

玩转 Milk-V Duo

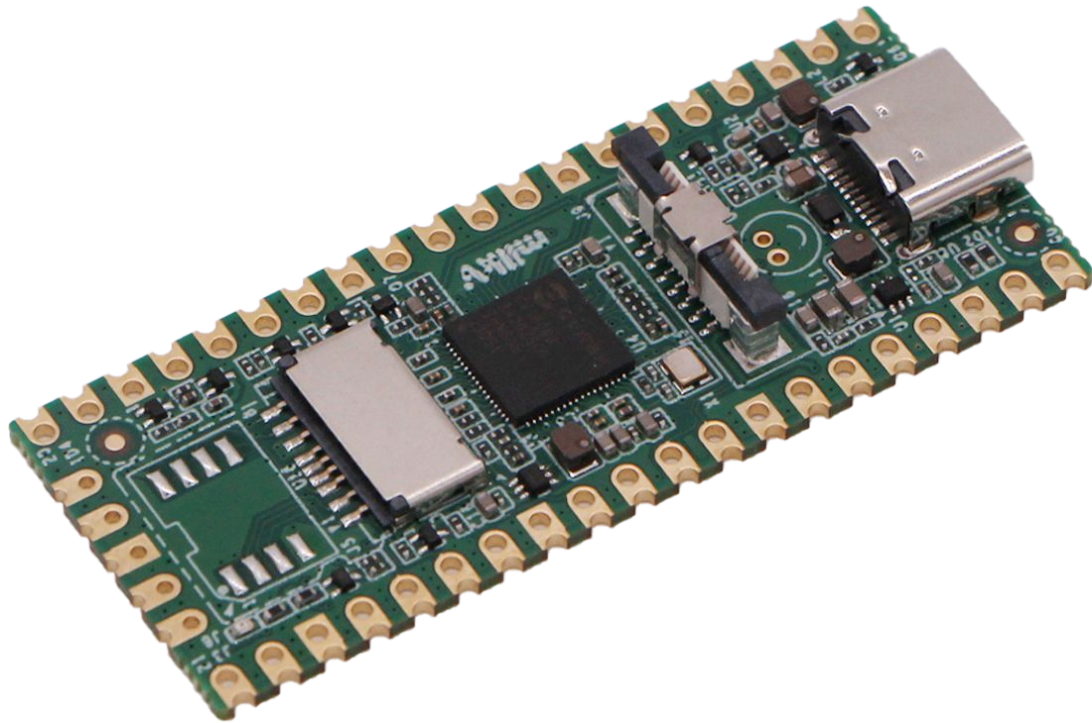
RVOS on Duo

许东

本章内容

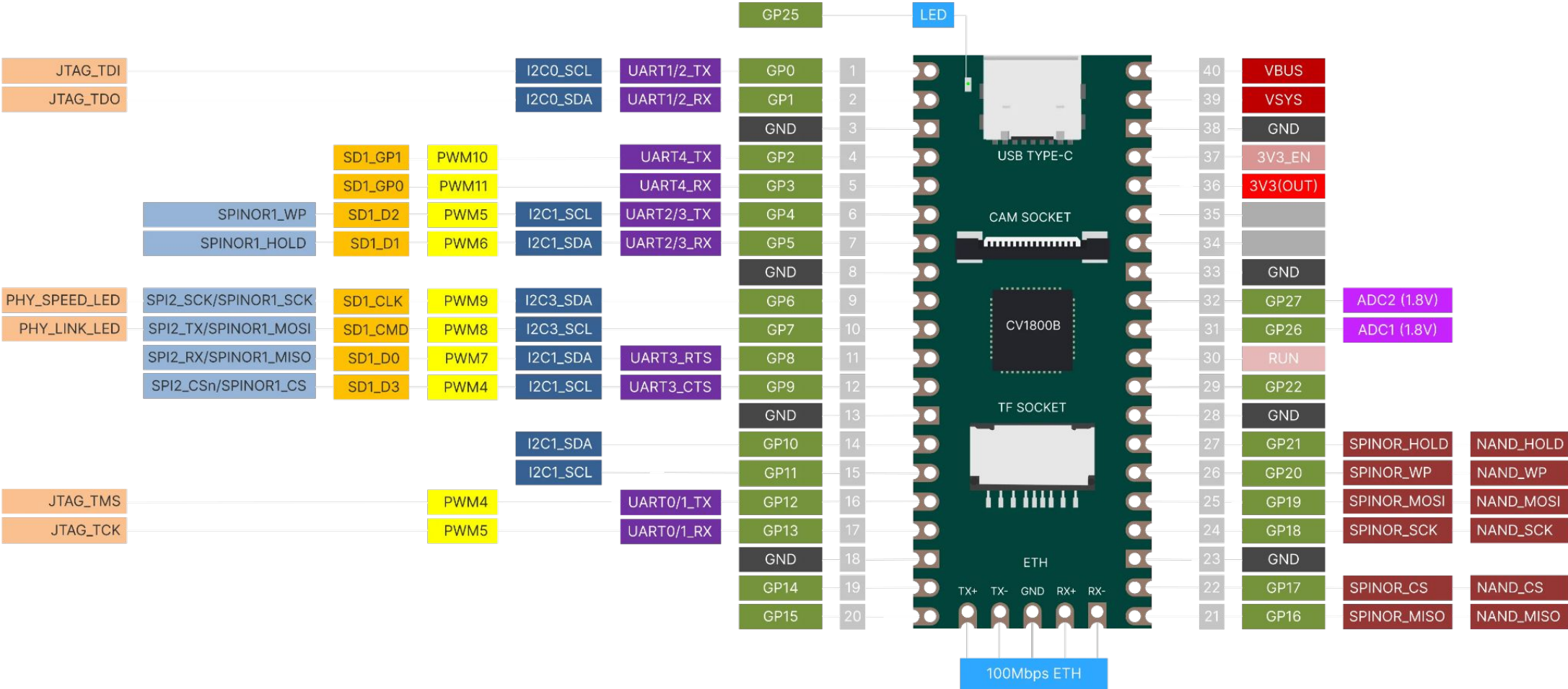
- Duo 介绍
- RVOS 介绍
- 快速移植 RVOS on Duo

MILK-V Duo



Milk-V Duo	规格
处理器	CVITEK CV1800B (C906@1Ghz + C906@700MHz)
内存	DDR2 64MB
Storage	1x Mirco SD slot,1x SD NAND solder pad
USB	1x Type-C for data and Power,1x USB2 solder pad
摄像	1x 16P FPC connector (MIPI CSI 2-lane)
芯片	up to 26 Pins available for general purpose I/O (GPIO)
尺寸	21mm*51mm

MILK-V Duo



RVOS

RVOS(<https://github.com/plctlab/riscv-operating-system-mooc>)是一个用于教学显示设计的操作系统内核, 诞生于 2021 年。

- 设计小巧, 整个核心有效代码只有 1000 行
- 可读性强, 易于维护, 大部分实现为 C 语言, 只有很少部分用到汇编
- 演示了 简单的内存分配管理实现
- 演示了可抢占多线程调度实现, 线程调度采用轮转调度法
- 演示了简单的软件互斥实现
- 演示了软件定时器实现
- 演示了系统调用实现 (M + U 模式)

移植前的准备 & 移植思路

- 移植前的准备

下载 duo-buildroot-sdk , 编译并写入 sd 卡 <https://github.com/milkv-duo/duo-buildroot-sdk.git>

下载 rvos , 开始移植 <https://github.com/plctlab/riscv-operating-system-mooc.git>

下载 duo-toolbox , 通过 toolbox 来生成写入 sd 卡的 fip.bin

<https://github.com/IEAST/duo-toolbox.git>

- 移植思路

架构移植 RISC-V32 RISC-V64

修改 UART, PLIC, CLINT 定义

UART:

基地址：

芯片的 6 组 UART 模块基地址

GPIO 模块	基地址
UART0	0x04140000
UART1	0x04150000
UART2	0x04160000
UART3	0x04170000
UART4	0x041C0000
RTCSYS_UART	0x05022000

芯片的 UART 寄存器概览

cv1800B 芯片手册 P440

IRQ:

