**基于树莓派的多功能相机**

产品文档

**版本： V1.0**

# 1 项目功能与环境

## 1.1系统描述

本项目是一个基于树莓派的多功能相机，主要包括以下功能：相机基本功能的模拟；延时视频的自动拍摄与处理；用作直播推流设备；用作个人或家庭监控设备；照片与视频文件的自动远程同步。

该项目基于树莓派4B单板电脑进行开发。在硬件方面，主要使用树莓派的CSI接口连接摄像头模块，使树莓派具备照片与视频的拍摄功能；其次，使用树莓派的GPIO引脚连接各个控制和显示部件，包括用于控制照片拍摄、视频拍摄、照片回放等功能的多个按钮，用于指示视频拍摄和相机开关状态的LED小灯，以及用于回看照片的LCD屏幕。在软件方面，主要使用python语言进行控制程序的编写，现有的gpiozero和picamera等模块提供了丰富的api，为实现各个控制部件及显示部件相应的功能提供了极大的便利。

对于延时视频拍摄与处理功能，则是将拍摄照片的基本功能与视频处理软件的进行结合。延时视频的本质就是固定时间间隔拍摄的照片的合成，因此，制作延时视频就是拍摄照片，通过程序设置拍摄时长，与拍摄照片的间隔，实现照片的自动的拍摄，再调用视频处理软件，将拍摄的照片合成为最终的延时视频。

而直播推流、个人监控等设备和远程同步功能则主要利用shell脚本，通过Linux管道，脚本开机自启等功能来实现。

## 1.2 硬件环境

CPU：ARM Cortex-A53 1.2GHz 四核；内存：1GB及以上；硬盘：16G及以上。

摄像头：Raspberry Pi Camera V1

屏幕：1.8inch SPI Module MSP1803

## 1.3 软件环境

Raspbian，Bash Shell

# 2 操作说明

**2.1 基本相机功能操作**

1. 相机模式的开关

由于本项目在设计时，将相机功能模块与其他功能模块分离，因此有一个独立的开关用于进入与退出相机模式。当树莓派通电开机后，即可点击该开关按钮(如图1)，进入相机模式，若要退出，则再次点击该按钮即可。当进入相机模式时，一个红色LED指示灯将会亮起 (如图2)，指示当前相机模式打开，可以使用所有的拍摄功能

