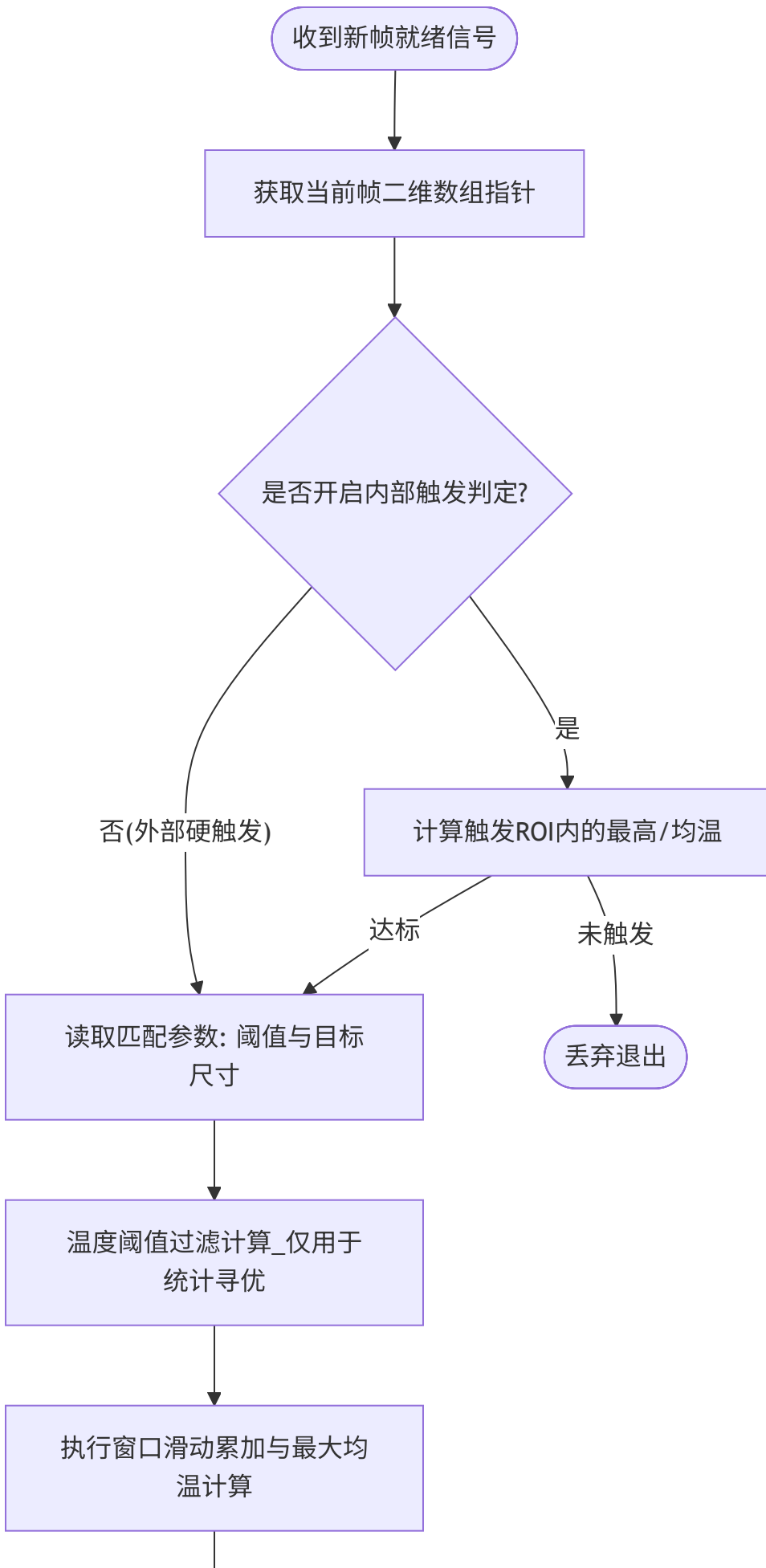
- **定位：** 底层硬件和信号抽象。
- **职责：** 负责与相机进行物理通信。对单片机其他业务代码而言，它是一个完全解耦的黑盒。
- **业务对接机制：** 采集模块内部自行处理相机通信、触发延时、DMA采集等。每完成一帧的采集，模块会将数据转化为统一的**二维整数矩阵 (Integer Matrix)**，存入单片机内存的环形缓冲区，并通过软中断、回调函数或信号量通知预处理模块接管数据。