MILKV_DUO_UART0_IRQ = 44,

中断号	中断源	中断号	中断源	中断号	中断源
38	SD1 中断	70	RTC GPIO 中断		
39	SPI_NAND 中断	71	RTC UART 中断		
40	I2S0 中断	72	RTC SPI_NOR 中断		
41	I2S1 中断	73	RTC I2C 中断		
42	I2S2 中断	74	RTC WDG 中断		
43	I2S3 中断	75	TPU 中断		
44	UART0 中断	76	TDMA 中断		
45	UART1 中断	77	保留		
46	UART2 中断	78	保留		
47	UART3 中断	79	Timer0 中断		

cv1800B 芯片手册 P104

UART:

- UART 波特率配置

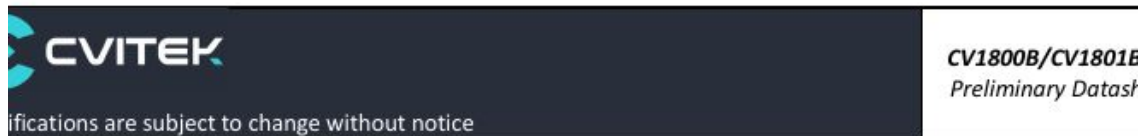
DLL、DLH 为 UART 控制器内部之波特率分频控制寄存器，DLH 为高 8 位、DLL 为低 8 位。配置 DLH、DLL 前须先配置 LCR[7] 为 1。此时可配置寄存器 RBR_THR_DLL (DLL)，IER_DLH (DLH)。

配置完成后，波特率即设定完成，公式为：

$$Baud\ rate = \frac{UART_SCLK}{16 * (256 * DLH + DLL)}$$

- 以 UART SCLK 25MHz 为例，配置 115200 波特率，算式为：

435



$$(256 * DLH + DLL) = \frac{25M}{16 * 115200} = 13.5$$

cv1800B 芯片手册 P435

波特率：DLL = 13.56 ,四舍五入为 14

PLIC:

(https://github.com/milkv-duo/duo-buildroot-sdk/blob/develop/build/boards/default/dts/cv180x_riscv/cv180x_base_riscv.dtsi)

基地址：

```
PLIC0: interrupt-controller@70000000 {
    riscv,ndev = <101>;
    riscv,max-priority = <0x07>;
    reg-names = "control";
    reg = <0x00 0x70000000 0x00 0x4000000>;
    interrupts-extended = <&cpu0_intc 0xffffffff &cpu0_intc 0x09>;
    interrupt-controller;
    compatible = "riscv,PLIC0";
    #interrupt-cells = <0x02>;
    #address-cells = <0x00>;
};
```

Milkv_duo 设备树信息

PLIC:

0x0001000	PLIC_IP0	R/W	0x0	1-31 号中断的中断等待寄存器
0x0001004	PLIC_IP1	R/W	0x0	32-63 号中断的中断等待寄存器
...
0x000107C	PLIC_IP31	R/W	0x0	992- 1023 号中断的中断等待寄存器
Reserved	-	-	-	-
0x0002000	PLIC_H0_MIE0	R/W	0x0	1-31 号中断的机器模式中断使能寄存器
0x0002004	PLIC_H0_MIE1	R/W	0x0	32-63 号中断的机器模式中断使能寄存器
0x0200000	PLIC_H0_MTH	R/W	0x0	机器模式中断阈值寄存器
0x0200004	PLIC_H0_MCLAIM	R/W	0x0	机器模式中断响应/完成寄存器

c906 芯片手册p57

```
#define PLIC_PENDING(id) (PLIC_BASE + 0x1000 + ((id) / 32) * 4)
```

```
#define PLIC_MENABLE(id) (PLIC_BASE + 0x2000 + ((id) / 32) * 4)
```

```
#define PLIC_MTHRESHOLD (PLIC_BASE + 0x200000)
```

```
#define PLIC_MCLAIM (PLIC_BASE + 0x200004)
```

```
#define PLIC_MCOMPLETE (PLIC_BASE + 0x200004)
```

PLIC:

10.5 中断使能寄存器 (PLIC_IE)

