AN-1382
应用笔记

One Technology Way • P.O. Box 9106 • Norwood, MA 02062-9106, U.S.A. • Tel: 781.329.4700 • Fax: 781.461.3113 • www.analog.com

ADP1055 EEPROM编程

作者：Navdeep Singh Dhanjal和Subodh Madiwale

简介

ADP1055提供其设置针对特定应用编程的寄存器图和EEPROM。本应用笔记重点介绍对ADP1055进行编程的软件程序。根据应用环境的不同，有些步骤(比如输入密码和写入板设置)并非必需，可以跳过。

扫描ADP1055

ADP1055可以编程为对64个不同的设备地址进行反应，地址范围为0x40至0x7F。这是通过把ADD引脚上的寄存器值与寄存器0xD0中[5:4]位的值相加得到的。

- 从设备0x40开始，逐步递增，直到0x7F为止。
- 对于每个设备地址，请尝试从寄存器0xD0读取。
- 如果未收到确认，则ADP1055不存在于该设备地址处。
- 如果收到确认且寄存器0xD0中的[7:6]位返回数据 = 1，则ADP1055存在于该设备地址处。
- 当连接的ADP1055的设备地址已知时，请对面向该设备的所有通信均使用这一个地址。

键控代码解锁与数据写准备

要对命令进行写操作，必须解锁设备，并将命令掩码编程为写模式。

更改键控代码密码

当设备由ADI工厂发货时，默认键控代码密码为0xFFFFFFFF。编程结束时，强烈建议把键控代码更改为新键控代码以保护知识产权。

要更改键控代码密码，请把四个字节的旧键控代码写入寄存器0xD7(KEY_CODE命令)，然后把四个字节的新键控代码写入寄存器0xD7。

另外，为了进一步保护知识产权，在工厂出货时，即可写入命令，以掩蔽某些或所有命令，从而保护它们，使其免受意外写和故意读等操作的影响。

设备解锁

如果不需要更改键控代码，则可用默认值0xFFFFFFFF，通过KEY_CODE命令，以块写入方式把四个字节写入寄存器0xD7，以解锁设备。接下来，对寄存器0xD7执行5个字节的块读操作：前四个字节为密码，第五个字节为1。这样可以确保设备解锁。

如果键控代码已更改，则用新的键控代码来解锁设备，其方法与前面描述的，在不更改键控代码条件下解锁设备的方法相同。

数据写准备

要启用对所有命令的读/写操作，对于每个字节，用数据0xFF，以块写入方式把32个字节写入寄存器0xF4的COMMAND_MASK，从而启用对所有PMBus命令的读/写操作。同时，对于每个字节，再用数据0xFF，以块写入方式把21个字节写入EXTCOMMAND_MASK(寄存器0xF5)，从而启用对所有特定制造命令的读/写操作。

EEPROM密码与解锁

当设备由ADI工厂发货时，默认EEPROM密码为0xFF。与键控代码类似，为了保护知识产权，强烈建议在编程结束时更改EEPROM密码。

更改EEPROM密码

要更改EEPROM密码，先把旧密码写入寄存器0xD5，然后把新密码写入寄存器0xD5。

解锁EEPROM

如果不更改EEPROM密码，把0xFF写入寄存器0xD5两次，以解锁EEPROM。

如果EEPROM密码已被更改，则用这个新密码来解锁EEPROM，其方法是把新密码写入命令0xD5两次。要检查EEPROM是否已经解锁，请从命令0xFE94进行一次字读取操作；如果位15 = 1，则EEPROM已经解锁。

保存数据与锁定EEPROM

要把数据保存至EEPROM并锁定EEPROM，请执行下列步骤：

1. 向命令0x15执行一条发送命令，把命令中的数据上传至EEPROM。
2. 等待50 ms，以便上传完成。

3. 要锁定EEPROM，向命令0xD5写入任何数据(密码除外)。

把数据写入设备

1. 把数据写入设备之前，从“.55s”文件读取命令设置，该文件是用ADP1055图形用户界面(GUI)生成的。
2. 运用I²C，对设备的下列命令执行字节写操作：