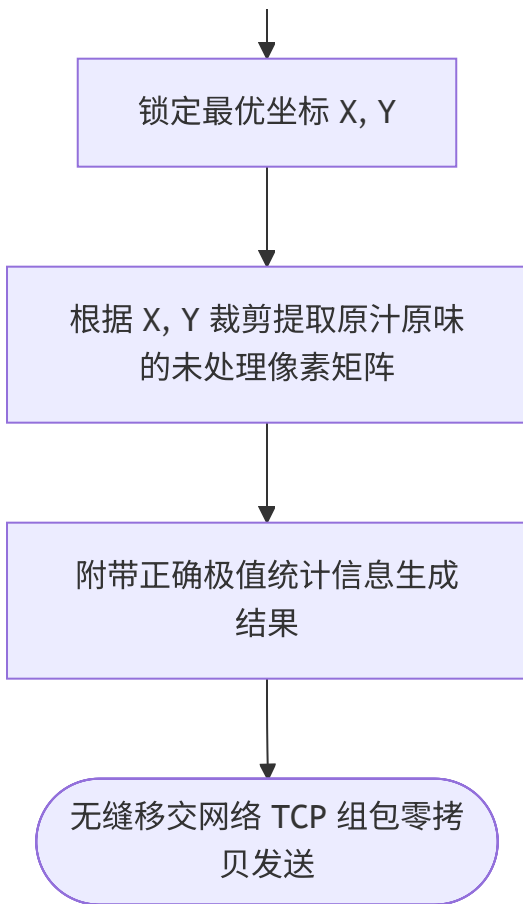
3.2 内部软触发判定模块（核心控制块）

- **定位：**轮询模式下的事件嗅探器。
- **职责：**在内部触发（TriggerMode = 0）模式下，高频处理由底层 DMA 源源不断抛出的完整原始测温满帧，判断并锁定有效目标进入画面的“瞬间”。
- **工作机制：**
 - 高速嗅探：**每收到一帧满帧，立即调用 Preprocess_CheckInternalTrigger2D API 快速浏览预设的触发感兴趣区域 (Trigger ROI)。这避免了对无需关注的无目标区域进行毫无意义的算力浪费。
 - 掩膜嗅探：**运用设定的温度阈值将低温背景瞬间剔除，并迅速计算出该小格 ROI 内部的最高温度或平均温度。
 - 定格：**当发现温度满足判定条件时，立刻向主控系统反馈命中信号（返回 1），主控随即接管该帧缓冲区的绝对控制权（防止它被后续 DMA 盲目轮询覆盖），将其确认为事件起点帧（第 0 帧），推入下一级的数据裁剪与提取阶段，并开启定时连拍。

