

淘宝店铺

优秀不够，你是否无可替代

知识从未如此性感。烂程序员关心的是代码,好程序员关心的是数据结构和它们之间的关系 --QQ群: 607064330 --本人
QQ:946029359 --淘宝 <https://shop411638453.taobao.com/>

随笔 - 779, 文章 - 0, 评论 - 325, 阅读 - 193万

导航

[博客园](#)
[首页](#)
[新随笔](#)
[联系](#)
[订阅](#) 
[管理](#)

公告



渡我不渡她 -

Not available

00:00 / 03:41

- 渡我不渡她
- 小镇姑娘
- PDD洪荒之力

 加入QQ群

昵称：杨奉武
 园龄：6年
 粉丝：670
 关注：1

搜索

我的标签

8266(88)
 MQTT(50)
 GPRS(33)
 SDK(29)
 Air202(28)
 云服务器(21)
 ESP8266(21)
 Lua(18)
 小程序(17)
 STM32(16)
 更多

随笔分类

Air724UG学习开发(5)
 Android(22)
 Android 开发(8)
 C# 开发(4)
 CH395Q学习开发(17)
 CH579M物联网开发(12)
 CH579M学习开发(8)
 ESP32学习开发(20)
 ESP8266 AT指令开发(基于STC89C52单片机)(3)
 ESP8266 AT指令开发(基于STM32)(1)
 ESP8266 AT指令开发基础入门篇备份(12)
 ESP8266 LUA脚本语言开发(13)

6-HC32F460(华大单片机)-时钟树

<p>
 <iframe name="ifd" src="https://mnifdv.cn/resource/cnblogs/LearnHC32F460"
 frameborder="0" scrolling="auto" width="100%" height="1500">
 </iframe>
 </p>

HC32F460(华大单片机)学习开发

开发板原理

图:<https://mnifdv.cn/resource/cnblogs/LearnHC32F460/HC32F460.PDF>

资料源码下载链

接:<https://github.com/yangfengwu45/learnHC32F460.git>

- 1-硬件使用说明
- 2-工程模板使用说明
- 3-GPIO输出高低电平
- 4-GPIO引脚电平检测
- 5-串口(基本使用)
- 6-时钟树

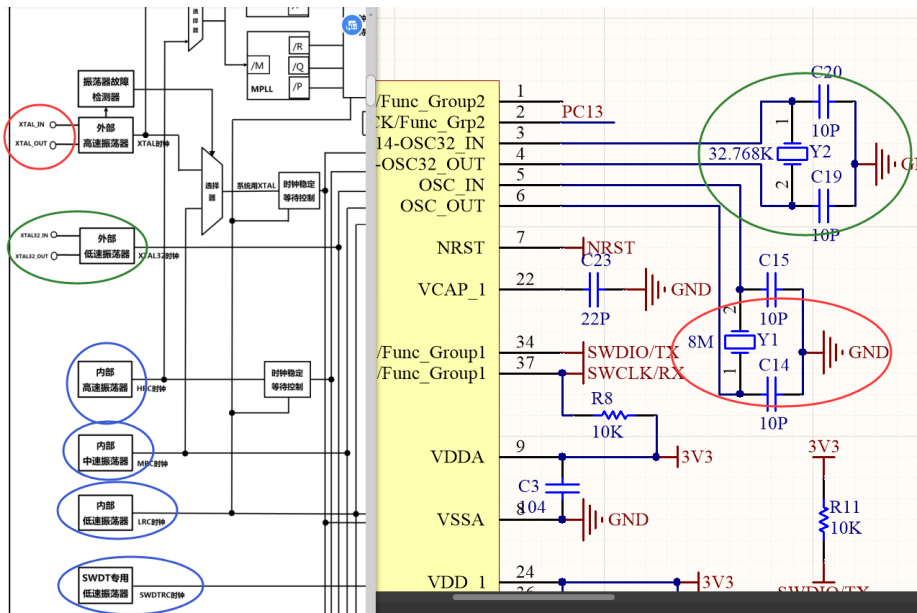
ESP8266 LUA开发基础入门篇
备份(22)
ESP8266 SDK开发(33)
ESP8266 SDK开发基础入门篇
备份(30)
GPRS Air202 LUA开发(11)
HC32F460(华大单片机)学习开
发(6)
NB-IOT Air302 AT指令和LUA
脚本语言开发(27)
PLC(三菱PLC)基础入门篇(2)
STM32+Air724UG(4G模组)
物联网开发(43)
STM32+BC26/260Y物联网开
发(37)
STM32+CH395Q(以太网)物
联网开发(24)
STM32+ESP8266(ZLESP8266/
物联网开发(1)
STM32+ESP8266+AIR202/30:
远程升级方案(16)
STM32+ESP8266+AIR202/30:
终端管理方案(6)
STM32+ESP8266+Air302物
联网开发(64)
STM32+W5500+AIR202/302
基本控制方案(25)
STM32+W5500+AIR202/302
远程升级方案(6)
UCOSii操作系统(1)
W5500 学习开发(8)
编程语言C#(11)
编程语言Lua脚本语言基础入
门篇(6)
编程语言Python(1)
单片机(LPC1778)LPC1778(2)
单片机(MSP430)开发基础入门
篇(4)
单片机(STC89C51)单片机开发
板学习入门篇(3)
单片机(STM32)基础入门篇(3)
单片机(STM32)综合应用系列
(16)
电路模块使用说明(12)
感想(6)
更多

