电脑屏幕的照片

低可信度描述已自动生成

# V3S-PI 基于全志V3S的随身终端

还记得去年发布的ESP32-S3-PIE吗？当初的项目，后面做完发现ESP32基本上做不了什么，于是，在今年的年底，作者学习了Linux，从F1C200S开始，摸透了全志平台后，制作了一款基于V3S芯片的随身终端，经过半个月的调试，项目已经基本调通。

V3S-PI 基于全志V3S的随身终端，项目使用四层板制作，全板采用0603电容电阻，相较于0402，制作更为方便，同时成本可压缩至100以内。

## 更新日志

2023/12/31 **首次发布**

## 芯片概述

全志V3S处理器集成了单个arm Cortex-A7 CPU，该CPU以高达1.2GHz运行，支持众多外围设备。处理器针对不断增长的汽车数字视频记录（DVR）和IP摄像机（IPC）监控系统的需求。512Mbit DDR2在V3中高度集成，此外，V3S对SPINAND/ NOR闪存，SD / MMC具有优化的外部存储器接口。包含专用视频引擎以提供高级多媒体应用程序和服务。视频引擎支持多种格式，如720P @ 60FPS，H.264解码器，1080p。

## 产品特性

1. 全志V3S主控，采用单核Cortex-A7，带硬件浮点，相较于F1C200S的A9内核有较大提升;
2. 芯片内封64Mbyte DDR2，无需外挂DDR2芯片;
3. RTL8723BS 2.4G WIFI(SDIO接口);
4. 通过CH334R扩展双USB口;
5. 耳机接口，采用V3S内置声卡;
6. 10M/100M以太网接口;
7. SDMMC接口，用于启动系统;
8. CH340N 串口转USB，用于连接串口终端;
9. LRADC，四个ADC按键（暂未调通）;
10. 2.4寸LCD彩屏，使用SPI接口;
11. 2x8扩展接口，用于扩展I2C/UART。

## 软件开源

在目前网上资料的基础上，作者自行移植了主线内核，Uboot以及root，目前仅支持SD卡以及SPI Nor启动，SPI Nand启动请不要参考当前代码。

硬件解码请考虑使用大佬打包的5.4内核。

### UBoot

<https://gitee.com/fhcloud/uboot-v3s>

快速开始

|  |
| --- |
| git clone <https://gitee.com/fhcloud/uboot-v3s>  cd uboot-v3s  make v3s\_pi\_defconfig  make |

输出的文件在源码根目录u-boot-sunxi-with-spl.bin

### Linux

<https://gitee.com/fhcloud/linux-v3s>

快速开始

|  |
| --- |
| git clone <https://gitee.com/fhcloud/linux-v3s>  cd linux-v3s  make v3s\_pi\_defconfig  make |

输出的内核文件在arch/arm/boot/zImage

输出的设备树请使用arch/arm/boot/dts/sun8i-v3s-pi.dtb

### Buildroot

<https://gitee.com/fhcloud/buildroot-v3s>

快速开始

|  |
| --- |
| git clone https://gitee.com/fhcloud/buildroot-v3s.git  cd buildroot-v3s  make v3s\_pi\_defconfig  make |

输出的文件在output/images/rootfs.tar

测试镜像账号root，密码123456

## 硬件开源

LCPI开源资料：<http://wiki.lcmaker.com/index.php?title=LC-PI-V3S>

V3S芯片数据手册：（见附件）Allwinner\_V3s\_Datasheet\_V1.0\_2.pdf

V3S硬件设计指南：（见附件）V3s硬件设计指南V1.0\_20150519.pdf

V3S公版原理图（可参考）：（见附件）V3S\_CDR\_STD\_V1\_0\_20180322.pdf

2.4寸彩屏数据手册：（见附件）LH240K-IG01.pdf

## 物料参考购买地址

V3S（这家应该是翻新，不过能用）：

<https://item.taobao.com/item.htm?spm=a1z09.2.0.0.21792e8dncSlgg&id=738899384776&_u=v2dklb45d0fc>

2.4’’彩屏（选焊接18PIN）：

<https://item.taobao.com/item.htm?spm=a1z09.2.0.0.21792e8dncSlgg&id=737889095324&_u=v2dklb458b08>

贴片排针（选2\*8P 贴片排针）：

<https://item.taobao.com/item.htm?_u=v2dklb45a979&id=644273830279&spm=a1z09.2.0.0.21792e8dncSlgg>