3.3 预处理与提取模块（核心业务块）

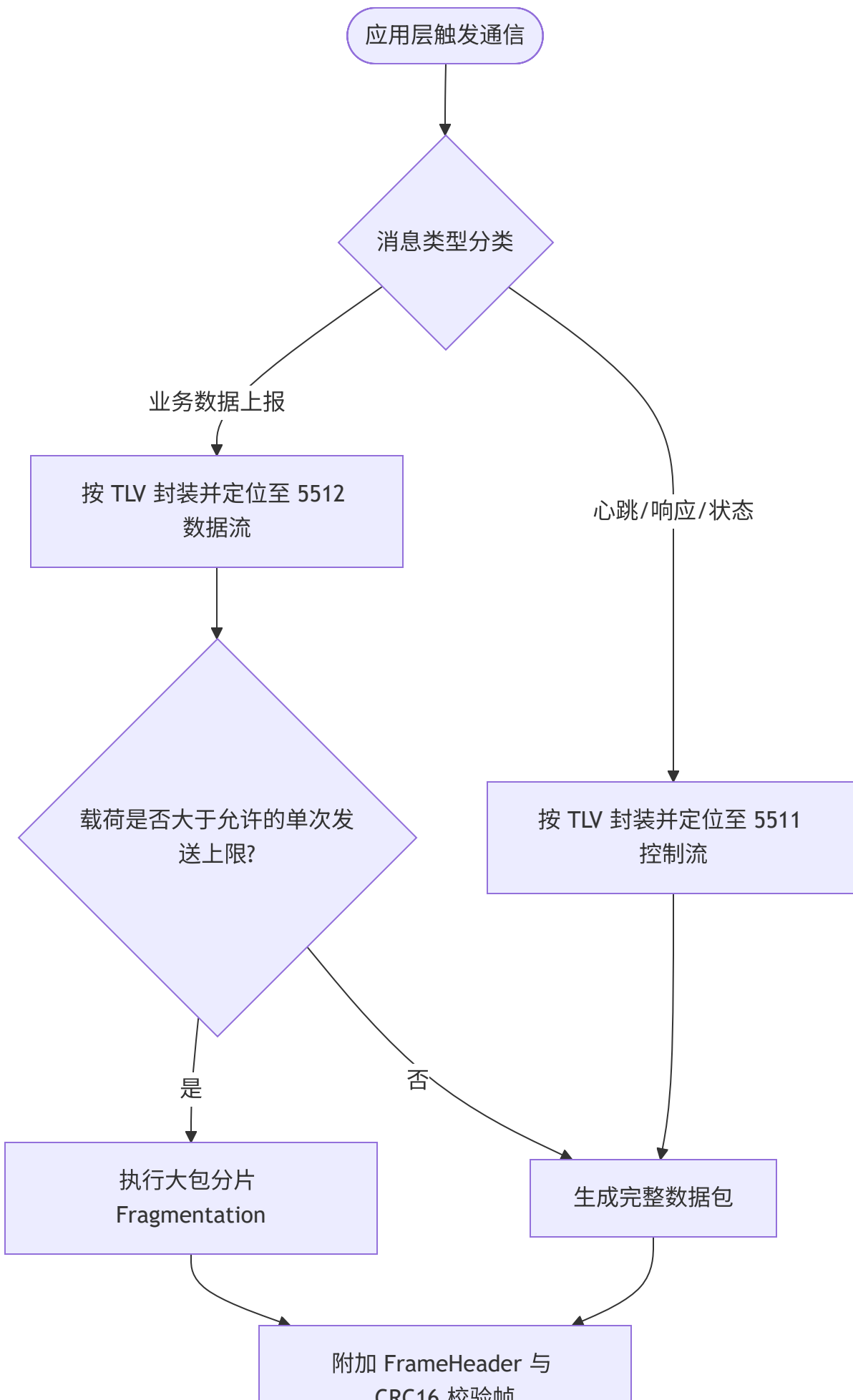
- **定位：**数据加工、过滤清洗与裁剪提炼。
- **职责：**接收整数矩阵，并根据当前的算法参数（如过滤阈值 TriggerTemperatureThreshold、目标尺寸 TargetWidth * TargetHeight）进行滑动滤波和坐标锁定。
- **核心处理逻辑（温度过滤与最大均温滑动计算）：**
 - 温度过滤预处理：**识别出低于阈值的像素，在计算中视作默认低温值（如 90，即 9.0°C），保留高于等于阈值的像素原始温度参与计算。
 - 滑动窗口计算：**通过列累加和向右滑动窗口（加右侧新列减左侧旧列），快速计算二维矩阵内连续 TargetWidth * TargetHeight 区域的总和（即平均温度）。
 - 位置锁定：**在预设或受控范围（如 ROI）内记录总和最大值的起始坐标 (x, y)，这即是触发帧的最佳裁剪位置。
 - 零拷贝极速搬运：**基于锁定好的坐标 (x, y)，从原始环形缓冲区中原样提取出这块 TargetWidth * TargetHeight 的**未被修改过的真实测温有效像素**。直接填入外部提供的已经偏移好的网络打包内存中。
- **执行流程：**