阅读排行榜

1. ESP8266使用详解(AT,LUA, SDK)(173577)
2. 1-安装MQTT服务器(Windows),并连接测试(102388)
3. 用ESP8266+android,制作自己的WIFI小车(ESP8266篇)(66199)
4. ESP8266刷AT固件与node mcu固件(65997)
5. 有人WIFI模块使用详解(39020)
6. (一)基于阿里云的MQTT远程控制(Android 连接MQTT服务器,ESP8266连接MQTT服务器实现远程通信控制---简单的连接通信)(36515)
7. 关于TCP和MQTT之间的转换(34472)
8. C#中public与private与static(34393)
9. android 之TCP客户端编程(32506)
10. android服务端+eps8266+单片机+路由器之远程控制系统(31523)

下面接着介绍

1.单片机呢可以选择性的使用5路时钟,看门狗那个是单独给看门狗的.



推荐排行榜

- 1. C#委托+回调详解(9)
- 2. 用ESP8266+android,制作自己的WIFI小车(ESP8266篇)(8)
- 3. 用ESP8266+android,制作自己的WIFI小车(Android 软件)(6)
- 4. ESP8266使用详解(AT,LUA,SDK)(6)
- 5. 关于TCP和MQTT之间的转换(5)

最新评论

- 1. Re:用ESP8266+android,制作自己的WIFI小车 (Android 软件) 百度盘都失效了 --ghggaojian
- 2. Re:201-STM32+Air724UG基本控制篇(阿里云物联网平台)-设备使用物模型Topic上报温湿度数据 你好，有源码吗？ --zsw1997

时钟源	规格
外部高速振荡器（XTAL）	晶振的频率范围：4~24MHz 外部时钟输入：最高 24MHz 振荡器故障检测功能
外部低速振荡器（XTAL32）	晶振的频率范围：32.768KHz

内部高速振荡器（HRC）	频率：16MHz 或者 20MHz 用户可写寄存器对频率微调
内部中速振荡器（MRC）	频率：8MHz 用户可写寄存器对频率微调
内部低速振荡器（LRC）	频率：32.768KHz 用户可写寄存器对频率微调 可用作 RTC 的计数时钟,唤醒定时器 WKTM 的计数时钟, XTAL32 的备份时钟
SWDT 专用内部低速振荡器（SWDTRC）	频率：10KHz

2.上面的也只需要了解,下面的才是应该学习的

HCLK是系统时钟,当然也是EFM,SRAM.....