USB母座（选四脚贴片（黑胶/有边））：

<https://item.taobao.com/item.htm?_u=v2dklb45ea3a&id=674663700649&spm=a1z09.2.0.0.21792e8dncSlgg>

CH334R（选CH334R）:

<https://item.taobao.com/item.htm?spm=a1z09.2.0.0.21792e8dncSlgg&id=685199066514&_u=v2dklb454d3a>

RTL8723BS:

<https://item.taobao.com/item.htm?spm=a1z09.2.0.0.21792e8dncSlgg&id=601770157864&_u=v2dklb45bf86>

耳机座（选PJ-342 镀金 耳机孔 3.5mm(5个)）：

<https://detail.tmall.com/item.htm?_u=v2dklb4575c2&id=14476732895&spm=a1z09.2.0.0.21792e8dncSlgg>

HR911105A RJ45连接器（这家应该是盗版，不过能用）：

<https://item.taobao.com/item.htm?spm=a1z09.2.0.0.21792e8dncSlgg&id=717161946752&_u=v2dklb45fab3>

SD卡接口（淘宝找个类似的就行，不用立创）：

<https://item.szlcsc.com/5851216.html>

IPEX座：

淘宝，选择（IPEX 内针 1代）即可

## 硬件方案

### 全志V3S主控

本项目主控采用全志V3S，详细说明已在上方说明，需要注意的是R3/R5电阻需使用1%，同时，EPHY存在两组电压，下方原理图中已标出，原理图可参考下方图1：

图示, 示意图

描述已自动生成

图1 主控原理图

### ADC按键

板载四个ADC按键，电压跨度为0.2V，通过分压电阻实现，该模块原理图如图2所示：

图示, 示意图

描述已自动生成

图2 ADC按键

### 2x16 PIN扩展接口

此处通过排针，引出了8个扩展接口，包含了一个串口/一个I2C总线，可用于外接其他设备，原理图参考图3所示。

图示, 示意图

描述已自动生成

图3 扩展接口

### 辅助供电

板载四组供电芯片，分别产生3.0V 3.3V 1.8V 1.2V，其中,3.0V用于PLL以及AVCC模拟供电，3.3V负责芯片IO以及其他外设供电，1.8V用于内存供电，1.2V用于V3S核心主控供电，三路DCDC最大输出电流2A，该部分原理图如图4所示：

图示

描述已自动生成

图4 辅助供电

### WIFI模块

板载2.4G WIFI模块，使用RTL8723BS模块，V3S与WIFI模块之间采用SDIO接口连接，引出一个IPEX接口用于外接天线，R20-R27为SDIO和芯片要求的上拉电阻，C50/C51/C70为外围滤波电容，模块采用3.3V供电。

图示, 示意图

描述已自动生成

图5 WIFI模块原理图

### 10M/100M以太网接口

百兆以太网需要使用两组差分，分别为RX/TX差分，以太网PHY一般带自动翻转功能，所以RX/TX可以对调，V3S以太网使用电压驱动，只需在网口变压器中心抽头处加100NF电容到地即可，内部封装的电阻和2KV电容用于泄放主板的静电，防止高压打坏主控芯片,R28/R29为LED限流电阻,SH1 SH2为外壳，直接接地即可，该部分原理图如图6所示。

图示, 示意图

描述已自动生成

图6 以太网接口原理图

### 2.4’’ LCD彩屏

2.4寸彩屏通过SPI与主控进行通信，PE8接彩屏重置线，PE9接DC,用于区分数据/指令，LEDA为背光阳极，LEDK接Q1 MOS，用于主控控制背光开关，屏幕通过焊锡焊接到PCB板上，该部分原理图如图7所示。

图示, 示意图

描述已自动生成

图7 2.4‘’ LCD彩屏原理图

### CH340 串口转USB

使用CH340N芯片，实现串口转USB，用于电脑连接终端,R30/R31用于确保双头TYPEC线正常供电，D1为TVS，用于保护USB接口，此处使用3.3V供电，所以CH340N的VCC和V3需要连接到一起，同时增加100nf电容（具体参考CH340N手册）。该部分原理图可参考图8：

图示, 示意图

描述已自动生成

图8 CH340串口转USB

### CH334R 1转4 USB HUB

通过CH334R芯片，实现一转四个USB Host口，此处采用统一供电，所以VDD33与V5连接到一起即可，XI XO输入12Mhz晶振，DPU/DMU接上行USB口。

