

视频监控系統程序設計說明書

1. 前言

编写目的

本文主要說明视频监控系統的詳盡設計與實現。本文詳盡介紹了系統中各個模塊的設計與實現細節以及要點技術的應用。此中，對每一個模塊會有相應的文字說明以及有關設計流程圖給出。

項目背景

本項目是對本實驗課程中所介紹的課程的綜合應用。

本項目主要針對 `unix/linux` 系統編程和鑑於 `C++` 語言的 `GUI` 程序設計 - `QT` 進行針對性訓練。旨在增強學生對 `unix/linux` 系統編程的理解和應用，提高學生的實踐能力、編程能力以及剖析問題和解決問題的能力。並且經過對本系統中客戶端的设计和实现，让学生领会和熟习面向对象编程的思想和一般方法。

定义

本系統 = 视频监控系統

本文 = 视频监控系統設計說明書

2 整體設計

需求概括

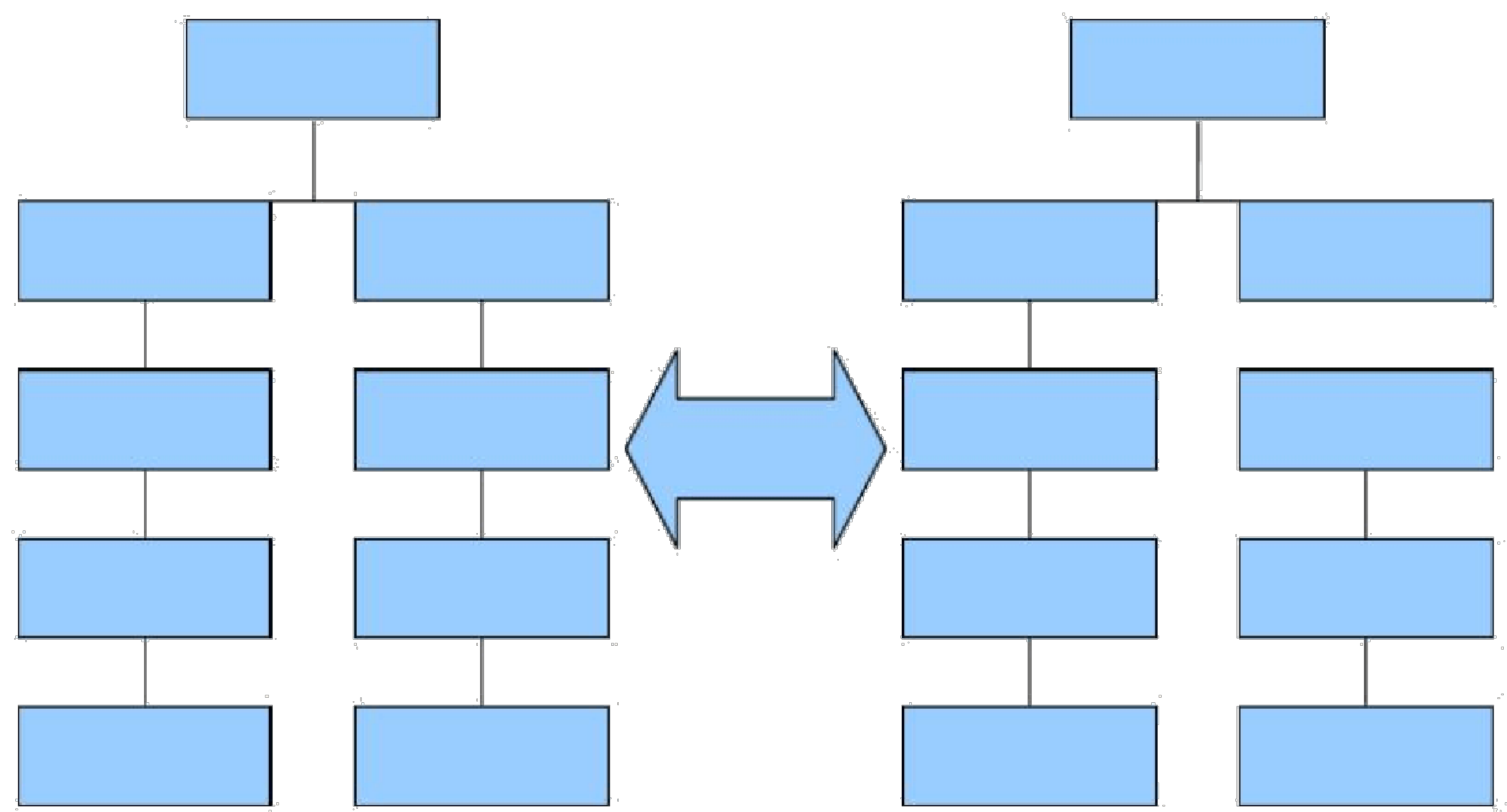
本系統是鑑於 `C/S` 架構的網絡视频监控系統，主要由負責圖形收集和傳輸的 `Server` 端和負責監控（顯示圖像）的 `Client` 端組成。