   0x01至0x02
   0x10
   0x20
   0x41
   0x45
   0x47
   0x49
   0x4C
   0x50
   0x56
   0x5A
   0x5C
   0x63
   0x69
   0xD0
   0xFE01至0xFE0C
   0xFE1D至0xFE2F
   0xFE34至0xFE43
   0xFE48至0xFE57
   0xFE5A至0xFE67

3. 接下来，通过I²C，对设备的下列命令执行字写入操作：

   0x21至0x24
   0x27至0x2A
   0x33
   0x35至0x40
   0x42至0x44
   0x46
   0x48
   0x4A至0x4B
   0x4F
   0x51
   0x55
   0x59
   0x5B
   0x5E至0x62
   0x64至0x66
   0x68
   0xFE0D至0xFE1C
   0xFE30至0xFE33
   0xFE44至0xFE47
   0xFE58至0xFE59

SMBALERT遮蔽命令采用独特的写入方式，如下所示：

1. 对命令0x1B执行8字写入操作。
2. 把.55s文件中针对寄存器0x1B的数据拆分成8个字，每个字必须写入0x1B(更多细节请参阅PMBus技术规格)。
3. 以来自.55s文件的数据，以块写入方式，把32个字节写入命令0xF4。
4. 以来自.55s文件的数据，以块写入方式，把21个字节写入命令0xF5。

PEC控制

如果程序员能够识别确认并且I²C通信中不存在确认，则在所有写操作中均使用PEC字节，确保数据传输正确无误。

PMBus控制器搭载有封装误差检查机制，以提高可靠性与通信鲁棒性。在消息传输结束时附加一个PEC字节，可实现封装误差检查功能。从开始到停止位，用CRC-8算法对所有ADDR、命令和数据字节计算PEC字节，不包括确认(ACK)、无确认(NACK)、开始、重启和停止位。

PEC字节由提供最后数据字节的设备附加于消息末尾。PEC字节的接收者负责计算其内部包误差代码并将其与接收到的PEC字节进行比较。

PMB从机设备可以与支持PEC以及不支持PEC的主机PMBus设备通信。如果PEC字节可用，PMB会检查PEC字节并确认其是否正确。如果PEC字节比较失败，PMB设备不会确认PEC字节，并且不会处理来自主机的命令。

PMB利用CRC-8多项式，通过内建硬件来计算PEC代码

$$C(x) = x8 + x2 + x1 + 1$$

PEC代码一次计算一个字节，计算顺序与接收顺序相同。在读处理中，PMB会附加位于最后数据字节之后的PEC字节。在写处理中，PMB会比较接收到的PEC字节与内部计算得到的PEC代码。

板设置

.55s文件中的板设置用于确保ADP1055 GUI能正常工作。另外，在评估流程中，这些设置存储在EEPROM的第5页中。在独立模式操作中，设备不需要把这些板设置保存到设备中。

要根据板设置中的寄存器值计算命令值，请按照图1中所描述的示例步骤执行。

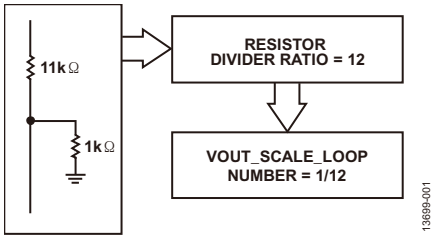

图1. 计算命令值

由于PMBus非常灵活，因此，多个不同的组合可能导致特定组件的最终值相同。例如，在依据命令值计算板设置用到的寄存器值时，如果所用组合太多，结果可能产生相同的数值，如图2所示。

