与外部SPI扩展模块进行通信

2020.09.04

目标：使用stm32F769ni discovery 与外部SPI扩展mcp23s17模块进行通信

重点：如何对具有SPI接口的外部模块进行软件抽象

上述一至四节，主要以USART子模块为例进行了深入的ST HAL驱动构架分析。本节讨论如何应用ST HAL驱动构架来实现对片外模块的驱动。

本节以MCP23S17芯片为例，有关MCP23S17的详细文档，可参考<https://www.microchip.com.cn/newcommunity/>中对应中英文数据手册。

简单地说，这是个通过SPI总线转换为16个（分为两个8位 bank）可配置IO的芯片。

下图是SPI接口

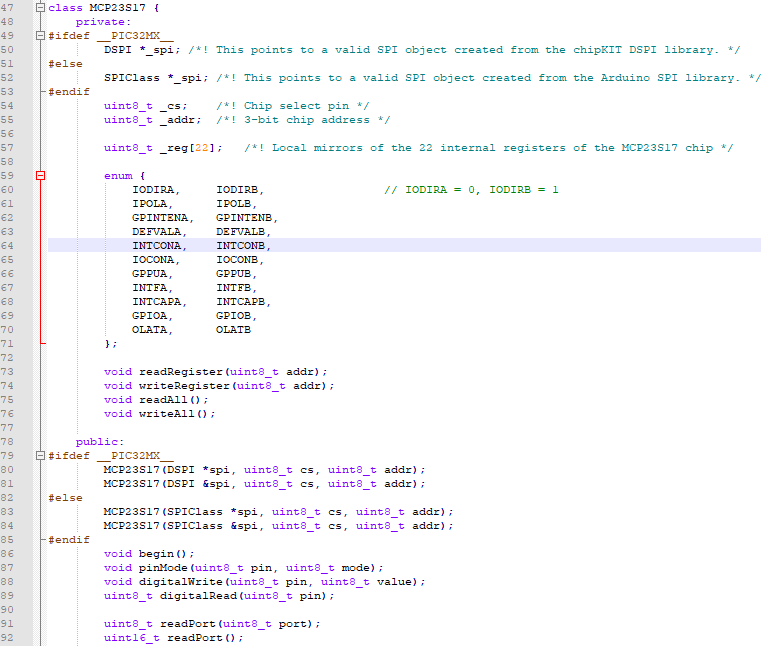
R/W位，0：写；1：读

每储存器长度为一字节8bits，因此，任意一次读写需要3字节24bits。

相比于MCU片内模块，片外模块的储存器无法通过硬件地址内存映射的方法进行寻址，所有的读写操作都必须通过SPI接口。

下面，我们先看一下基于arduino的C++面向对象对MCP23S17的软件封装，这来至于通过github.com检索“MCP23S17 arduino”关键词获得源代码。





比较重要的定义是第52行SPIClass \*\_spi，第57行uint8\_t \_reg[22]和第59-70行。因为片外模块必须通过SPI接口进行操作，因此，MCP23S17软件模块需要包含一个SPI句柄指针。其次，MCP23S17内部储存器数据也在软件模块中存储了一份，以方便根据这些储存器值组织动作函数。最后，由IOCON.BANK=0时，MCP23S17储存器地址连续顺序排列，因此，可以使用第59-70行的枚举变量来表示不同功能的储存器地址。

通过上述分析，可以参考UART\_HandleTypeDef类型定义方法，定义MCP23S17软件模块的类型句柄：



相关的实现这里篇幅有限就不作详细讨论了。