PCLK1是串口1-4, SPI1-4, Timer0等的时钟

书签	×
Q 书签查找	
🔍 📄 📁 📌	
4.3 总线功能	95
5 复位控制（RMU）	96
6 时钟控制器（CMU）	110
6.1 简介	110
6.2 系统框图	111
6.2.1 系统框图	111
6.2.2 时钟频率测量框图	112
6.3 时钟源规格	113
6.4 工作时钟规格	115
6.5 晶振电路	117
6.6 内部RC时钟	123
6.7 PLL时钟	125
6.8 时钟切换步骤	125
6.9 时钟输出功能	128
6.10 时钟频率测量	129
6.11 寄存器说明	131
7 电源控制（PWC）	166
8 初始化配置（ICG）	224
9 嵌入式FLASH（EFM）	230

6.4 工作时钟规格

时钟	作用范围	规格
HCLK	CPU、DMA(n=1、2)、EFM(主闪存)、SRAM0、SRAM1、SRAM2、SRAM3S、Ret-SRAM、MPU、GPIO、DCU、INTC、QSPI	最高频率 168MHz 可选时钟源的分频：1、2、4、8、16、32、64
PCLK0	Timer6 计数器用时钟	最高频率 168MHz 可选时钟源的分频：1、2、4、8、16、32、64
PCLK1	USARTn (n=1~4)、SPI(n=1~4)、USBFS (控制逻辑)、Timer0(n=1、2)、TimerAn(n=1~6)、Timer4(n=1~3)、Timer6 (控制逻辑)、EMB、CRC、HASH、AES、I2S(n=1~4)控制逻辑	最高频率 84MHz 可选时钟源的分频：1、2、4、8、16、32、64
PCLK2	AD 变换时钟	最高频率 60MHz 可选时钟源的分频：1、2、4、8、16、32、64 可独立选择时钟源：UPLLPUPLLQUPLLRLMPLLPMLLQMPLLR
PCLK3	RTC (控制逻辑)、I2C(n=1、2、3)、CMP、WDT、SWDT (控制逻辑)	最高频率 42MHz 可选时钟源的分频：1、2、4、8、16、32、64
PCLK4	ADC (控制逻辑)、TRNG	最高频率 84MHz 可选时钟源的分频：1、2、4、8、16、32、64 可独立选择时钟源：UPLLPUPLLQUPLLRLMPLLPMLLQMPLLR
EXCLK	SDIO(n=1、2)、CAN	最高频率 84MHz 可选时钟源的分频：1、2、4、8、16、32、64
UCLK	USBFS 通信用时钟	频率 48MHz 时钟源可选系统时钟分频 2、3、4。 可独立选择时钟源：UPLLPUPLLQUPLLRLMPLLPMLLQMPLLR
CANCLK	CAN 通信时钟	频率范围 4~24MHz

STICCLK	CPU 的 SysTickTimer 计数器用时钟, 时钟源为 LRC	可配置为时钟源 LRC 或系统时钟
SWDTCLK	SWDT 计数器用时钟	频率 10KHz
TCK	JTAG 用时钟	最高频率 25MHz
TPIUCLK	Cortex-M4 调试跟踪器用时钟	最高频率 42MHz 可选时钟源的分频: 1, 2, 4
I2SnCLK (n=1~4)	I2Sn(n=1~4)	最高频率 168MHz 可独立选择时钟源: UPLL, UPLLQ, UPLL, MPLL, MPLLQ, MPLLR

表 6-1 各个内部时钟的规格

注意:

各时钟之间需遵守下列规则:

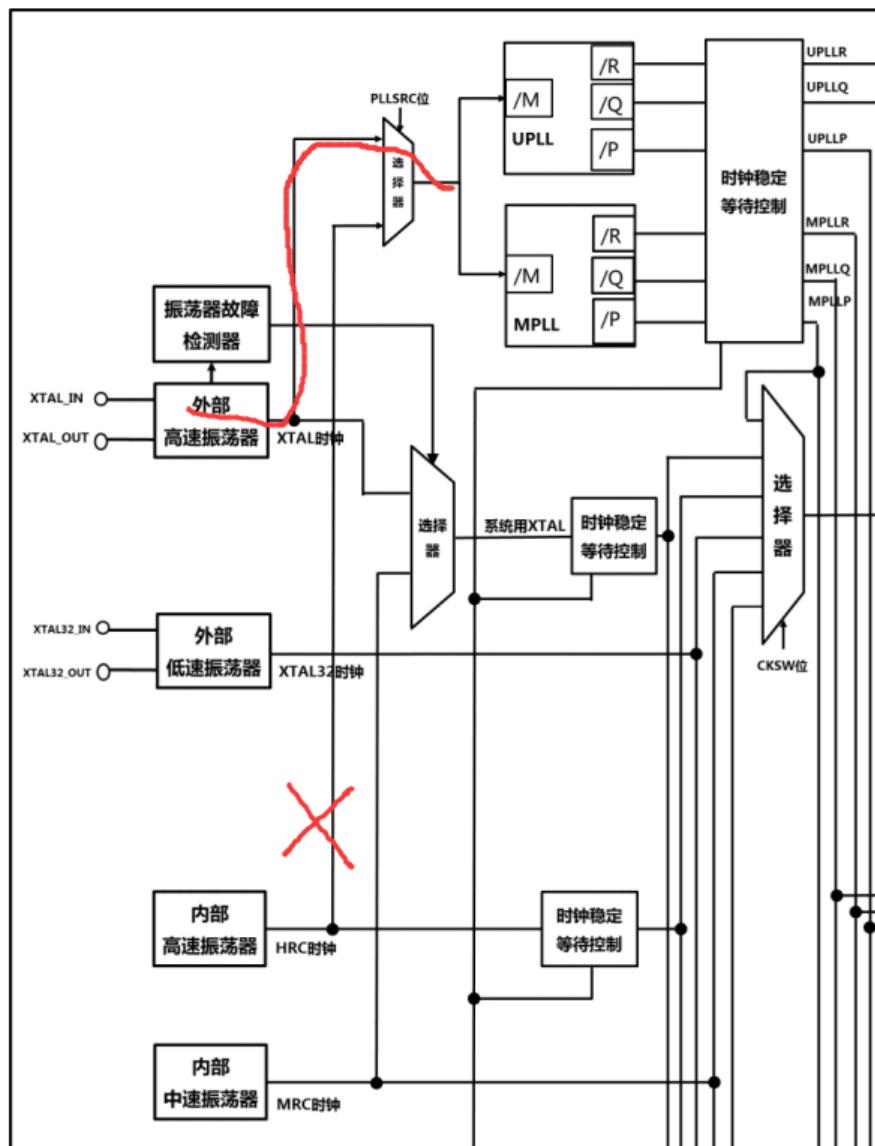
- HCLK 频率 \geq PCLK1 频率、HCLK 频率 \geq PCLK3 频率、HCLK 频率 \geq PCLK4 频率
- HCLK 频率: EXCLK 频率 = 2:1, 4:1, 8:1, 16:1, 32:1
- PCLK0 频率 \geq PCLK1 频率、PCLK0 频率 \geq PCLK3 频率
- HCLK 频率: PCLK0 频率 = N:1, 1:1
- PCLK2 频率: PCLK4 频率 = 1:8, 1:4, 1:2, 1:1, 2:1, 4:1, 8:1

3. 然后呢华大的几乎每个例程里面都有时钟的初始化程序

首先记住一句话, 时钟设置的时候是先配置好参数再去选择和启用时钟.