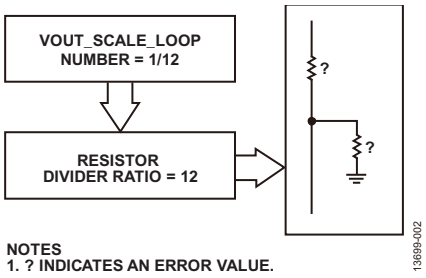

图2. 值相同的组合

如果在生产过程中出现这种情况(如图2所示)，则在连接设备时，GUI会发出通知，称EEPROM的第5页中缺失板设置。然后会发送一条提示消息，从.55s文件加载对应的板设置(见图3)。

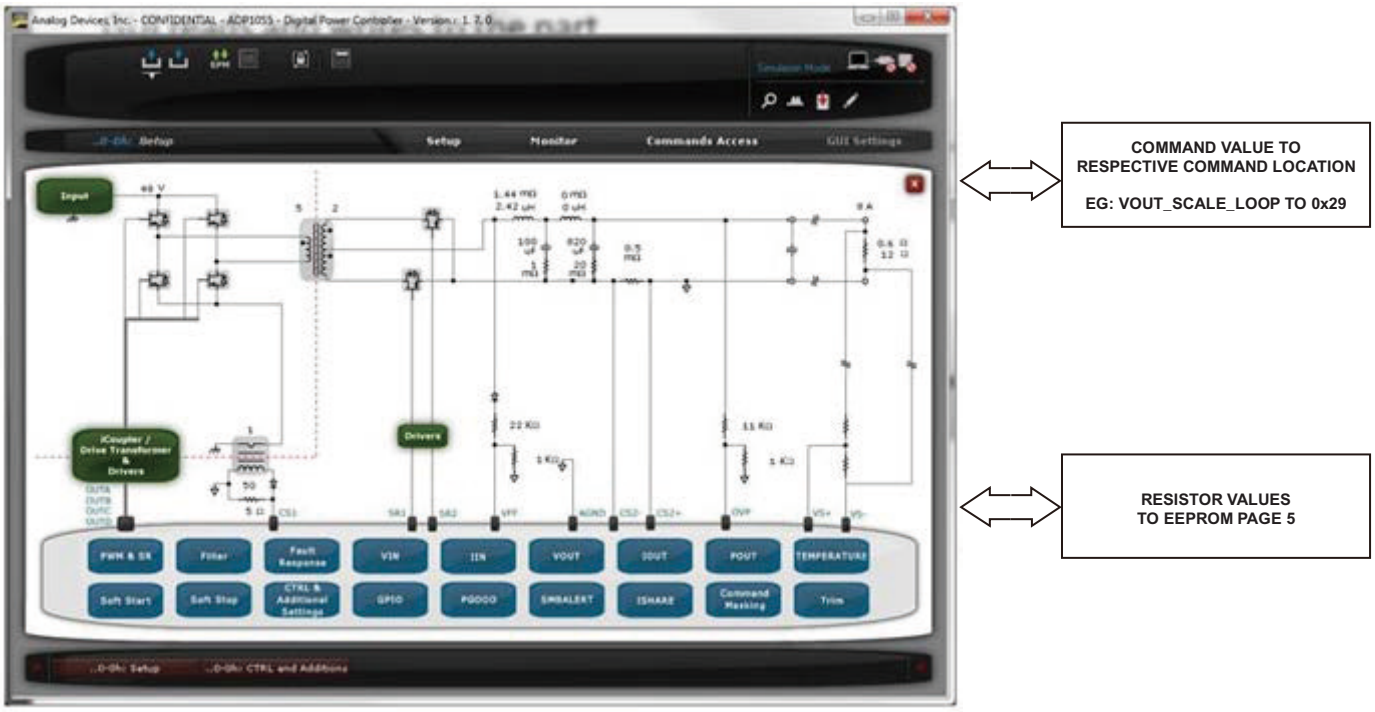

图3. ADP1055 GUI，主设置窗口

AN13699sc-0-10/15(0)

ANALOG DEVICES

www.analog.com