

CoDePi

高性能软 PLC 控制器

用户手册

主要特性：

- 全铝外壳，优秀的 EMC 屏蔽能力
- -20°C至+70°C宽温工作范围
- 兼容 12-24V 直流供电
- 1 个 1000Mbps, 1 个 10Mbps 以太网端口
- 内置 BMS 系统，默认提供 30 秒的关机时间
- 可通过 USB 拓展网口或 232/485 串口
- 双指示灯
- 提供 CoDeSys 包支持，支持以下工业总线：
 - Modbus-TCP Master / Slave
 - Modbus RTU Master / Slave
 - EtherNet/IP Scanner / Adapter
 - PROFINET Controller / Device
 - EtherCAT Master
 - CANopen Master / Slave
 - J1939
- 支持完整的 IEC61131-3 编程语言
- 支持 SoftMotion (CoDeSys 3.5.16.20)
- 支持 Visualization (Chromium WebVisu)
- 支持 Trace、断点、流控制 Debug
- 支持 SOEM
- 支持 Python、C、Node-Red 等第三方开发



CoDePi 是基于 Raspberry Pi Compute Module 4 的工业控制器，以极高的性价比提供基于 CoDeSys 的软 PLC 系统，适用于自动化行业、电力电子、智能家居等。

除了使用标准配置的 PLC 架构，我们还支持 SOEM 等非标准产品的研发，您可以使用诸如 DLIB、NODE-RED、DASH 等架构您自己的应用程序。

我们希望技术文档尽量简洁易用，您可以给我们发送电子邮件或访问我们的官方网站和 Github 获取更新的信息。

若要获取更多信息，您可以访问以下链接或联系您的经销商

官方网站: <https://www.codeaw.com/>

GITHUB: <https://github.com/feecat/codepi>

官方店铺: <https://shop178189185.taobao.com/>

官方邮箱: codeaw@hotmail.com

目录

- [基础架构-P3](#)
- [开始使用-P4](#)
- [进阶应用-P6](#)
- [常用指令-P7](#)
- [从源构建-P8](#)
- [常见问题-P12](#)

基础架构

硬件架构

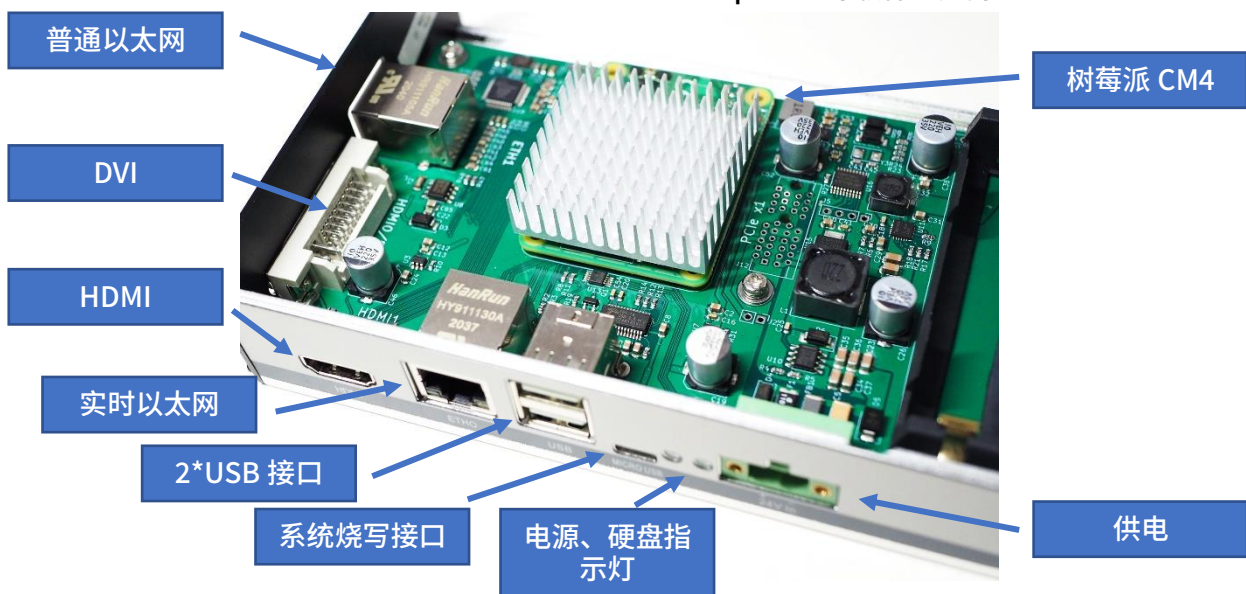
内部由标准的 Raspberry Pi Compute Module 4 单板计算模块与外围附件组成，外围附件包括：