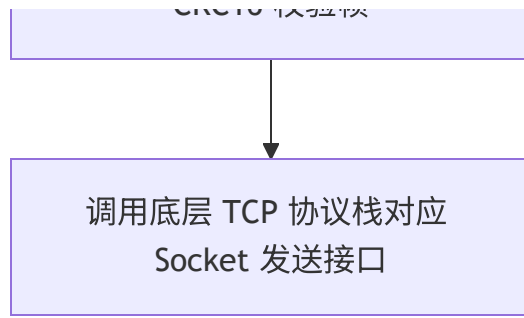




3.4 TCP 打包与发送模块（核心业务块）

- **定位：**基于 TCP Raw Stream 2.0 协议的应用层通信封装、链路管理与调度引擎。
- **主要功能总结：**
 - 双流管理与主动连接：**采用“采集端主动连接，上位机被动监听”模式，解决 IP 不固定的问题。系统维护两个独立的 TCP 长连接以实现控制与数据解耦：
 - **控制流 (Port 5511)：**用于配置下发、指令控制与心跳维护。
 - **数据流 (Port 5512)：**专属用于高频温度数据（如 TemperatureFrame_t）的实时上报。
 - 身份握手与动态绑定：**连接建立后发送首包进行握手（Handshake），携带硬件 UUID。支持服务端对逻辑设备 ID (DevID) 的冲突检测和动态重新分配，保证多设备并发下的正确识别。
 - 数据封装与打包：**按照 2.0 小端序规范，将业务数据封装为 TLV（Type-Length-Value）结构。自动在其外侧包裹核心帧头（包含 Magic 标识(0x55AA)、序列号、时间戳、载荷属性等）以及尾部的 CRC16 校验和。
 - 可靠性传输机制：**
 - **重传与确认：**针对控制指令和握手实施 ACK 确认与重传。
 - **大包分片 (Fragmentation)：**当业务载荷超过 MTU/分片最大限制时，自动进行序列分片传输，以确保网络利用率。
 - 链路诊断与恢复：**维持周期心跳机制（Keep-Alive）具备超时判定功能；在遭遇断网或对端重置后，可执行指数退避式的高容忍度重连，并自动重启握手绑定流程。
- **执行流程概览：**





3.5 参数更改模块（控制业务块）

- **定位：**系统的“神经中枢”与配置通道。
- **职责：**解析基于 TCP 连接传来的上位机控制指令，动态修改单片机的工作行为及参数（如：预处理阈值，网络打包频率等），而无需复位单片机。
- **执行流程：**
 - i. **解析：**TCP 栈触发接收回调，提取载荷中的指令码及配置数据。
 - ii. **校验：**检查数据的合法性。
 - iii. **应用：**使用**原子操作**或**影子寄存器（Shadow Buffer）机制**更新全局参数结构体，确保正在进行图像处理的流水线不会因为参数中途突变而导致指针越界或计算崩溃。
 - iv. **回执：**向客户端回传 ACK 包，告知参数修改生效。

3.6 硬件 IO 与触发反馈模块（交互边界）

- **定位：**处理与外部硬件的物理数字电平交互（DI/DO），连接“触发动作”与“剔除执行”。
- **职责与工作机制：**负责实时响应外部脉冲输入或程序内部事件，并能根据网络判定结果执行动作以控制外部设备。

① 触发采集机制（影响 workflow 起点）

采集流水线的启动方式严格受触发模式配置参数的影响：

1. 外部触发 (External Trigger):

