

# ESM7000 Cortex-M4 硬件资源使用说明

ESM7000 主 CPU 是 NXP 的 i.MX7D 异构多核处理器，其配置了主频 1GHz 的双核 Cortex-A7 和一颗主频 240MHz 的 Cortex-M4 MCU。Cortex-A7 支持 Linux 系统运行，Cortex-M4 用于开发实时应用，支持裸机或 FreeRTOS 程序。

Cortex-M4 硬件资源可根据用户实际使用情况进行定制，本文列出了缺省配置下 ESM7000 Cortex-M4 可用的硬件资源，包括 M4 程序的 CODE、DATA 内存空间缺省分配，M4 可用的 GPIO 和外设等。

## 1. Cortex-M4 程序 CODE 及 DATA 空间配置

ESM7000 Cortex-M4 程序可以加载到 i.MX7D 的片内 TCM、OCRAM 或 ESM7000 的系统 DDR 上运行，在不同配置下支持的 Cortex-M4 程序 CODE 和 DATA 空间如下表所示：

配置	CODE	DATA	A7 & M4 Sheard Memory	CoreMark
TCM	32K (007F8000-007FFFFFFF)	32K (00800000-00807FFF)	64K (00910000-0091FFFF)	626.213
OCARM	128K (00920000-0093FFFF)	64K (007F8000-00807FFF)		450.288
DDR	1MB (80100000-801FFFFFFF)	128K (00920000-0093FFFF)		439.811

表 1:Cortex-M4 程序 CODE 和 DATA 配置

上表中的 CoreMark 是不同配置下的性能测试，可以看到程序在 TCM 中运行时性能最佳，在 DDR 中运行损失了一些的性能，但提供了更大的程序空间。

## 2. Cortex-M4 调试串口

ESM7000 的 ttys2 为 Cortex-M4 程序的调试串口，在 Cortex-M4 程序中调用 PRINTF 将从此串口输出信息。ESM7000 主板 ttys2 缺省配置为 RS232 电平。ttys2 作为 Cortex-M4 调试串口时，通讯格式与 ESM7000 的控制台串口完全一样，默认为 115200-8-N-1。

## 3. Cortex-M4 可用外设

理论上 Cortex-M4 可以使用 ESM7000 主板上的所有硬件资源，但根据实现的使用情况，为了主板硬件资源在 Linux 和 Cortex-M4 上的合理分配，我们对 Cortex-M4 可用的硬件资源做了划分，一个硬件资源不能

在 Linux 和 Cortex-M4 中同时被使用。

ESM7000 的大多数 GPIO 都具有复用功能，在 Cortex-M4 侧使用这些 GPIO 时，有可能只能使用其复用功能，也有可能只能作为 GPIO 使用，也有可能复用功能和 GPIO 功能都能使用，详细说明如下：

ESM7000 主板硬件资源	ESM7000 Cortex-M4 可用硬件资源	
	Cortex-M4 可用的 GPIO	Cortex-M4 可用的外设
GPIO0		
GPIO1		
GPIO2 / ttys5_RXD		
GPIO3 / ttys5_TXD		
GPIO4 / ttys6_RXD		
GPIO5 / ttys6_TXD		
GPIO6 / PWM1		PWM1
GPIO7 / PWM2		
GPIO8 / PWM3		PWM3
GPIO9		
GPIO10 / CAN1_RXD		
GPIO11 / CAN1_TXD		
GPIO12 / CAN2_RXD		
GPIO13 / CAN2_TXD		
GPIO14	GPIO14 <sup>6</sup>	
GPIO15	GPIO15 <sup>6</sup>	
GPIO16 / SD_CLK	GPIO16 <sup>5</sup>	
GPIO17 / SD_CMD	GPIO17 <sup>5</sup>	
GPIO18 / SD_D0	GPIO18 <sup>5</sup>	
GPIO19 / SD_D1	GPIO19 <sup>5</sup>	
GPIO20 / SD_D2	GPIO20 <sup>5</sup>	
GPIO21 / SD_D3	GPIO21 <sup>5</sup>	
GPIO22 / SD_DETn	GPIO22 <sup>5</sup>	
GPIO23 / GPT_Capture	GPIO23 <sup>6</sup>	GPT1
GPIO24		
GPIO25	GPIO25 <sup>5</sup>	
GPIO26 / I2C_SDA		
GPIO27 / I2C_SCL		
GPIO28 / SPI_MISO	GPIO28 <sup>6</sup>	SPI_MISO
GPIO29 / SPI_MOSI	GPIO29 <sup>6</sup>	SPI_MOSI
GPIO30 / SPI_SCLK	GPIO30 <sup>6</sup>	SPI_SCLK
GPIO31 / SPI_CS <sub>n</sub>	GPIO31 <sup>6</sup>	SPI_CS <sub>n</sub>
精简 ISA 总线		EIM/ISA

表 2：ESM7000 Cortex-M4 可用硬件资源

注：

2: 属于 i.MX7D 的 GPIO\_PORT5

5: 属于 i.MX7D 的 GPIO\_PORT6

### 3.1 在 Cortex-M4 上使用 GPIO 的进一步说明

GPIO 在 i.MX7D 处理器上是按组(PORT)操作的, 每组 32 位 IO, ESM7000 引出的 GPIO 来至于 i.MX7D 不同 GPIO\_PORT 的不同 IO。为了避免冲突, 在 Cortex-M4 中使用了某位 GPIO 后, 其所在的这一组 IO 就都不能再在 Linux 中作为 GPIO 使用, 但 IO 对应的复用功能不受此影响, 在 Linux 中仍可正常使用。GPIO 的分组信息如表 2 注释信息所示。

表 2 中列出的 Cortex-M4 可用 GPIO 均支持中断操作。

表 2 已经列出了 ESM7000 Cortex-M4 能用的所有 GPIO, 由于硬件的限制, ESM7000 的其它 GPIO 即使在 Linux 中完全没有使用, 也不能在 Cortex-M4 中作为 GPIO 使用。

## 4. 版本历史

版本	简要描述	日期
V1.0	创建 ESM7000 Cortex-M4 硬件资源使用说明文档	2021-8