本系統 `Server` 端是鑑於 `Linux` 系統的網絡服務類軟件，主要由數據采集和網絡服務兩部分組成。數據收集部分利用 `v4l2`（`video for linux II`）對網絡攝像頭進行控制和接見並且獲取由攝像頭捕捉到的視頻數據，而後再由網絡服務部分將獲取到的視頻數據進行封裝成 `http` 幀並利用 `http` 協議發送到客

户端。

因为本系统 **Server** 端的数据发送是鉴于 **http** 协议的，因此本系统客户端没有平台限制，只需支持以上的平台就能够接见本系统的服务。并且，本系统服务能够直接经过 **web** 阅读器接见（对 **firefox** 阅读器支持最正确）。为了提升传输效率我们还需要进行客户端软件的开发，因为对客户端没有平台要求，所以在 **Windows**、**Linux**、**Android**、等平台都能够实现。本文主将以 **linux** 为例用 **Qt** 实现一个简略的客户端软件。

软件构造



图视频监控系统构造图

3. 程序描绘

Server:: 数据收集

功能

配置设施并从设施中获取图像数据。

依据设施信息利用 v4l2 (video for linux II) 经过 ioctl 对 USB 摄像头进行配置，翻开设施抓取功能开始收集并转存数据，收集完成此后封闭设施功能，最后封闭设施。

技术纲要

本模块波及到大批的对于设施的 I/O 操作，主要经过 video for linux II(v4l2) 框架对设施进行设置和控制。此中包含一些特别的构造体属性的赋值以及一些特别控制标志的实质意义需要详尽查阅 v4l2 供给的 API。