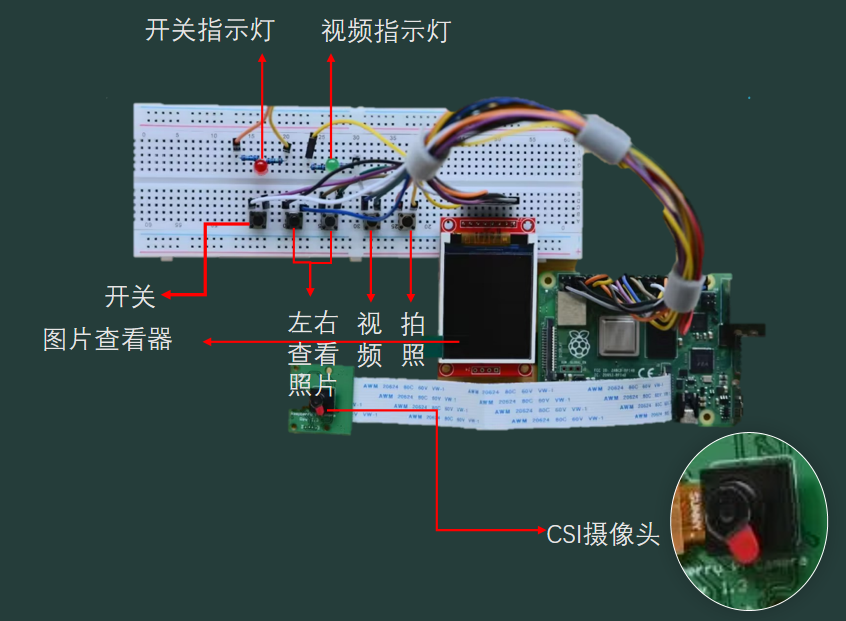


图1 相机硬件功能展示图

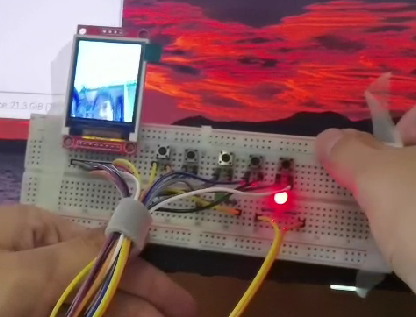


图2 相机开关及指示灯

1. 拍摄照片

同一般的相机一样，照片拍摄都是通过快门完成，本项目也将一个按钮(如图1)模拟成相机的快门。在相机模式下，通过点击一次该按钮，即可完成一张照片的拍摄。

1. 拍摄视频

对于视频的拍摄，本项目并没有将其与照片拍摄的功能结合在一个按钮之上，而是单独设置了一个按钮，同时还额外设置了一个视频拍摄指示灯，即为一个绿色的LED小灯(如图3），当按下视频拍摄按钮，小灯亮起，即视频拍摄开始进行，而当要停止视频拍摄时，再次按下按钮，小灯熄灭，拍摄视频结束。

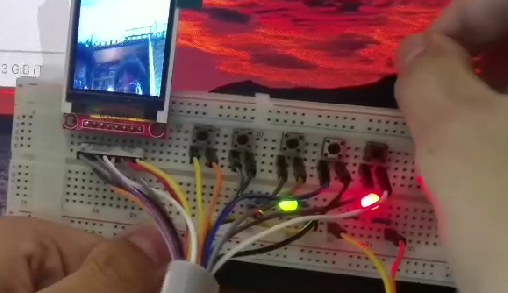


图3 视频拍摄指示灯

1. 回放照片

对于照片的回放功能，本项目设置了两个按钮，不同于一般相机，通常设有一个进入回放界面的按钮，而这两个按钮（如图1）都能够查看照片，第一次随便点击哪一个按钮，屏幕上都会显示最新的照片，而后再次点击时，左边按钮向前查看一张照片（即更早拍的一张），右边按钮向后查看一张照片（即更近拍的一张照片）。

**2.2 延时视频拍摄功能**

1. 间隔照片的拍摄

raspistill应用程序具有内置的延时模式，使用--timelapse（或-tl）命令行开关。

切换之后的值是拍摄之间的时间（以毫秒为单位）。

raspistill -t 30000 -tl 2000 -o image%04d.jpg

请注意%04d输出文件名中的数字：这表示文件名中要显示照片编号。因此，例如，上面的命令将每隔两秒（2000毫秒）产生一次照片的捕获，总时间为30秒（30000毫秒），名为image0001.jpg，image0002.jpg，依此类推，直到image0015.jpg。

该%04d指示四位数字，有前导零添加，以弥补所需的位数。因此，例如，%08d会产生一个八位数的数字。0 这个位,如果不想要前导零，可以不加。

如果输入的- t 为0，则应用程序将以其最小时间间隔进行拍摄照片。请注意，捕获之间存在大约30毫秒的最小强制暂停，以确保可以进行曝光计算。

1. 将图像合成为视频

通过使用avconv软件，将JPEG文件转换为H264格式的视频文件

avconv -r 24 -i image%04d.jpg -r 24 -vcodec libx264 -vf scale=800:480 timelapse.mp4

**2.3 直播推流**

使用树莓派进行直播推流主要使用树莓派相机拍摄视频的功能，通过linux中的管道将直播视频数据流pipe到一个视频处理软件，然后通过该软件，将视频数据实时推送到你的直播rtmp地址（该地址可以在直播平台开启直播后获取）。