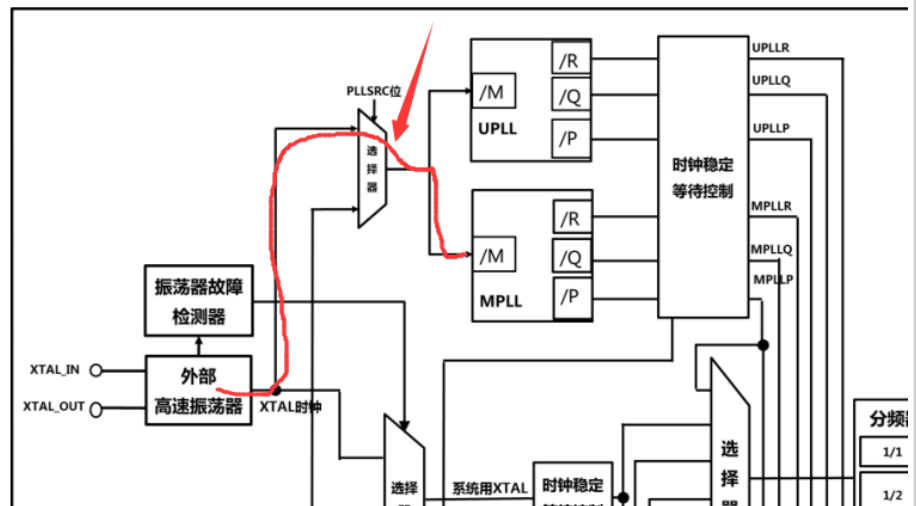
```

134  //*****
135  static void ClkInit(void)
136  {
137      stc_clk_xtal_cfg_t   stcXtalCfg;
138      stc_clk_mpll_cfg_t   stcMpllCfg;
139      en_clk_sys_source_t   enSysClkSrc;
140      stc_clk_sysclk_cfg_t stcSysClkCfg;
141
142      MEM_ZERO_STRUCT(enSysClkSrc);
143      MEM_ZERO_STRUCT(stcSysClkCfg);
144      MEM_ZERO_STRUCT(stcXtalCfg);
145      MEM_ZERO_STRUCT(stcMpllCfg);
146
147      /* Set bus clk div. */
148      stcSysClkCfg.enHclkDiv = ClkSysclkDiv1; /* Max 160MHz */
149      stcSysClkCfg.enExclkDiv = ClkSysclkDiv2; /* Max 84MHz */
150      stcSysClkCfg.enPclk0Div = ClkSysclkDiv1; /* Max 160MHz */
151      stcSysClkCfg.enPclk1Div = ClkSysclkDiv2; /* Max 84MHz */
152      stcSysClkCfg.enPclk2Div = ClkSysclkDiv4; /* Max 60MHz */
153      stcSysClkCfg.enPclk3Div = ClkSysclkDiv4; /* Max 42MHz */
154      stcSysClkCfg.enPclk4Div = ClkSysclkDiv2; /* Max 84MHz */
155      CLK_SysClkConfig(&stcSysClkCfg);
156
157      /* Switch system clock source to MPLL. */
158      /* Use Xtal as MPLL source. */
159      stcXtalCfg.enMode = ClkXtalModeOsc;
160      stcXtalCfg.enDrv = ClkXtalLowDrv;
161      stcXtalCfg.enFastStartup = Enable;
162      CLK_XtalConfig(&stcXtalCfg);
163      CLK_XtalCmd(Enable);
164
165      /* MPLL config. */
166      stcMpllCfg.pllmDiv = 1ul;
167      stcMpllCfg.plln = 50ul;
168      stcMpllCfg.pllpDiv = 4ul;
169      stcMpllCfg.pllqDiv = 4ul;
170      stcMpllCfg.pllrDiv = 4ul;
171      CLK_SetPllSource(ClkPllSrcXTAL); //选择外部高速时钟作为MPLL/UPLL的时钟
172      CLK_MpllConfig(&stcMpllCfg);
173  }
  