程序逻辑

本节主要以流程图的方式介绍 Server 端数据收集模块的主要程序逻辑设计。

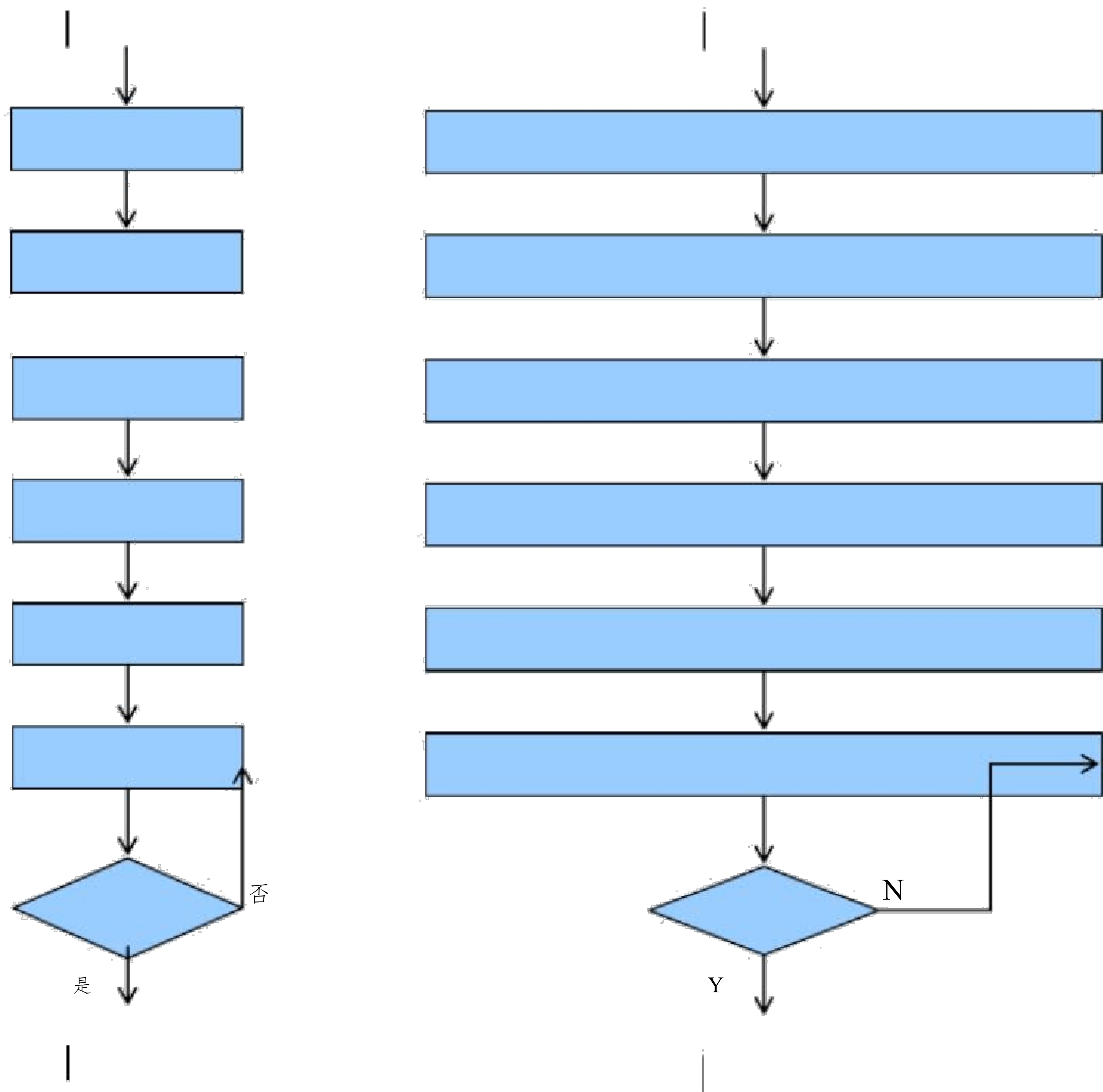


图 Server 端数据收集流程图

图主要描绘了服务端数据收集部分的主要流程和编程步骤。下边将对每一个步骤进行详尽的介绍：

1. 翻开设施：

翻开视频设施（这里指我们的 USB 摄像头）。利用 linux 系统供给的接口函数 `open` 来对设施进行翻开的操作。图对这一过程进行了详尽的描绘。

2. 查问设施信息（或许叫获取设施信息）：

查问设施驱动有关信息以及和视频收集有关的参数，依据获取到的信息初步判断设施种类以及将要办理的数据种类。利用 `ioctl` 函数以及 `v4l2` 供给的 `v4l2_capability` 和 `v4l2_format` 两个构造体来保留获取到的设施驱动信息和帧信息。详尽流程见图。 `VIDIOC_G_FMT` 等属性的含义需查问 `v4l2` 供给的 API。