图示, 示意图

描述已自动生成

图示, 示意图

描述已自动生成

图示, 示意图

描述已自动生成

图9 CH334R原理图

### 耳机接口

耳机接口采用PJ-342接口，带音频和录音，该部分原理图参考图10：

图示, 示意图

描述已自动生成

图10 PJ-342耳机接口

### SD卡接口

SD卡接口用于插入MicroSD卡，R32-R36为SD卡上拉电阻，这里CLK不需要上拉，否则可能会影响SDIO通信，该部分原理图请参考图11。

图示, 示意图

描述已自动生成

图11 SD卡接口

## 软件方案

### Linux内核

内核采用Linux5.15.143主线内核，在其基础上，修改了部分代码，适配V3S大部分外设。

### 2.4寸LCD彩屏驱动

LCD彩屏使用SPI接口与主控连接，软件部分采用TinyDRM，与传统的fbtft相比，不再是固定帧率刷新，同时采用DRM架构，可以与新架构程序更快集成，由于各家屏幕初始化代码不同，所以st7735r.c文件在原来的基础上，修改了初始化代码，按照原有代码，我们只需修改st7735r\_pipe\_enable这个函数中的代码即可，修改完成的代码参考下方文件：

<https://gitee.com/fhcloud/linux-v3s/blob/master/drivers/gpu/drm/tiny/st7735r.c#>

设备树配置参考：

|  |
| --- |
| &spi0 {  status = "okay";  pinctrl-names = "default";  pinctrl-0 = <&spi0\_pins>;  display@0 {  compatible = "jianda,jd-t18003-t01";  reg = <0>;  spi-max-frequency = <95000000>;  backlight = <&panel\_backlight>;  dc-gpios = <&pio 4 9 0>; // PE9  reset-gpios = <&pio 4 8 0>; // PE8  rotation = <0>;  };  }; |

除spi外，需要添加一个背光节点，这样就可以在用户层操作背光：

|  |
| --- |
| panel\_backlight: panel-backlight {  compatible = "gpio-backlight";  gpios = <&pio 4 7 GPIO\_ACTIVE\_HIGH>; // PE7  default-on;  status = "okay";  }; |

### USB

设备树配置参考：

|  |
| --- |
| &usb\_otg {  dr\_mode = "host"; // peripheral  status = "okay";  };  &usbphy {  status = "okay";  }; |

除引用节点外，一定要添加ochi/echi，不然USB插入会没有反应：

|  |
| --- |
| soc {  ehci0: usb@01c1a000 {  compatible = "allwinner,sun8i-v3s-ehci", "generic-ehci";  reg = <0x01c1a000 0x100>;  interrupts = <GIC\_SPI 72 IRQ\_TYPE\_LEVEL\_HIGH>;  clocks = <&ccu CLK\_BUS\_EHCI0>, <&ccu CLK\_BUS\_OHCI0>;  resets = <&ccu RST\_BUS\_EHCI0>, <&ccu RST\_BUS\_OHCI0>;  status = "okay";  };  ohci0: usb@01c1a400 {  compatible = "allwinner,sun8i-v3s-ohci", "generic-ohci";  reg = <0x01c1a400 0x100>;  interrupts = <GIC\_SPI 73 IRQ\_TYPE\_LEVEL\_HIGH>;  clocks = <&ccu CLK\_BUS\_EHCI0>, <&ccu CLK\_BUS\_OHCI0>,  <&ccu CLK\_USB\_OHCI0>;  resets = <&ccu RST\_BUS\_EHCI0>, <&ccu RST\_BUS\_OHCI0>;  status = "okay";  };  }; |

### 以太网

直接引用DTSI文件中的emac节点即可，设备树参考：

|  |
| --- |
| &emac {  allwinner,leds-active-low;  status = "okay";  }; |

### 无线网卡RTL8723BS

Linux下SDIO架构类似于USB，设备插入后，如果有对应驱动，会自动加载，所以设备树中，我们只需要配置好MMC1接口，然后编译对应的驱动ko，在rootfs中加载即可，设备树配置参考下方：

|  |
| --- |
| &mmc1 {  broken-cd;  bus-width = <4>;  vmmc-supply = <&reg\_vcc3v3>;  status = "okay";  }; |