- [1] 充放电电路及 UPS 电源管理（BMS）
- [2] 提供 2 路 USB 及 1 路 1000Mbps 以太网
- [3] 内置 W5500 标准以太网
- [4] 集成 BOOT Select，可从 MicroUSB 刷机
- [5] 双指示灯

软件架构

基于 Debian Linux 深度定制的系统，定制驱动及 dtoverlay，默认安装 CoDeSys 并做好相关设置。

- [1] Raspbian 官方系统
- [2] Preempt RT 实时补丁
- [3] CoDeSys 运行环境
- [4] 附加补丁，使 PLC 符合 Easy-To-Use 目标
- [5] 可选 Python、C 与 C++、Node-Red、Dlib、OpenCV 等软件包支持



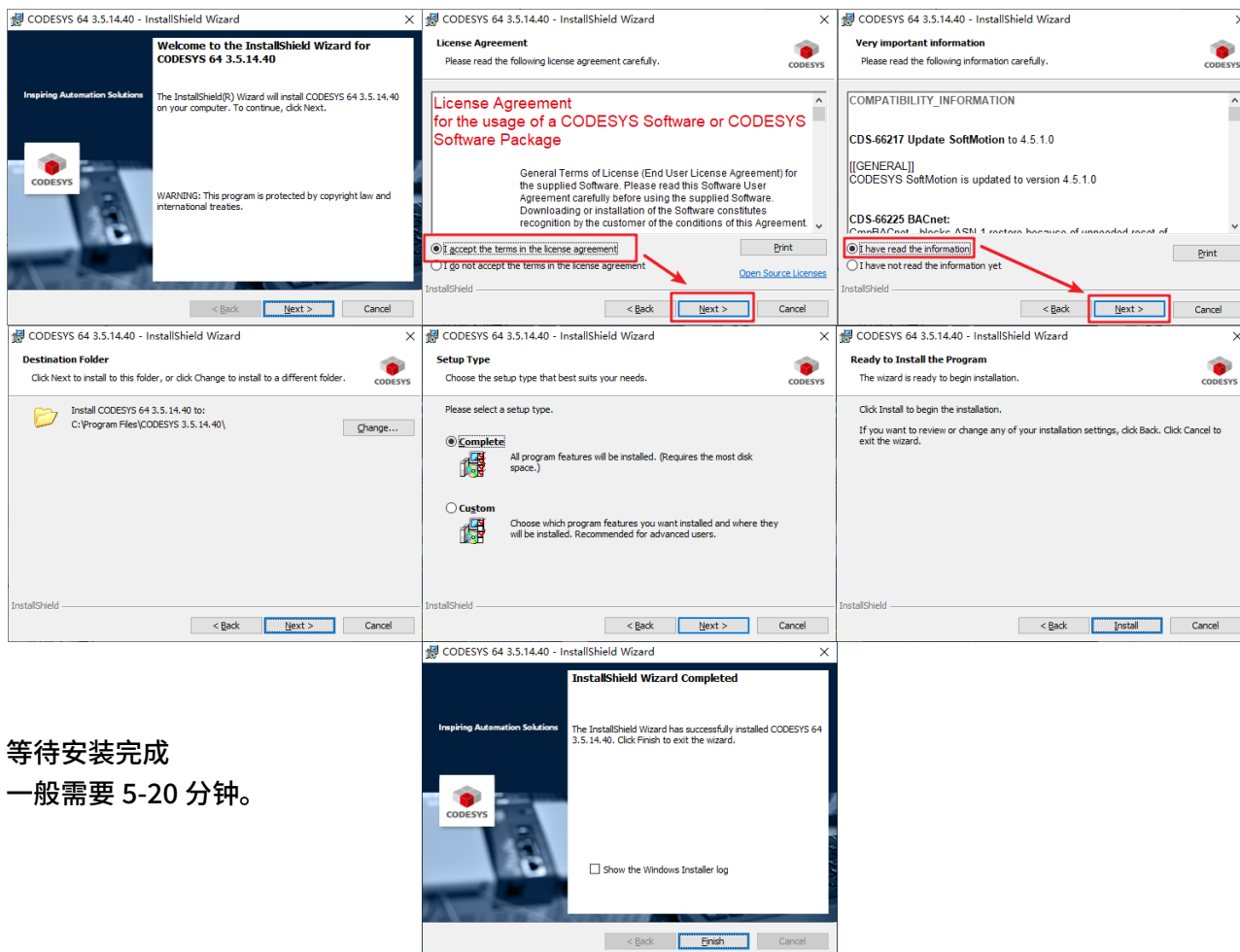
名称	配置	名称	配置
三围尺寸	150*90*30MM	重量	500g
安装方式	DIN 导轨	防护等级	IP20
供电电压	DC12-36V	功率	15W
工作温度	-20 to +70°C	存储温度	-40 to +85°C
处理器	BCM2711	CPU 频率	1.5GHz 4Core
RAM/运行内存	2G	FLASH/存储空间	8G
接口	2*USB2.0 TypeA	接口	1*HDMI+1*DVI
接口	2*RJ45 10/100Mbit/s	接口	1*MicroUSB
备用电源	2Hours	充电时间	4Hours(20->80%)
一般关机时间	15Sec	一般重新上电时间	30Sec
ESD/EMI	IEC61131-2	震动及粉尘	Passed

