

## 开发的硬件环境

---

1. CPU: Intel(R) Core(TM) i5-8260U CPU @ 1.60GHz
2. RAM: 24G
3. SSD: 1TB

## 运行的硬件环境

---

1. CPU: RK3568 Quad-Core ARM Cortex-A55
2. RAM: 2G
3. ROM: 16GB

## 开发该软件的操作系统

---

Ubuntu 16.04

## 软件开发环境/开发工具

---

1. 编译器: Buildroot-arm-linux-gnueabi-hf-gcc、Buildroot-arm-linux-gnueabi-hf-g++
2. 构建工具: CMake, Make

## 该软件运行的平台/操作系统

---

嵌入式Linux

## 软件运行支撑环境/支持软件

---

1. 支持C++11标准的任何编译器

## 编程语言

---

C、C++

## 源程序量

---

230行

## 开发目的

---

1. 实现红外图像处理与推流一体化:
  - 设计一个软件平台，能够在Linux系统上处理红外图像并将其推流至Android层。
  - 目的是为用户提供一个统一的解决方案，方便处理和查看红外图像，减少用户使用不同软件的复杂性。
2. 提供多摄像头并发输入功能:
  - 支持连接多个摄像头，包括红外和可见光摄像头，以提供更全面的图像信息。
  - 目的在于拓展应用场景，使得用户可以同时获取多个角度的图像数据，满足不同需求。
3. 实现高效数据处理:
  - 通过V4L2接口对摄像头数据进行处理，包括图像增强、滤波、温度计算等。

- 目的是保证数据处理过程的高效性和准确性，提高红外图像的质量和可用性。
4. 支持RTSP协议推流：
- 利用RTSP协议将处理后的红外图像流进行推流，实现远程实时查看处理后的图像。
  - 目的在于实现远程监控和控制，增强用户对红外图像的实时性需求。
5. 提供Android端图像接收与显示功能：
- 在Android层，提供专门的应用接收和显示RTSP推流的红外图像。
  - 目的是为用户提供友好的操作界面，方便实时观察红外图像，增强用户体验。
6. 设计灵活适配多个应用场景：
- 平台设计灵活，适用于工业、电力、医疗等领域，并可以应用于其他需要红外图像处理的领域。
  - 目的在于满足不同用户的需求，提供广泛的应用前景。
7. 保证实时性与稳定性：
- 通过使用RTSP协议和系统优化，保证数据传输的实时性和稳定性。
  - 目的是确保用户在不同场景下获得高质量、低延迟的红外图像展示，提升用户满意度。
8. 设计用户友好的界面：
- Android端应用注重用户体验，提供直观、简洁的界面，使用户能够轻松操作、查看并控制红外图像处理平台。
  - 目的在于提高用户接受度和使用效率，使用户能够更方便地利用该软件进行红外图像处理。

## 软件技术特点

---

1. 多摄像头数据并发输入：支持连接多个摄像头，实现数据并发输入，提供更全面的图像信息，拓展了应用场景。
2. 高效数据处理：通过V4L2接口实现对摄像头数据的处理，包括图像增强、滤波、温度计算等，保证了数据处理过程的高效性与准确性。
3. RTSP协议推流：利用RTSP协议将处理后的红外图像流进行推流，实现远程实时查看处理后的图像，保证了数据传输的实时性。
4. Android端图像接收与显示：提供专门的应用接收RTSP推流，并以类似相机的方式展现图像，提高了用户体验。
5. 灵活的应用适配性：设计灵活，适用于多个应用场景，满足用户多样化的需求，具有广泛的应用前景。
6. 实时性与稳定性：通过RTSP协议的使用和系统优化，保证了数据传输的实时性和稳定性，提供高质量、低延迟的红外图像展示。
7. 用户友好的界面设计：Android端应用设计注重用户体验，提供直观、简洁的界面，方便用户操作、查看并控制红外图像处理平台。