

## 命令模式

---

CUSTOM [(filename)] [parameters]

Executes a "custom-designed" command which has been provided by special programming using the GHS Programming Interface.

通过 GHS 程序接口，执行一个“用户设计”的命令，该命令由其他特殊程序提供。

## 参数说明

---

filename

The file name of the program containing the implementation of the custom-designed command. This is normally a Dynamic Link Library file with default extension ".DLL".

程序文件名包含了用户设计的命令的执行过程。这通常是一个动态链接库文件，其拓展名一般为 ".DLL"。

parameters

Information to be passed to the custom command. The format and interpretation of this information depends on the design of each particular custom command.

传递到用户命令中的信息。由于每个用户命令各异，所传递信息的格式与注释也不相同。

## Operation

### 操作

---

This command attempts to load and execute a program file which is assumed to be compatible with the GHS system. When the execution of that program is successful and complete, control returns to the GHS command interpreter and the dialog area of the screen is cleared.

这个命令尝试去载入和执行一个可以和 GHS 系统兼容的程序文件。当该程序成功执行完毕，控制将回到 GHS 命令输入区，并且屏幕对话框会清除。

As an example, the command "ADD2" is built in to the standard GHS and is accessed through the CUSTOM command. This presents an interactive screen for entering two-point nonuniform distributed weights. (Since it is built-in, an external ADD2 file is not required.)

例如，命令“ADD2”已在标准的 GHS 中，并且通过 CUSTOM 命令导入。这将呈现一个可操作窗口用来输入两个点的非均布的分布重量。（因为这是内置的，所以不需要外部 ADD2 文件）。

### Custom formats for the Sensor Interface (SI) module

传感器接口(SI)模块的用户格式

The optional SI module uses CUSTOM (GHSERIAL) command formats to communicate with real-time gauging systems by means of supported pre-programmed protocols:

可选模块 SI 使用 CUSTOM (GHSERIAL)命令格式，通过所支持的预先设定的协议，实现与实时测深系统通讯：

CUSTOM (GHSERIAL) START [configfile] [/Log | Dump[:logfile]]

Starts serial communication between GHS and external real-time systems. Communication is configured according to the specified configfile (or GHSERIAL.DAT by default). If the /LOG or /DUMP parameter is present, then each received serial message is logged to logfile (default GHSERIAL.LOG; formatted for /LOG or dumped raw for /DUMP).

GHS 与外部实时系统之间开始进行串行通信。通信根据指定的配置文件 configfile 设定（默认为 GHSERIAL.DAT）。如果出现参数/LOG 或/DUMP，则每个接收信息都被录入到文件 logfile（默认是 GHSERIAL.LOG；/LOG 为格式编排后的数据，/DUMP 为原始数据）。

The configuration file is composed of serial parameter lines beginning with the "\$" character, followed by an indexed list of data items from the serial stream. Serial parameter formats are:

配置文件由一系列参数组成，以字符"\$"开头，其后为串行数据流的索引。系列参数格式如下：

\$TYPE protocol - the name of a supported serial protocol type (required);

\$TYPE protocol - 支持的串行协议类型名称（必须）；

\$PORT name - serial port name (default=COM1; others may vary by \$TYPE);

\$PORT name - 串行端口名称（default=COM1；其他会因为\$TYPE 而不同）；

\$BAUD rate - baud rate in bits per second;

\$BAUD rate - 波特率，单位：位/秒；

\$PARITY method - parity (0=none, 1=odd, 2=even);

\$PARITY method - 奇偶校验（0=空, 1=奇, 2=偶）；

\$DATABITS bits - number of data bits per byte;

\$DATABITS bits - 每字节数据位数；

\$STOPBITS bits - number of stop bits (1, 1.5, or 2);

\$STOPBITS bits - 结束位数(1, 1.5, or 2)；

\$IDFIELD field - ID field position index in each record (default=1);