■安装 CoDeSys 运行环境

您可以向我们索取一切需要的文件。CoDeSys 安装包名称为 **CODESYS 3.5.16.20.exe** (32 位)、**CODESYS 64 3.5.16.20.exe** (64 位)，您也可以从 **3S 官方商店** 找到 CoDeSys 安装包。

如果您之前安装过其它版本的 CODESYS，请卸载并手动删除 C:\ProgramData\CODESYS 文件夹后再安装。

- 确保退出所有杀毒软件或阻拦启动项目的软件
- 双击对应系统版本的安装包开始安装，如下图



等待安装完成
一般需要 5-20 分钟。

安装完成后，确保右下角系统托盘显示有以下三个图标：



Codesys Control Win（用于仿真，32 位图标上没有.64 字样）

Codesys Gateway（用于扫描及通讯，如果没有开启则无法与 PLC 通讯）

Codemeter（用于授权）

■ 安装 Package

必选： 安装 **CODESYS Control for Raspberry PI 3.5.16.20.package**，设备描述及相关库的包文件。

可选： **EsclImageButton.package**，一种通过图片切换表达按钮状态的库。

OSCATBasic.package，一种开源库，包含一些常用指令。

OSCATBuilding.package，一种开源库，包含一些常用指令。

OSCATNetwork.package，一种开源库，包含一些常用指令。

■ 开机默认选项

我们默认开机自动打开 Chromium 并全屏访问 Visualization。首次连接时您可能需要接入显示器并连接键盘鼠标以进行测试及设置。

连接电源、显示器、键盘及鼠标后，您可以使用 **ALT+F4** 关闭 Chromium，选择桌面左上角的 **Terminal emulator** 打开终端，即可进行设置操作。

■ IP 设置

我们默认将 ETH0 的 IP 固定为 192.168.1.100，ETH1 的 IP 为 DHCP 协商。如果该方案不满足您的需求，可进行手动更改 IP 操作。具体操作步骤如下：