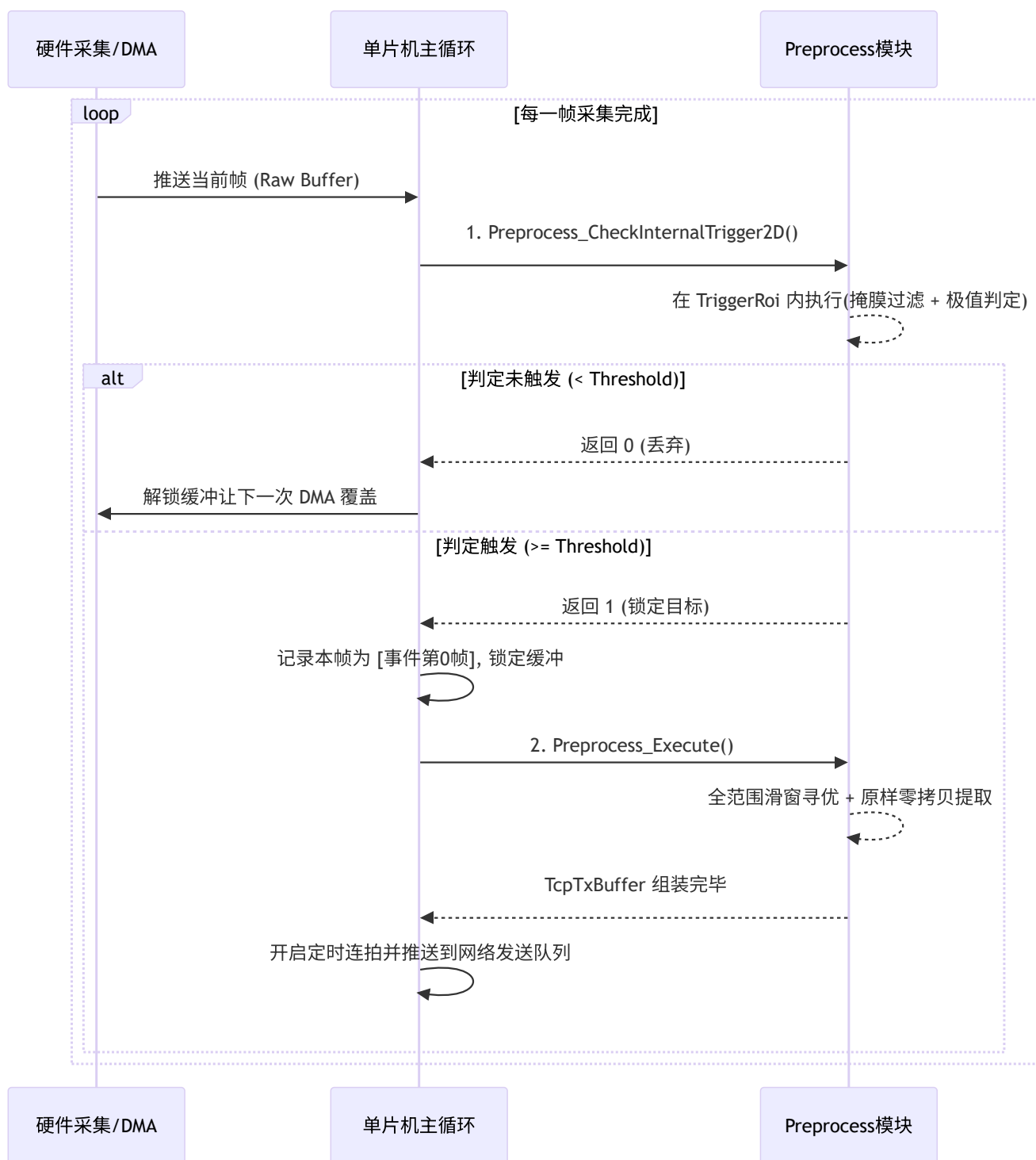
- **行为逻辑：**系统实时监听预设的 **DI 触发引脚**（例如：接收光电传感器的输入）。当检测到有效电平变化，并在通过设定的消抖滤波间隔（`TriggerDebounceIntervalMs`）确认信号稳定后，视为正式触发。
- **影响：**系统在经历特定的触发延时后，命令采集模块连续捕捉设定的张数（`TriggerBurstCount`）。只有被这批动作捕捉到的二维矩阵数据才会进入预处理与 TCP 发送队列。非触发状态下，系统处于待机且不产生多余计算和网络流量。