\$IDFIELD field - 每次记录的位置 ID 索引（默认=1）；

\$IDLEN chars - ID field character length;

\$IDLEN chars - 编码字符长度；

\$DEFFIELD field - default data field position index (default=2);

\$DEFFIELD field - 默认数据区位置索引（默认=2）；

\$\$SCALEn factor - scale factor for nth field data (default=none);

\$\$SCALEn factor - 第 n 项数据比例因子（默认=无）；

\$\$SLAVE address - responding slave (def 0=broadcast; <0=-master's slave);

\$\$SLAVE address - 反应装置（默认 0=传播；<0=-主控装置）；

\$\$INMIN id - minimum record ID queried every timeout if master (def=all);

\$\$INMIN id - 在时间段中记录最小的 ID（默认=全部）；

`$INMAX id` - maximum record ID queried every timeout if master (def=all);  
`$INMAX id` - 在时间段中记录最大的 ID (默认=全部)

`$READPLUSn offset` - record ID offset for nth extra READ variable (def=0);  
`$READPLUSn offset` - 记录第 n 个额外 READ 变量的 ID 变址 (默认=0);

`$WRITEPLUS offset` - record ID offset for WRITE variable (default=0);  
`$WRITEPLUS offset` - 记录 WRITE 变量的 ID 变址;

`$WRITEDIV[n] factor` - record ID divisor for WRITE variable (default=1);  
`$WRITEDIV` - 记录 WRITE 变量的 ID 除数 (因子) (默认=1);

`$PLUSFIELDn field` - field position for nth extra READ (or WRITE if n=0);  
`$PLUSFIELDn field` - 第 n 次额外读取 (n=0 时为写入) 位置;

`$MODPLUS offset` - Modbus ID offset (1 for 1-based register IDs; default=0);  
`$MODPLUS offset` - Modbus ID: 1-制定的 ID, 默认为 0;

`$DATAPLUS n` - extra records per ID increasing data precision (default=0);  
`$DATAPLUS n` - 为了提高精度, 对 ID 的额外记录, 默认为 0;

`$FLOATING n` - floating-point binary data if n=1 (default=integer);  
`$FLOATING n` - 如果 n=1 浮点二进制数据 (默认值=整数);

`$LITTLEEND n` - little-endian binary data if n=1 (default=big-endian);  
`$LITTLEEND n` - 如果 n=1, 则小端二进制数据 (默认值=大端);

`$PREFIX str` - prefix applied to record IDs from data items (default=none);  
`$PREFIX str` - 给记录数据 ID 加前缀 (默认无);

`$SUFFIX str` - suffix applied to record IDs from data items (default=none);  
`$SUFFIX str` - 给记录数据 ID 加后缀 (默认无);

`$STARTCHAR value` - start of packet character ASCII value;  
`$STARTCHAR value` - 开始数据包的字符的 ASCII 值;

`$SEPCHARS value,...` - list of field separator ASCII values;  
`$SEPCHARS value,...` - 列出区域分离符的 ASCII 值;

`$MAXGAP seconds` - maximum interval gap between characters in a packet;  
`$MAXGAP seconds` - 数据包中字符间的最大间隔时间;

`$MAXIDLE seconds` - maximum idle time before clearing unfinished packet;  
`$MAXIDLE seconds` - 清除未完成包之前的最大停歇时间;

`$TIMASTER seconds` - wait time between data requests if master (def=10);  
`$TIMASTER seconds` - 主程序需求数据的等待时间 (默认=10 秒)。

`$TIMEOUT seconds` - timeout for reading or sending packets.  
`$TIMEOUT seconds` - 读取或发送的超时设定。

Each line in the indexed list of data items is composed of a record ID string (which may be surrounded by quotes), optionally followed by a data field position index (default set by `$DEFFIELD`). For example, a line containing "TANK6P 3" indicates data from the

3rd field of the record marked with ID="TANK6P". If the ID string is empty (e.g. ", 3"), then the indexed field is taken from any common or system data not associated with particular records. The format "readID>writeID" (e.g. "3>103") may be used to specify a separate write ID.