在终端输入 **sudo pcmanfm** 打开文件管理器，定位到 **/etc/dhcpd.conf**，将结尾处改为如下格式：

```
interface eth0
```

```
static ip_address=192.168.1.100      #<-将其修改为您希望 ETH0 固定的 IP 地址
```

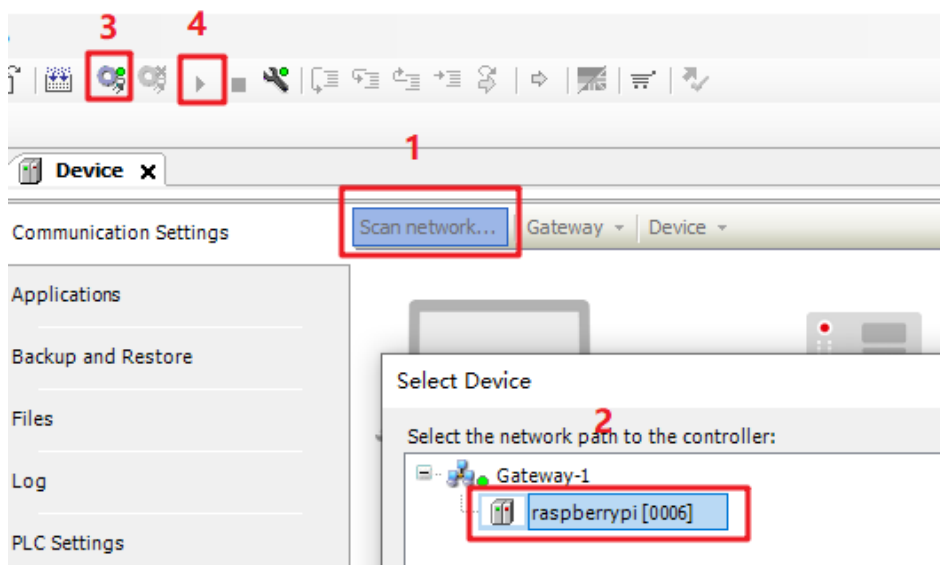
```
interface eth1
```

```
static ip_address=192.168.2.100      #<-将其修改为您希望 ETH1 固定的 IP 地址
```

请注意：多个以太网端口不可以设在同一网段下，否则会引起网络冲突。

■ 连接并下载程序

使用 CoDeSys 打开示范程序/Demo/Demo.project，转到 Device 页面，单击 Scan network，选择您扫描出的设备，单击 Download 进行下载，下载完毕后单击 Run 运行。



■ 恭喜

您已经完成了所有基础设置内容，您可以在 Demo Project 上架构您自己的应用，也可以将您预构建的应用下载到 PLC 中运行。如需更多帮助，请继续浏览或咨询我们。

■使用 SSH 连接

IP 设置完成后，您可以使用 SSH 连接。建议使用的工具在 [/Tools/Putty.exe](#)。

建立连接后，使用用户名：**pi**，密码：**pi** 登录。

我们强烈建议您使用 `passwd` 命令修改默认密码。

■打开 VNC 连接

在调试阶段您可能没有连接显示器，将 VNC 打开可以直观地进行桌面操作。

在建立连接的 putty 中输入 `sudo raspi-config`，转到 **Interfacing Options**，VNC

弹出提示框 **Would you like the VNC Server to be enabled?** 选 **Yes** 即可。

您可以使用 [/Tools/vnc-viewer-5.3.2-windows-32bit.exe](#) 进行 VNC 连接。

请注意：VNC 会不时弹出对话框用于传输文件，在最终交付产品上请关闭 VNC 服务。

■修改系统时间

进入终端后，使用如下命令修改时间：

```
sudo date -set '2020-01-01 12:00:00' #<-英文单引号内改为需要设置的时间
```

```
sudo hwclock -w
```

您可以使用 `sudo hwclock -r` 来检查当前时间。

■查看网卡状态

使用以下命令查看网卡状态

```
ifconfig
```

■手动关机或重启

使用以下命令立即关机：

```
sudo shutdown now
```

使用以下命令立即重启：

```
sudo reboot now
```

■禁用自动关机

编辑 [/boot/config.txt](#)，在 `dtoverlay=gpio-shutdown,gpio_pin=18` 最前面加 `#` 号保存并重启。