2. 内部触发 (Internal Trigger / 轮询阈值触发):

- **行为逻辑：**采集模块通过底层 DMA 持续不间断地循环向单片机缓冲内存区抛出测温画面。主程序每收到一帧，都会调用 `Preprocess_CheckInternalTrigger2D` 进行轻量级的 ROI 温度计算嗅探。

- **算法双轨制设计：(核心重点)** 无论是触发嗅探还是后续的截取寻优，系统都会使用配置的恒定底噪温度（如 9.0°C）掩盖低温像素。但**最终打包进网络的数据与统计报告（最高/最低/平均温）必须是原汁原味、从未被过滤过的测温像素**。这种分离设计既保证了单片机极速抓取目标的抗干扰能力，又避免了上位机 AI 模型因收到修改过的纯色背景图而发生误判。
- **影响：**一旦目标进入画面，触发算法判定达标，系统会立刻“锁定”该帧所在缓冲区（防止被 DMA 再次覆盖），确认其为事件起点帧（第0帧），并在提取发出后稳定连拍指定的定格张数。

内部触发 (基于轮询) 行为动作图:



② 自动剔除与报警输出机制 (驱动机构)

本架构支持全链路防错的闭环响应机制，将上位机的智能分析结果迅速转译为实际动作：

- **接收判定 (触发下发)**：TCP 接收任务 (控制流) 收到来自主机的特定结果回调封包 (如 `DetectionResult_t` 宣告当前帧检测失败/缺陷，即 `Result == 1 / NG`)。
- **执行输出 (DO 直接驱动)**：
 - i. 识别到 NG，系统内部回调会立即响应，拉高预先配置的 **DO 输出引脚** (对应生产流水线的废料剔除气缸、拨片或是声光报警装置)。
 - ii. 系统内部开启定时器，维持该高电平动作状态 `NGioDelay` 毫秒时长，这确保了较慢的机械分拨机构能吃满行程去推落次品。
 - iii. `NGioDelay` 时间期满后，定时中断会自动将对应 DO 信号电平拉低复位，干净利落地收回气缸，复归常态以应对下一次测试。

3.7 TCP 协议栈功能库 (底层支持)

- **定位**：可靠的流式网络传输支持 (如 LwIP 移植)。
- **职责**：
 - **Socket 抽象**：提供标准化、极简的创建、监听、连接、发送、接收 API。
 - **状态机维护**：自动处理三次握手、四次挥手、重传及断线侦测操作。
 - **缓冲管理**：管理底层的发送缓冲和接收窗口 (TCP Window)。

4. 性能保证与优化策略

为确保“采集 -> 处理 -> 发送”这一重负载流水线在 CH32 上具备高帧率、低延迟的性能，需采用以下策略保障：

1. 零拷贝 (Zero-Copy) 内存流转：

在整个数据链路中，严禁在不同业务逻辑之间进行大块内存的 `memcpy` 复制。应当采用传递内存指针的方式，让不同的模块轮流获取同一块内存的读写权限。

2. Ping-Pong 双缓冲设计 (Double Buffering)：

为“矩阵输出”和“预处理”分配 A/B 两块缓冲区。当采集模块(黑盒)向 Buffer A 写入第 N 帧时，预处理模块正在读取 Buffer B 处理第 N-1 帧。两者物理隔离，实现 CPU 和外设间的全并发，无缝衔接。

3. 事件驱动与 RTOS 并发：

抛弃低效的死循环轮询模式。围绕 RTOS 任务构建模型：采集结束触发信号量 -> 唤醒高优先级处理任务 -> 唤醒发送任务。其余时间 CPU 处于休眠或处理低速网络事件。

4. 发包防阻塞与 TCP 合并传输：

将不重要且碎片的包合并发送；针对大数据块发送，采用异步队列机制或者结合 DMA 送到网络

MAC 接口，防止 CPU 长时间处于等待 TCP ACK 的阻塞状态。