WIFI配置可参考资料：<https://zhuanlan.zhihu.com/p/665212465?utm_id=0>

编译完成的ko文件在buildroot的overlay目录下面，可以参考下方资料：

<https://gitee.com/fhcloud/buildroot-v3s/tree/master/board/v3s-pi/overlay/usr/lib/modules/5.15.143-v3s-pi>+

r8723bs.ko就是编译完成的内核模块，此处的模块backport了5.19内核的驱动，相比于5.15的驱动更加稳定。

除ko文件外，此处还需要加载网卡固件，参考下方目录：

<https://gitee.com/fhcloud/buildroot-v3s/tree/master/board/v3s-pi/overlay/usr/lib/firmware/rtlwifi>

rtl8723bs\_nic.bin可以从github或者其他网站上找到。

连接WIFI请参考下方脚本：

|  |
| --- |
| [root@buildroot ~]# cat conn.sh  modprobe r8723bs.ko  wpa\_supplicant -B -i wlan0 -c /etc/wpa\_supplicant.conf  udhcpc -i wlan0 |

配置文件请参考下方：

|  |
| --- |
| ctrl\_interface=/var/run/wpa\_supplicant  ap\_scan=1  network={  ssid="###############"  psk="###############"  } |

### LVGL

LVGL参考官方提供的fb驱动：<https://github.com/lvgl/lv_port_linux_frame_buffer>

WSL配置好环境之后，可以直接通过CLion或者VSCode在Windows端进行开发。

照片中的程序代码还在整理，后续会进行开源。

### ADC按键

测试文件参考下方：

|  |
| --- |
| #include <sys/types.h>  #include <sys/stat.h>  #include <fcntl.h>  #include <stdio.h>  #include <poll.h>  #include <signal.h>  #include <sys/types.h>  #include <unistd.h>  #include <fcntl.h>  #include <linux/input.h>  int fd;  void my\_signal\_fun(int signum)  {  struct input\_event buttons\_event, leds\_event;  /\* [cgw]: 异步通知产生时返回的数据 \*/  read(fd, &buttons\_event, sizeof(struct input\_event));  /\* [cgw]: 打印事件类型，事件码，事件值 \*/  printf("type: 0x%x code: 0x%x value: 0x%x\n", buttons\_event.type, buttons\_event.code, buttons\_event.value);  // /\* [cgw]: 返回的是KEY\_L或KEY\_S值 \*/  // if (buttons\_event.code == KEY\_L || buttons\_event.code == KEY\_S) {  // /\* [cgw]: 按键弹起 \*/  // if (buttons\_event.value == 0) {  // /\* [cgw]: 构造一个EV\_LED事件 \*/  // //leds\_event.type = EV\_SND;  // leds\_event.type = EV\_LED;  // //leds\_event.code = SND\_BELL;  // leds\_event.code = LED\_MUTE;  // /\* [cgw]: KEY\_L和KEY\_S控制LED的亮灭 \*/  // if (buttons\_event.code == KEY\_L) {  // leds\_event.value = 0xAA;  // } else if (buttons\_event.code == KEY\_S) {  // leds\_event.value = 0xEE;  // }  // /\* [cgw]: 发送LED控制事件 \*/  // write(fd, &leds\_event, sizeof(struct input\_event));  // printf("led write!\n");  // }  // }  }  int main(int argc, char \*\* argv)  {  int Oflags;  /\* [cgw]: 设置需要处理的信号SIGIO，即输入文件会请求一个SIGIO  \* 信号，当有新数据到来这个信号会发给filp->f\_owner进程  \*/  signal(SIGIO, my\_signal\_fun);  fd = open("/dev/input/event0", O\_RDWR | O\_NONBLOCK);  // printf("fd = 0x%x\n", fd);  if(fd < 0) {  printf("can't open!\n");  }  /\* [cgw]: 根据文件标识符fd，设置能够获得这个文件的进程(owner)  \* getpid()获得当前进程ID  \*/  fcntl(fd, F\_SETOWN, getpid());  /\* [cgw]: 获得file->f\_flags标志 \*/  Oflags = fcntl(fd, F\_GETFL);  /\* [cgw]: 置位FASYNC使能异步通知 \*/  fcntl(fd, F\_SETFL, Oflags | FASYNC);  while(1) {  /\* [cgw]: 休眠 \*/  sleep(1000);  }  return 0;  } |