UPS 默认提供 30 秒关机时间，请注意在此之前必须完成关机流程。

■关闭 NTP 服务

若已连接到广域网（WAN），NTP 功能会自动更新时间，但它可能造成系统时钟抖动。

使用以下指令禁用 NTP 服务：

```
sudo systemctl stop systemd-timesyncd
```

```
sudo systemctl disable systemd-timesyncd
```

■传输文件

我们建议使用 WinSCP 或 FileZilla 进行连接，也可以使用 VNC 进行文件传输。

如需挂载外部 USB 存储，建议将桌面打开 (`lxpanel --profile LXDE-pi`)，可以自动挂载。

如需手动挂载，请参阅 <https://www.raspberrypi.org/documentation/configuration/external-storage.md>

■系统恢复

您可以使用 [/Tools/balenaEtcher-Portable-1.5.71.exe](#) 或 [Win32DiskImager](#) 进行系统恢复

在没有连接电源且系统关机状态下，先连接 USB 线缆到编程端口（MicroUSB），再连接供电。

在 balenaEtcher 中选择镜像 ([Linux\template.img](#))，选择 CM4 设备并烧写。

请注意：烧写后的系统中不包含 CoDeSys License，请根据备份内容进行恢复或咨询我们。

常用指令

下表为常用 Raspbian 指令，在终端中运行。

打开文件夹	<code>sudo pcmanfm</code>
打开桌面	<code>lxpanel --profile LXDE-pi</code>
查看网卡状态	<code>ifconfig</code>
关机	<code>sudo shutdown now</code>
重启	<code>sudo reboot now</code>
打开设置	<code>sudo raspi-config</code>
诊断开机过程	<code>dmesg</code>
诊断 DT 覆盖层加载	<code>sudo vcdbg log msg</code>
获取 GPIO 状态	<code>raspi-gpio get 18</code>
获取内核版本	<code>uname -r</code>
查看运行中的任务	<code>top</code>
查看运行中的任务	<code>htop</code>
结束进程	<code>kill</code>
写入日期时间	<code>sudo date -set '2020-01-01 00:00:00'</code> <code>sudo hwclock -w</code>
读取日期时间	<code>sudo hwclock -r</code>
手动设置 IP 地址（临时）	<code>sudo ifconfig eth0 192.168.0.100</code>
手动挂载外部设备（临时）	<code>sudo mount /dev/sda4* /media/pi</code>

更多指令和指令说明，请参阅 <https://git.io/linux> 和 <https://github.com/feecat/codepi>

从源构建

如果您使用 Raspbian 官方镜像进行安装，则可在系统安装完成后跳到第二步，使用我们提供的内核包。如果您想从源构建，请从第一步开始。

1、从源构建

从源构建部分参考：<https://lemariva.com/blog/2018/07/raspberry-pi-preempt-rt-patching-tutorial-for-kernel-4-14-y>

获取包及工具：(rpi-4.14.y-rt 可替换为最新版本，例如 rpi-4.19.y-rt)

```
mkdir ~/rpi-kernel
cd ~/rpi-kernel
mkdir rt-kernel
git clone https://github.com/raspberrypi/linux.git -b rpi-4.14.y-rt
git clone https://github.com/raspberrypi/tools.git
```

克隆完毕（或从 github 下载后解压），开始设置系统变量：

```
export ARCH=arm
export CROSS_COMPILE=~/.rpi-kernel/tools/arm-bcm2708/gcc-linaro-arm-linux-gnueabihf-raspbian-x64/bin/arm-linux-gnueabihf-
export INSTALL_MOD_PATH=~/.rpi-kernel/rt-kernel
export INSTALL_DTBS_PATH=~/.rpi-kernel/rt-kernel
export KERNEL=kernel7-rt
cd ~/.rpi-kernel/linux/
make bcm2709_defconfig
```