数据项目的每一行索引都由一个记录字符串 ID 组成（可由引号引出），其后还能可选加上一个数据区域位置索引（默认设置为\$DEFFIELD）。例如，"TANK6P 3"表示来自的第三数据区域，ID 为"TANK6P"的记录。如果 ID 为空（即：", 3"），那么索引区域将取自任何非特殊记录的普通或系统数据。

Supported \$TYPEs include read/write types "TextFile", "Modbus" (serial), "Modbus TCP" (Ethernet), and "VIGO", along with read-only data file type "Bergan", serial master/slave type "NCA DL 94", and serial broadcast types "AN-PRO3", "Datasound 53", "Musasino", "NMEA 0183", and "Sentry IV".

支持的\$TYPEs 的读/写类型的通讯协议有：“TextFile”、“Modbus”（串行）、“Modbus TCP”（以太网）和“VIGO”，以及只读数据文件类型的“Bergan”、串行从属类型“NCA DL 94”和串行广播类型“AN-PRO3”、“Datasound 53”、“Musasino”、“NMEA 0183”和“Sentry IV”。

Configuring "\$TYPE TEXTFILE" uses the text file specified by \$PORT instead of a serial port for communications. Fields on each record line are Comma-Separated Values (CSV) unless \$SEPCHARS are specified. Consecutive separators are treated as a single separator unless repeated in the \$SEPCHARS list. Spaces or tabs are trimmed around fields without acting as separators if listed after any others in \$SEPCHARS. Surrounding quotes are removed around fields, and internal quotes should be doubled. The text file is checked every 10 seconds or as specified by \$TIMEOUT. Any WRITE values are sent to a parallel output text file specified by \$PORT with "+" appended to the file name before the extension (e.g. "FILE+.TXT").

设定"\$TYPE TEXTFILE"时将使用\$PORT 所指定的文本文件，替代了串行端口。在没有定义\$SEPCHARS 的情况下，每行记录区域都是逗号分隔 (CSV)。除非分隔符在\$SEPCHARS 列表里重复，连续分隔符和一个分隔符功能相同。如果列在\$SEPCHARS 中其他字符之后，空格和跳格不会有分隔功能。区域外围的引号将会被删除，内部若有引号需重复打引号。如果没在\$TIMEOUT 里另外定义，文本文档将每 10 秒检查一次。任何 WRITE 值都会发送到由扩展名前附加“+”的\$PORT 指定的并行输出文本文件（例如“FILE+.TXT”）。

Multiple simultaneous connections can be started, with each configured by its own protocol \$TYPE and the parameter lines that follow it. Data item lines are associated with the previous \$TYPE but are consecutively indexed across connections; for example, if the first \$TYPE has 10 data lines starting at index 1, then the next \$TYPE starts at index 11. If the /LOG or /DUMP parameter is present, then separate log files are written, appending "2" to the log filename for the second connection, "3" for the third, etc.

可以同时启动多个连接，每个连接由其自己的协议\$TYPE 及其后面的参数行进行配置。数据项行与前一个\$TYPE 相关联，但跨连接连续编制索引；例如，如果第一个\$TYPE 有 10 个数据行从索引 1 开始，则下一个\$TYPE 从索引 11 开始。如果存在 /LOG 或 /DUMP 参数，则会写入单

独的日志文件，将“2”附加到第二个连接的日志文件名中，将“3”附加到第三个连接的日志文件名中，依此类推。

CUSTOM (GHSERIAL) READ listindex varname [extra1]...

