# lwip

## 前言

本文对lwip做简要的介绍，希望有助于实际的工程应用

## 简介

lwip是瑞典计算机科学院(SICS)的Adam Dunkels 开发的一个小型开源的TCP/IP协议栈。实现的重点是在保持TCP协议主要功能的基础上减少对RAM的占用。

LwIP是Light Weight (轻型)IP协议，有无操作系统的支持都可以运行。LwIP实现的重点是在保持TCP协议主要功能的基础上减少对RAM的占用，它只需十几KB的RAM和40K左右的ROM就可以运行，这使LwIP协议栈适合在低端的嵌入式系统中使用。

lwIP协议栈主要关注的是怎么样减少内存的使用和代码的大小，这样就可以让lwIP适用于资源有限的小型平台例如嵌入式系统。为了简化处理过程和内存要求，lwIP对API进行了裁减，可以不需要复制一些数据。

LwIP提供了三种编程接口，分别为 RAW/Callback API、Netconn API、Socket API。

利用FSMC写地址，不用手工设置地址线，MCU相应地址线自动设置高低电平选中存储外设的地址

例如

\*(volatile uint16\_t \*)(0x60002468UL)=0xABCD;

数据0xABCD被写入0x1234这个地址

由于内部每个地址对应一个字节,外部设备16位宽,FSMC的一个地址对应两个字节

在STM32中Bank1的256M字节空间由28根地址线(HADDR[27:0])寻址。这里HADDR，是内部AHB地址总线。FSMC写16位宽存储设备时，一次写入16位数据，然而STM32分区的映射地址是按8字节分配的。所以此时HADDR应该向双数地址写入数据，这样把两个字节地址对应外部设备的16位地址。所以写入HADDR地址0x60002468UL，在存储外设则写入0x1234。

## 控制TFT-LCD屏

TFT-LCD屏没有地址，但是有命令数据控制位，数据线通存储类似。

把命令数据控制位接到FSMC的地址线的一条中则可以利用地址控制命令数据选择

例如把把命令数据控制位接到FSMC-A[10]

写地址0X07FE，如上节所讲16位地址线输出0X03FF，即0011 1111 1111，第十位位标红数据位位0，这时命令数据控制位为低电平，状态为读写命令。

写地址0X0800，地址线输出0X0400，即0100 0000 0000，第十位位标红数据位位1，这时命令数据控制位为高电平，状态为读写数据。

typedef struct{

vu16 LCD\_REG; //A10=0

vu16 LCD\_RAM; //A10=1

}LCD\_TypeDef;

// 0，读写命令；1，读写数据

//使用NOR/SRAM的 Bank1.sector4,地址位HADDR[27,26]=11 A10作为数据命令区分线

//注意设置时STM32内部会右移一位对其!

#define LCD\_BASE ((u32)(0x6C000000 | 0x000007FE))

#define LCD ((LCD\_TypeDef \*) LCD\_BASE)

LCD为LCD\_TypeDef类型指针指向地址0x6C000000 | 0x000007FE，则LCD->LCD\_REG为起始地址0x6C000000 | 0x000007FE，LCD\_REG为16位两个字节LCD->LCD\_RAM地址为LCD->LCD\_REG地址+2，即0x6C000000 | 0x00000800。所以读写LCD->LCD\_RAM即为读写命令读写LCD->LCD\_REG即为读写数据。

## 参考资料

版权声明：本文参考了其他CSDN博主的文章，遵循CC 4.0 BY-SA版权协议，现附上原文出处链接及本声明。

1. https://blog.csdn.net/as480133937/article/details/123740365
2. https://blog.csdn.net/weibo1230123/article/details/80210097