之后修改配置

```
make menuconfig
```

最重要的配置为：

- 启用 CONFIG_PREEMPT_RT_FULL
- 启用 HIGH_RES_TIMERS
- 设置 CONFIG_HZ 为 1000Hz
- 启用 WIZnet W5100 SPI MODE 驱动支持

编译内核：（需要较长时间）

```
make -j4 zImage
make -j4 modules
make -j4 dtbs
make -j4 modules_install
make -j4 dtbs_install
```

打包：

```
mkdir $INSTALL_MOD_PATH/boot
./scripts/mkknlimg ./arch/arm/boot/zImage $INSTALL_MOD_PATH/boot/$KERNEL.img
cd $INSTALL_MOD_PATH
tar czf ../rt-kernel.tgz *
```


2、安装标准 Raspbian 系统、设置、启用实时内核

- [1] 使用 win32diskimager 或 balenaEtcher 烧写官方 raspbian 镜像
- [2] 开机、进行如下设置：
 - a) 国家地区选 china, chinese, shanghai
 - b) 勾选 use english language 和 use us keyboard
 - c) 之后有 skip 点 skip, 没有的点 next
 - d) 主菜单-preferences-raspberry pi configuration
 - e) Splash Screen-Disable
 - f) Interface 开启 SSH、VNC、SPI、I2C, 关闭其它选项。重启
- [3] 用 vncviewer 连接 raspberrypi, 用户名 pi, 密码 raspberry
- [4] 将内核文件 (rt-kernel.tgz) 和 dtbo 文件 (w5100-rpi.dtbo) 通过 VNC 传输到树莓派中
- [5] 打开终端, 输入 `sudo pcmanfm` 打开文件管理器
- [6] 在 /home/pi 下新建 tmp 文件夹, 将文件剪切进去。关闭文件管理器, 回到终端
- [7] 依次输入
 - a) `cd tmp`
 - b) `sudo tar xzf rt-kernel.tgz`
 - c) `cd boot`
 - d) `sudo cp -rd * /boot/`
 - e) `cd ../lib`
 - f) `sudo cp -rd * /lib/`
 - g) `cd ../overlays`
 - h) `sudo cp -d * /boot/overlays`
 - i) `cd ..`
 - j) `sudo cp -d bcm* /boot/`
 - k) `sudo cp -d w5100-rpi.dtbo /boot/overlays`
- [8] 输入 `sudo pcmanfm` 再次打开文件管理器
- [9] 进入 boot 文件夹, 在 config.txt 右键 Leafpad(或 mousepad)打开, 在末尾加上
 - a) `kernel=kernel7-rt.img`
 - b) `dtoverlay=w5500,int_pin=22,speed=10000000`
 - c) `hdmi_force_hotplug=1`
- [10] 同样的方法编辑 cmdline.txt, 最前面加上 `logo.nologo`
- [11] 关闭文件管理器, 返回终端, 输入 `sudo raspi-config` 回车
 - a) Interface i2c 打开
 - b) Interface Serial 关闭
 - c) Advance option 内 resolution 改为 DMT Mode82
 - d) Finish, reboot? yes
- [12] 开机后打开终端, 输入 `uname -r` 确认内核版本
 - a) 4.14.91-rt49-v7+ (以实际版本为准)
 - b) 输入 `ifconfig` 确认 eth 状态: ETH0、ETH1、lo
 - c) 将网线插在 W5500 网口上。重新开启 VNC
 - d) 终端输入 `i2cdetect -y 1` 确认 RTC 状态 (1、50 显示 68 或 UU)。

