PGS7000通过Modbus操作E703注意事项

|  |  |  |  |
| --- | --- | --- | --- |
| 修改更新记录 | | | |
| V0.1 | 2025年2月10日 | Lilu | 第一版 |
| V0.2 | 2025年3月11日 | Lilu | 完善了性质5. |

0：这是为了给邓工调试上位机初步编制的文档，相对简陋，之后随着项目功能的完善，本文档会更加完善。

1：性质1

默认情况，系统上电会自动导入默认的E703参数，目前导入的是邓工提供的一套参数。

2：性质2

外部可以通过Modbus读取E703所有的参数，包括所有的寄存器数据，也包括所有的CM参数。

（CM参数就是掉电记忆的参数，类似于EEPROM的参数，分为两个区域，一个是原厂的参数，记为CM\_FCT，一个是用户参数，记为CM\_User，芯片只有CM\_User允许修改。）

3：性质3

由于有些CM\_User数据修改后会导致系统异常，所以不再做通过Modbus对任意CM\_User修改功能了。而只允许对个别的几个CM\_User参数修改的功能。

能修改的CM\_User区域的几个参数如下：

|  |  |  |
| --- | --- | --- |
| Modbus Register Address and E703 CM Address Map | | |
| MB\_Addr | CM\_Addr | Note |
| 1296 | 0x20 | CFG\_EN |
| 1298 | 0x24 | CFG\_PADS0 |
| 1301 | 0x2A | CFG\_PERIOD |
| 1304 | 0x30 | CFG\_ADC |
| 1305 | 0x32 | CFG\_LP0 |
| 1306 | 0x34 | CFG\_LP1 |
| 1308 | 0x38 | CFG\_AFE1 |
| 1310 | 0x3C | User data |

（注意1，上面这些参数只是目前设计的允许修改的CM\_User参数，

CM\_User还有其他很多参数，都会保持指定的默认值。

注意2，寄存器数据目前都是可以写的，未做限制。

）

说明：

第一列MB\_Addr：指的是Modbus地址。

第二列CM\_Addr：指的是E703中对应的CM区域地址。

第三列Note：对改参数的阶段说明。

4：性质4

上电是否自动设置为默认参数？

默认上电会导入默认参数，在修改了对应参数后，下次再上电就可能将修改后的参数又改为了默认参数了。所以必须得有一个办法使系统知道参数已经被修改过了，不要再更新为默认参数了。

怎样告诉系统设置为默认参数还是保持旧参数呢？

方法是约定User data参数的意义，如果这个参数为0，则表示没有更新过参数，每次上电都要更新为默认参数，如果修改了指定的参数，应该同时将这个参数修改为1，表示已经修改过参数了，之后上电不要再自动的设置为默认参数了。

5：性质5

怎样修改指定参数并且使它起作用。

要想让对应的参数起作用，必须满足两个条件，1，上电的时候CM参数同步到寄存器2，CM参数校验和必须正确。所以要想使修改后的参数起作用必须告诉系统，让他重新计算一下校验和并且更新校验和，然后在同步一下参数。通知系统的方式是向User data寄存器位置写入指定的数据。对应寄存器的地址和Modbus地址对应关系如下。

|  |  |  |
| --- | --- | --- |
| Modbus Register Address and E703 Register Address Map | | |
| MB\_Addr | Reg\_Addr | Note |
| 1068 | 0x6A | User data |

先向该寄存器写入12后，则系统会重新计算CRC16，并且将其写入到对应位置，这样CRC就很正确了。

然后向寄存器写入11，系统会复位，这样E703的CM的新数据就同步到E703的寄存器了。对应的参数就起作用了。

所以正确的过程是。

Step1：写入对应的新数据。

Step2：向CM\_User\_0x3C写入1。（通知系统，下次不要自动恢复默认参数了。）

Step3：向Reg\_User\_0x6A写入12。（通知系统，重新计算并且更新CRC16，新的CRC16写入到对应的位置。）

Step4：向Reg\_User\_0x6A写入11。（重启系统，这样新的数据就会从NVM更新到各个寄存器里面了。）

Step5：系统重启成功后，就是新参数起作用的状态了。