Reads the current value of the indexed item from the configured data list into the named variable. For example, "CUSTOM (GHSERIAL) READ 5 X" sets the existing X variable equal to the data item indicated by the 5th data line listed in GHSERIAL.DAT. If listindex is zero, then varname is loaded with the number of seconds since last valid data was received (positive if any changed data received since READ 0 was last done, negative if not, or zero if no valid data received since communication started). If extra variable names are present, each extran variable is read from the item configured for the indexed record ID added to the corresponding \$READPLUSn offset then divided by any \$READDIVn factor.

将当前配置数据中指定位置 listindex 的数据，赋值给指定变量 varname。例如，"CUSTOM (GHSERIAL) READ 5 X"设定变量 X 的值为 GHSERIAL.DAT 中第五行所指数据。如果 listindex 为 0，那么 varname 将记入上次接收有效数据所用的秒数（如果最后执行 READ 0 后接收到了更改数据，则为正值，反之为负；如果连通开始后没有接收任何有效数据，则为 0）。如果出现额外的变量名，每个额外变量都是配置数据中 ID 的变址，用于添加到 \$READPLUSn 中。

CUSTOM (GHSERIAL) WRITE listindex value

Writes the specified value to the indexed item from the configured data list, marked for later transmission when using protocols that support bidirectional communication. If no separate write ID was specified using "readID>writeID" format, then the indexed record ID is added to any \$WRITEPLUS offset and divided by any \$WRITEDIV factor to determine the ID to write and transmit. For Modbus protocol type, \$WRITEPLUS can be -10000 to convert holding register (relative to 40001) to input register addresses (relative to 30001). If extra values are present, each extran value is written to the item for the indexed record ID added to the \$WRITEPLUSn offset divided by any \$WRITEDIVn factor.

将指定数据 value 写入到配置数据的指定位置 listindex 中，供支持双向通信协议之后传送。如果未使用“readID>writeID”格式指定单独的写入 ID，则将索引记录 ID 添加到任何 \$WRITEPLUS 偏移量并除以任何 \$WRITEDIV 因子以确定要写入和传输的 ID。对于 Modbus 形式，默认 \$WRITEPLUS 是 -10000 来转换从保存寄存器（对应地址为 40001）到输入寄存器的 ID（对应地址为 30001）。如果出现额外的数值，每个额外数值都是配置数据中的 ID，用于添加到由 \$WRITEDIVn 分布系数的 \$WRITEPLUSn 偏移中。

Clears the current values of all data items.

清除所有数据项目的当前值。

CUSTOM (GHSERIAL) STOP [/Keep]

Stops serial communication. If /KEEP is present, subsequent CUSTOM (GHSERIAL) READ commands can still be used to fetch the final values of data items when communication terminated.

停止串行通信。如果出现 /KEEP，之后的命令 CUSTOM (GHSERIAL) READ 在传输终止后仍然能够获得最终数据。

## CUSTOM (GHSERIAL) TEST [testfile]

Loads test data from the specified testfile (or GHSERIAL.TST by default), simulated as if received via serial communication.

从指定的文件 testfile 载入文字数据（默认为 GHSERIAL.TST），可以模拟似从串行通信中接收到的数据。

## Output

### 输出

---

Any output would be from the custom-designed program.

所有输出都能来自用户设计程序。

## Examples

### 样例

---

Executing a custom command implemented in the file NEWSTAT.DLL and passing to it the parameter string "PROFILE RA":

执行在文档 NEWSTAT.DLL 中的用户命令，并且传送参数"PROFILE RA":

```
CUSTOM (NEWSTAT) PROFILE RA
```

Accessing the built-in command for entering weight items:

访问内部指令从而输入重量信息:

```
CUSTOM (ADD2)
```

Fetching the value of the second configured data item after ending serial communication:

终止串行通信后，获取第二位配置的数据值:

```
CUSTOM (GHSERIAL) START
```

```
WAIT 10
```

```
CUSTOM (GHSERIAL) STOP /KEEP
```

```
VARIABLE TANKLEVEL
```

```
CUSTOM (GHSERIAL) READ 2 TANKLEVEL
```