- [13] 终端输入 `sudo pcmanfm`, 修改 `/boot/config.txt`, 末尾加上
 - a) `dtparam=i2c_arm=on`
 - b) `dtoverlay=i2c-rtc,pcf8563`
 - c) `dtoverlay=gpio-shutdown,gpio_pin=4`
- [14] 关闭 `pcmanfm`, 终端输入 `sudo reboot now` 重启
- [15] 终端输入
 - a) `Sudo hwclock -systohc`
 - b) `Sudo apt-get purge fake-hwclock`
- [16] `sudo pcmanfm`, 在 `/etc/udev/rules.d/` 下创建 `85-hwclock.rules` 文件, 文件内容为
 - a) `KERNEL=="rtc0", RUN+=" /sbin/hwclock --rtc=$root/$name --hctosys"`
- [17] 关闭文件管理, 终端输入
 - a) `Sudo date --set '2020-01-01 12:00:00'` (改为正确时间)
 - b) `Sudo hwclock -w`
 - c) `Sudo hwclock -r`
 - d) 确认时间已经被写入硬件时钟内
- [18] 安装 CoDeSys
 - a) `Tools-update raspberry pi`
 - b) Name 填 `pi`, password 填 `raspberry`, IP 填 `192.168.1.100` (或当前树莓派 IP)
 - c) 点 `install`
 - d) Configure runtime 选 `standard` (单核) 或 `MultCore` (多核), `ok`
- [19] 编辑 `/etc/CODESYSControl_User.cfg`, 将
 - a) `Command.0=shutdown`
 - b) 改为
 - c) `Command=AllowAll`
- [20] 编辑 `/etc/dhcd.conf`, 在末尾加上
 - a) `interface eth0`
 - b) `static ip_address=192.168.1.100`
- [21] 编辑 `/etc/xdg/lxsession/LXDE-pi/autostart`, 在末尾加上
 - a) `@xset dpms 0 0 0`
 - b) `@xset s off`
 - c) `@/usr/bin/chromium-browser --kiosk --disable-restore-session-state --disable-translate --incognito http://localhost:8080/webvisu.htm`
- [22] 编辑 `/etc/modules-load.d/modules.conf`, 在所有内容前面加 `#`
- [23] 编辑 `/etc/systemd/system.conf`, 修改如下两行:
 - a) `DefaultTimeoutStartSec=10s`
 - b) `DefaultTimeoutStopSec=10s`
- [24] 清空 `bash` 记录: `history -c && history -w`
- [25] 使用 `image-backup` 备份整个系统
<https://github.com/scruss/RonR-RaspberryPi-image-utils>
- [26] 激活 CoDeSys License、上传 DEMO PLC、Reboot 检查、结束。

3、您还可以删除一些不必要的软件，此处提供了部分列表：

使用 `sudo apt-get remove --purge libreoffice*` 移除 libreoffice，下同
建议移除的列表有：

- libreoffice*
- claws-mail
- minecraft*
- python-games
- smartsim
- scratch*
- sense-emu*
- sonic-pi*

移除完成后，使用如下指令进行清理

```
sudo apt-get clean  
sudo apt-get autoremove
```

4、隔离核心

您可以在 `/boot/cmdline.txt` 中插入 `isolcpus=3` 以隔离最后一个核心，再在 CoDeSys 项目中将实时性任务设为第三个核心，这将有助于改善 MC 授权下运动控制和 PLC 逻辑的实时性。

5、关于计算模块的实时核心源

请参考 <https://github.com/DatanoiseTV/linux-rpi-5.4.y-rt>

Q: 为什么不使用 poweroff 来断开 UPS?

A: 设计目的是为了断电而准备, 单独的 PLC 或运动控制器没有长时间运行的必要。为了防止 Linux 卡死不能关机导致 UPS 电量耗尽, 规定关机拥有 30 秒时间, 该时间可以通过烧写固件修改。

Q: 使用多个 SoftPLC 并连接到 DHCP 服务器, 但 DHCP 没有正确配置

A: 删除 dhcpid: `sudo rm /etc/dhcpd.duid`

Q: 如何禁用 NTP 自动同步

A: `sudo timedatectl set-ntp false`

Q: 在 SoftMotion 项目中开机偶尔发生通讯中断

A: PLC 带负荷时会影响运行周期, 您可以采用多核授权+隔离核心的方式规避。

Q: 我是新手, 如何开始学习 CoDeSys?

A: 参阅以下链接

<http://bbs.hicodesys.com/data/attachment/forum/201711/07/130521bhxfgyux80llxpxu.attach>

<https://item.jd.com/12401418.html>