### 音频播放

Buildroot中集成了alsa，默认声卡会静音状态，打开终端，输入alsamixer，首先解除静音：

在当前界面，选中Headphone，然后按下M键解除静音，然后使用键盘↑，调整音量到合适大小即可，界面可参考图12所示：

图片包含 图形用户界面

描述已自动生成

图12 alsamixer

调整完成后，输入mpv 文件名 --no-video，插入耳机，即可实现音乐播放：

|  |
| --- |
| [root@buildroot ~]# mpv 2.flac --no-video  Video --vid=1 [P] (mjpeg 500x500 1.000fps)  (+) Audio --aid=1 (flac 2ch 48000Hz)  File tags:  Album: 西厢寻他  Title: 西厢寻他  Track: 1  AO: [alsa] 48000Hz stereo 2ch s32  A: 00:00:04 / 00:03:43 (2%)  Exiting... (Quit) |

### 命令提示符显示当前目录

编辑/etc/profile文件，添加一行：

|  |
| --- |
| export PS1='[\u@\h \w]\$ ' |

然后export /etc/profile重新加载配置即可

### libcurl

通过libcurl获取当前天气，参考程序：

|  |
| --- |
| //  // Created by he on 2023/12/24.  //  #include "weather\_api.h"  extern pthread\_mutex\_t mutex;  extern lv\_obj\_t \*label, \*label1, \*label2, \*label3, \*label4;  size\_t now\_weather\_cb(char \* ptr, size\_t size, size\_t nmemb, void \* userdata)  {  // 解析json数据  cJSON \* json = cJSON\_ParseWithLength(ptr, size \* nmemb);  if(!json) {  goto end;  }  json = cJSON\_GetObjectItem(json, "now");  if(!json) {  goto end;  }  cJSON \* temp = cJSON\_GetObjectItem(json, "temp");  cJSON \* humidity = cJSON\_GetObjectItem(json, "humidity");  cJSON \* text = cJSON\_GetObjectItem(json, "text");  cJSON \* pressure = cJSON\_GetObjectItem(json, "pressure");  pthread\_mutex\_lock(&mutex);  if(cJSON\_IsString(text)) {  lv\_label\_set\_text\_fmt(label1, "当前天气\n%s", text->valuestring);  }  if(cJSON\_IsString(pressure)) {  lv\_label\_set\_text\_fmt(label2, "气压\n%s hPa", pressure->valuestring);  }  if(cJSON\_IsString(temp)) {  lv\_label\_set\_text\_fmt(label3, "室外温度\n%s ℃", temp->valuestring);  }  if(cJSON\_IsString(humidity)) {  lv\_label\_set\_text\_fmt(label4, "湿度\n%s %%", humidity->valuestring);  }  pthread\_mutex\_unlock(&mutex);  cJSON\_Delete(json);  end:  return size \* nmemb;  }  void get\_now\_weather(void)  {  char \* url =  "https://devapi.qweather.com/v7/weather/now?location=" LOCATION "&key=xxx";  CURL \* curl = curl\_easy\_init();  curl\_easy\_setopt(curl, CURLOPT\_URL, url);  curl\_easy\_setopt(curl, CURLOPT\_NOPROGRESS, 1L);  // curl\_easy\_setopt(curl, CURLOPT\_VERBOSE, 1L);  curl\_easy\_setopt(curl, CURLOPT\_SSL\_VERIFYPEER, 0L);  curl\_easy\_setopt(curl, CURLOPT\_WRITEFUNCTION, now\_weather\_cb);  curl\_easy\_setopt(curl, CURLOPT\_ENCODING, "gzip");  CURLcode res = curl\_easy\_perform(curl);  if(res != CURLE\_OK) {  printf("%d\n", res);  printf("访问失败\n");  }  curl\_easy\_cleanup(curl);  } |

编译时，请链接libcurl.so

|  |
| --- |
| target\_link\_libraries(${PROJECT\_NAME} PRIVATE curl) |

## 展示图片

卡通人物

中度可信度描述已自动生成

图片包含 游戏机, 电子, 电路

描述已自动生成

电子零件

中度可信度描述已自动生成

电子零件

中度可信度描述已自动生成

电脑屏幕的照片

低可信度描述已自动生成

电子设备

中度可信度描述已自动生成

## 写在结尾

本文写于2023/12/31的晚上，2023年的结尾，希望大家在新的一年里梦想成真,心想事成,事事顺利,福祉满家！