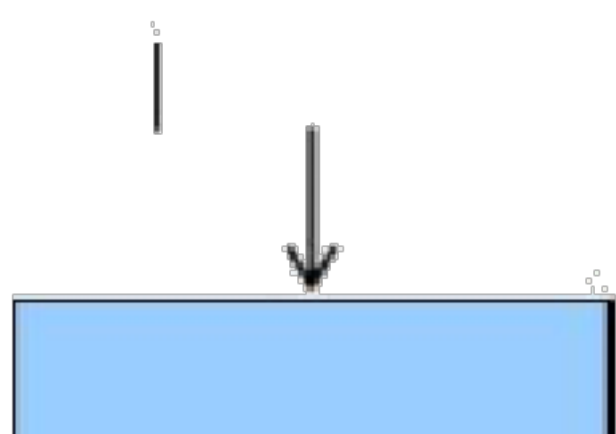


图 翻开设施子图

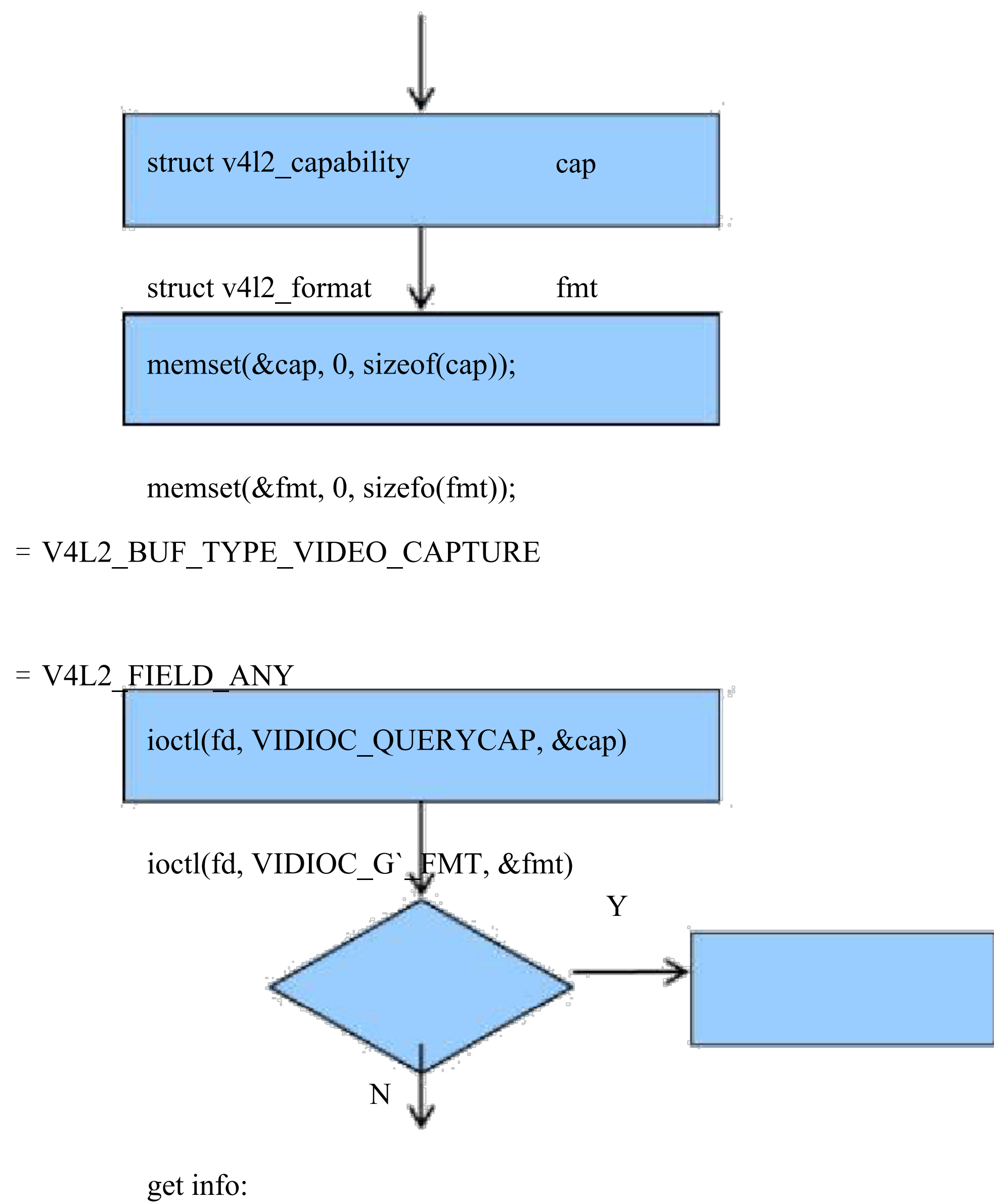


图 获取设施信息子图

3. 设置收集方式

设置视频设施（这里指 USB 摄像头）的收集方式，比如收集图像的分辨率，帧的收集速率（既每秒收集帧的数目）等选项的设置。用到 v4l2 中的 v4l2_format 和 v4l2_streamparm 两个结构体以及 VIDIOC_S_PARM 和 VIDIOC_SFMT 两个标志，有关介绍请查阅 v4l2 框架的 API 信息。实现步骤详见图。

```
// 本模块用到的宏定义

struct v4l2_format      fmt      2

#define WIDTH      320

struct v4l2_streamparm      parm

#define HEIGHT      240

memset(&parm, 0, sizeof(parm));

#define FPS      5

memset(&fmt, 0, sizeof(fmt));

= V4L2_BUF_TYPE_VIDEO_CAPTURE

= V4L2_FIELD_ANY
```


图 设置收集方式子图

4.

向驱动申请几块缓冲区（此处缓冲区是由内核管理的，因此需要申请），用来寄存设施收集到的数据。用到 v4l2 中的 v4l2_requestbuffer 构造体和 VIDIOC_REQBUFS 标志来向设施驱动申请缓冲区。实现步骤详见图。

```
// 本模块用到的宏定义
struct v4l2_requestbuffer rb 3

// 表示申请缓冲区的数目

#define NB_BUFFER 5

memset(&rb, 0, sizeof(rb));

= NB_BUFFER

= V4L2_BUF_TYPE_VIDEO_CAPTURE

= V4L2_MEMORY_MMAP

ioctl(fd, VIDIOC_REQBUFS, &rb)

N

Y
```


申请缓存子图

5. 设置缓存属性

对申请到的缓存进行设置，包含设置缓存的接见方式（使用内存映照接见方式），对缓存进行映照，成立缓存行列。

```
struct v4l2_buffer      buf

unsigned char ** mem_list
mem_list = (uchar **)calloc(NB_BUFFER, sizeof(uchar));

int i = 0;

N

Y

= i
```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/535321203001011304>