每个中断目标对每个中断源均有一个中断使能位（机器模式中断使能和超级用户模式中断使能），可用于使能对应中断。其中机器模式中断使能寄存器用于使能机器模式中断，超级用户模式中断使能寄存器用于使能超级用户模式中断。

对于中断 ID 为 N 的中断，其中断使能信息存储于 PLIC_IE x (x=N/32) 寄存器中的 IE y 上 (y = N mod 32)。其中 ID0 对应的 IE 位固定绑 0。寄存器读写权限参考权限控制寄存器 (PLIC_CTRL) 描述。寄存器位分布和位定义如图 10.3 所示。

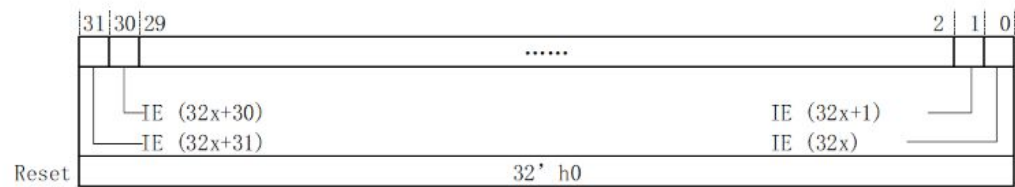


图 10.3: x 号中断使能寄存器 (PLIC_IEx)

c906 芯片手册p57

```
*(uint32_t*)PLIC_MENABLE(UART0_IRQ)=(1 << (UART0_IRQ % 32));
```

CLINT:

(https://github.com/milkv-duo/duo-buildroot-sdk/blob/develop/build/boards/default/dts/cv180x_riscv/cv180x_base_riscv.dtsi)

基地址和时基频率：

```
clint@74000000 {
    interrupts-extended = <&cpu0_intc 0x03 &cpu0_intc 0x07>;
    reg = <0x00 0x74000000 0x00 0x10000>;
    compatible = "riscv,clint0";
    clint,has-no-64bit-mmio;
};

c906_cpus:cpus {
    #address-cells = <0x01>;
    #size-cells = <0x00>;
    timebase-frequency = <25000000>;
};
```

Milkv_duo 设备树信息

CLINT:

表 9.1: CLINT 寄存器存储器映射地址

地址	名称	属性	初始值	描述
0x4000000	MSIP0	读/写	0x00000000	机器模式软件中断配置寄存器 高位绑 0, bit[0] 有效
Reserved	-	-	-	-
0x4004000	MTIMECMPLO	读/写	0xFFFFFFFF	机器模式系统计时器 比较值寄存器 (低 32 位)
0x4004004	MTIMECMPHO	读/写	0xFFFFFFFF	机器模式系统计时器 比较值寄存器 (高 32 位)
Reserved	-	-	-	-
0x400C000	SSIP0	读/写	0x00000000	超级用户模式软件中断配置寄存器 高位绑 0, bit[0] 有效
Reserved	-	-	-	-

c906 芯片手册p51

```
#define CLINT_MTIMECMPL (CLINT_BASE + 0x4000 )
#define CLINT_MTIMECMPH (CLINT_BASE + 0x4004 )
#define CLINT_TIMEBASE_FREQ 25000000
```

谢谢

Group: 玩转 milkv duo 交流群



Valid until 4/7 and will update upon joining group

微信



東東

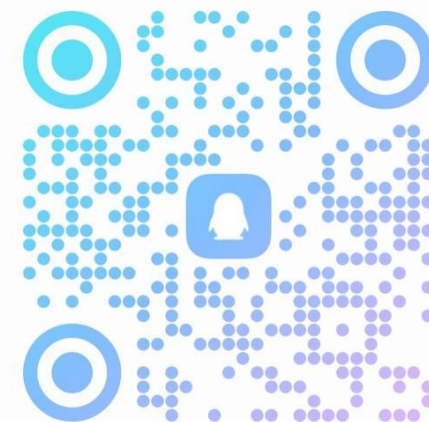


Scan the QR code to add me as friend



玩转 milkv duo 交流群

群号: 534192911



扫一扫二维码，加入群聊

QQ