```



1.设置MPLL时钟

```
124 static void SysClkIni(void)
125 {
126     en_clk_sys_source_t    enSysClkSrc;
127     stc_clk_sysclk_cfg_t    stcSysClkCfg;
128     stc_clk_xtal_cfg_t      stcXtalCfg;
129     stc_clk_mpll_cfg_t      stcMpllCfg;
130
131     MEM_ZERO_STRUCT(enSysClkSrc);
132     MEM_ZERO_STRUCT(stcSysClkCfg);
133     MEM_ZERO_STRUCT(stcXtalCfg);
134     MEM_ZERO_STRUCT(stcMpllCfg);
135
136     /* Set bus clk div. */
137     stcSysClkCfg.enHclkDiv = ClkSysclkDiv1; // Max 168MHz
138     stcSysClkCfg.enExclkDiv = ClkSysclkDiv2; // Max 84MHz
139     stcSysClkCfg.enPclk0Div = ClkSysclkDiv1; // Max 168MHz
140     stcSysClkCfg.enPclk1Div = ClkSysclkDiv4; // Max 42MHz
141     stcSysClkCfg.enPclk2Div = ClkSysclkDiv4; // Max 60MHz
142     stcSysClkCfg.enPclk3Div = ClkSysclkDiv4; // Max 42MHz
143     stcSysClkCfg.enPclk4Div = ClkSysclkDiv2; // Max 84MHz
144     CLK_SysClkConfig(&stcSysClkCfg);
145
146     /* Switch system clock source to MPLL. */
147     /* Use Xtal as MPLL source. */
148     stcXtalCfg.enMode = ClkXtalModeOsc; //选择外部高速时钟
149
150     /*根据高速时钟的频率设置这个参数
151     ClkXtalHighDrv      = 0u,    ///< High drive capability.20MHz~24MHz.
152     ClkXtalMidDrv       = 1u,    ///< Middle drive capability.16MHz~20MHz.
153     ClkXtalLowDrv       = 2u,    ///< Low drive capability.8MHz~16MHz.
154     ClkXtalTinyDrv      = 3u,    ///< Tiny drive capability.8MHz.
155     */
156     stcXtalCfg.enDrv = ClkXtalLowDrv; //我使用的8M的
157     stcXtalCfg.enFastStartup = Enable; //超高速驱动(禁止的话可以低功耗)
158     CLK_XtalConfig(&stcXtalCfg); //配置
159     CLK_XtalCmd(Enable); //使能
160
161     /* MPLL config. */
162     /*system clk = 168M, pclk1 = 84M, pclk3 = 42M*/
163     stcMpllCfg.pllmDiv = 1u;
164     stcMpllCfg.plln = 42u;
165     stcMpllCfg.FltpDiv = 2u;
```



2.设置进入MPLL的时钟频率和出去的三路时钟的频率


[刷新评论](#) [刷新页面](#) [返回顶部](#)

发表评论

[编辑](#) [预览](#)

B    

支持 Markdown

 自动补全

[提交评论](#) [退出](#)

[Ctrl+Enter快捷键提交]

【推荐】并行超算云面向博客园粉丝推出“免费算力限时申领”特别活动

【推荐】跨平台组态\工控\仿真\CAD 50万行C++源码全开放免费下载！

【推荐】和开发者在一起：华为开发者社区，入驻博客园科技品牌专区



编辑推荐：

- 以终为始：如何让你的开发符合预期
- 五个维度打造研发管理体系
- 不会SQL也能做数据分析？浅谈语义解析领域的机会与挑战
- Spring IoC Container 原理解析
- 前端实现的浏览器端扫码功能

最新新闻：

- 美国版“非升即走”瞄准终身教授，2年评审不通过就减薪撤职（2021-10-21 09:40）
 - 软硬结合：Google借AI增强Pixel 6系列智能机的拍摄体验（2021-10-21 09:30）
 - “宁王”哭穷，时代变了？（2021-10-21 09:25）
 - 上万人受害，涉案金额达6.3亿元...揭秘上海一起网贷“连环套”诈骗案（2021-10-21 09:15）
 - 开源趣闻：AMDGPU驱动调用了Linux内核中的Intel新伙伴分配器代码（2021-10-21 09:08）
- » 更多新闻...

Powered by:

博客园

Copyright © 2021 杨奉武

Powered by .NET 6 on Kubernetes



单片机,物联网,上位机,...

扫一扫二维码, 入群聊。