通过以下脚本即可实现以上功能：

raspivid -o - -t 0 -vf -hf -fps 30 -b 6000000 | ffmpeg -re -stream\_loop -1 -i "/home/pi/music/你喜欢的音乐.mp3" -f h264 -i - -vcodec copy -acodec aac -b:a 192k -f flv "你的rtmp地址/你的直播码"

raspivid -o - -t 0 -vf -hf -fps 30 -b 6000000这个命令是用raspivid从摄像头流中以每秒30帧读取数据，然后通过管道符将视频数据传送给ffmpeg， ffmpeg将视频和音频编码后发送到rtmp服务器。注意：如果你的摄像头画面存在翻转，可以在raspivid命令后加上"-rot 180"命令对画面进行180度翻转。

**2.4 监控功能设置**

监控功能的设置也同样相对比较简单，主要运用到一个名为motion的软件，该软件可以通过包管理器下载获得（sudo apt-get install motion）。

对于该软件的配置如下：

1. 首先将motion软件的后台进程改为开启，让它能够一直在后台运行。将文件/etc/default/motion中 “start\_motion\_daemon=no” 的no改为yes。
2. 打开motion的配置文件/etc/motion/motion.conf，将相应配置更改为如下：

daemon on #off改成on

width 640

height 480 #根据摄像头像素自行更改

framerate 50 #帧率

stream\_maxrate 200

stream\_localhost off #设为off

1. 重启motion进程，关闭：sudo killall -TERM motion，打开：sudo motion 。
2. 在与树莓派同一网段内的设备中访问http://树莓派IP:8081，即可看到相应的监控画面（如图4）。

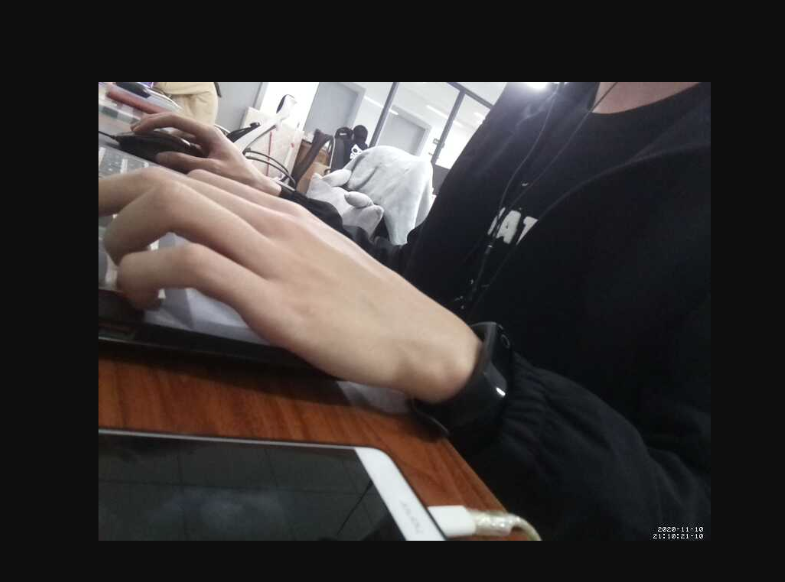


图4 监控画面

**2.5 自动同步远程仓库的配置**

该功能主要利用git、shell脚本以及crontab这一定期执行程序的命令。

1. 首先，我们需要在需要同步的目录下（即照片和视频文件存放的目录下)，初始化一个git仓库：

git init

1. 然后将你的本地仓库与远程仓库关联起来（这里以github仓库为例）：

git add remote origin <https://github/username/reporitory>

1. 使用脚本来将本地仓库与远程仓库同步,执行以下脚本即可将实现同步的功能：

#!/bin/bash

cd /path/to/your/reporitory

git add .

git commit -m "commit imformation"

git push -u origin master

1. 使用crontab命令来实现定时自动同步仓库

\* \* \* \* \* /Users/3y/autoSave.sh > ~/b.txt 2>&1 &

* \* \* \* \* \* 标志我这行命令需要在每分钟执行一次
* /Users/3y/autoSave.sh 标志我的脚本位置（这里需要用****绝对路径****）
* > ~/b.txt 2>&1 & 将脚本执行后的结果 输入到b.txt文件上