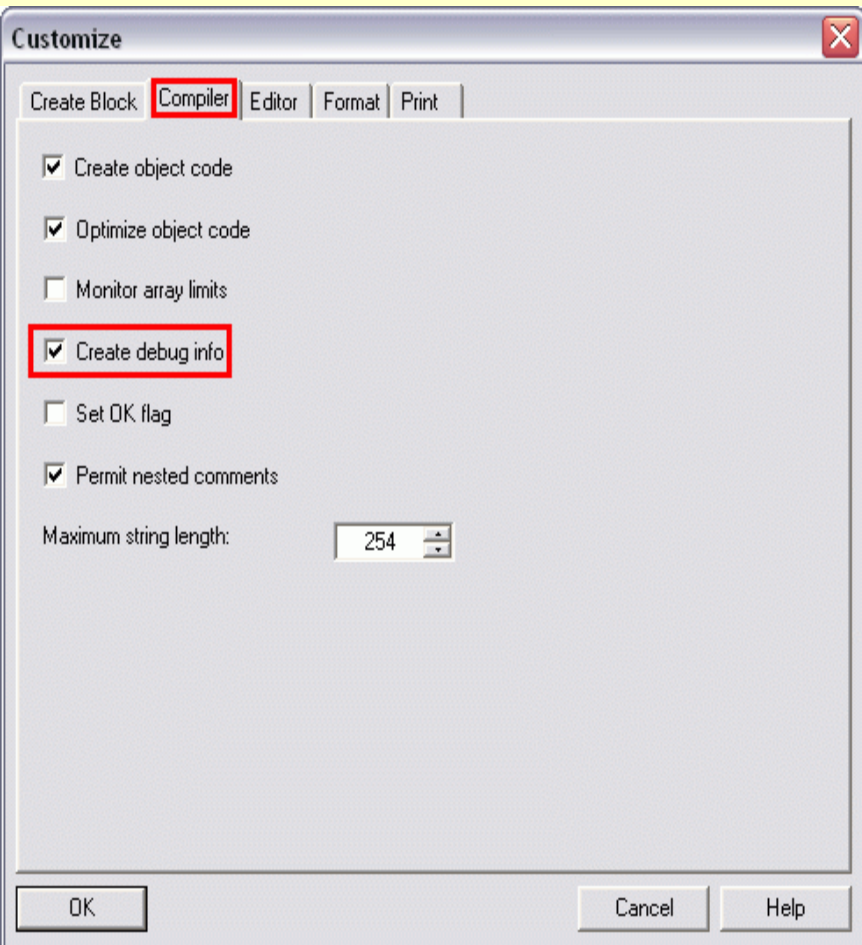


**描述:**

调试器帮助对 S7-SCL 源代码进行一个简单调试，如果在编译 S7-SCL 源代码过程没有错误发生，能够通过利用断点找出程序逻辑错误，并在运行时监视 S7-SCL 代码的函数功能。过程如下面表格中描述:

1.	<b>过程:</b>
1	<p>在 S7-SCL 编辑器中通过"Tools &gt; Settings..." 打开"Settings"对话框，选择"Compiler"选项，激活创建诊断信息项，确认并退出。</p>  <p>图 01</p> <p><b>注意:</b> 在这种情况下，测试所必需的参考数据在编译过程自动创建，这增加了程序内存的要求，同时自动化系统运行时间也会加长。</p>
2	<p>通过"Debug &gt; Operation &gt; Test operation"，改变 CPU 在测试模式下。</p> <p><b>注意:</b> 在这种模式下，所有的调试功能通过 PG/PC 可以不受限制地使用，这大大加长了 CPU 的循环周期。如果 CPU 是在测试模式，必须确认 CPU 或者生产能够允许循环周期的变长。例</p>

- 如，在这种情况下,由于通过程序控制的语句状态在每个周期都被要求得到，导致 CPU 的循环周期变长。
- 3 通过 "File > Save"或者工具栏保存按钮保存程序。
  - 4 通过"File > Compile" 或者通过工具栏编译按钮编译程序。
  - 5 通过"Target system > Load" 或者通过工具栏下载按钮下载程序到 CPU 中。
  - 6 放光标在要调试的程序行位置，通过菜单"Debug > Set breakpoint"。在 S7-SCL 语句旁边，断点显示为一个有颜色的小圆点。
  - 7 如果对一个 FB 功能块有多个实例，必须对特定实例定义特定功能，即处理前需要定义相应的背景 DB 块。

通过"Debug > Call environment blocks...", 在下图 02 中选择背景 DB 块并确认。

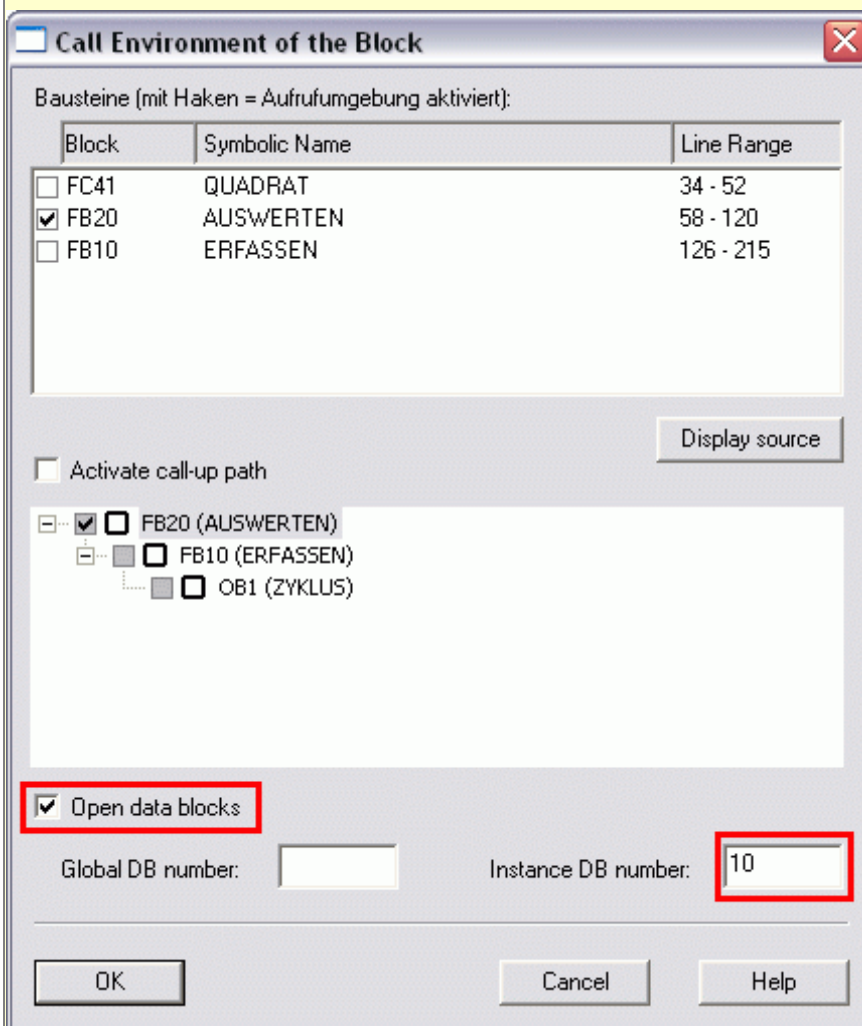


图 02

这种情况下，此背景块对应的功能块被调用时，背景块也开始执行。

- 8 通过"Debug > Edit breakpoints..." 选项，点击打开数据块，选择相应 DB，如图 02 所示并确认。

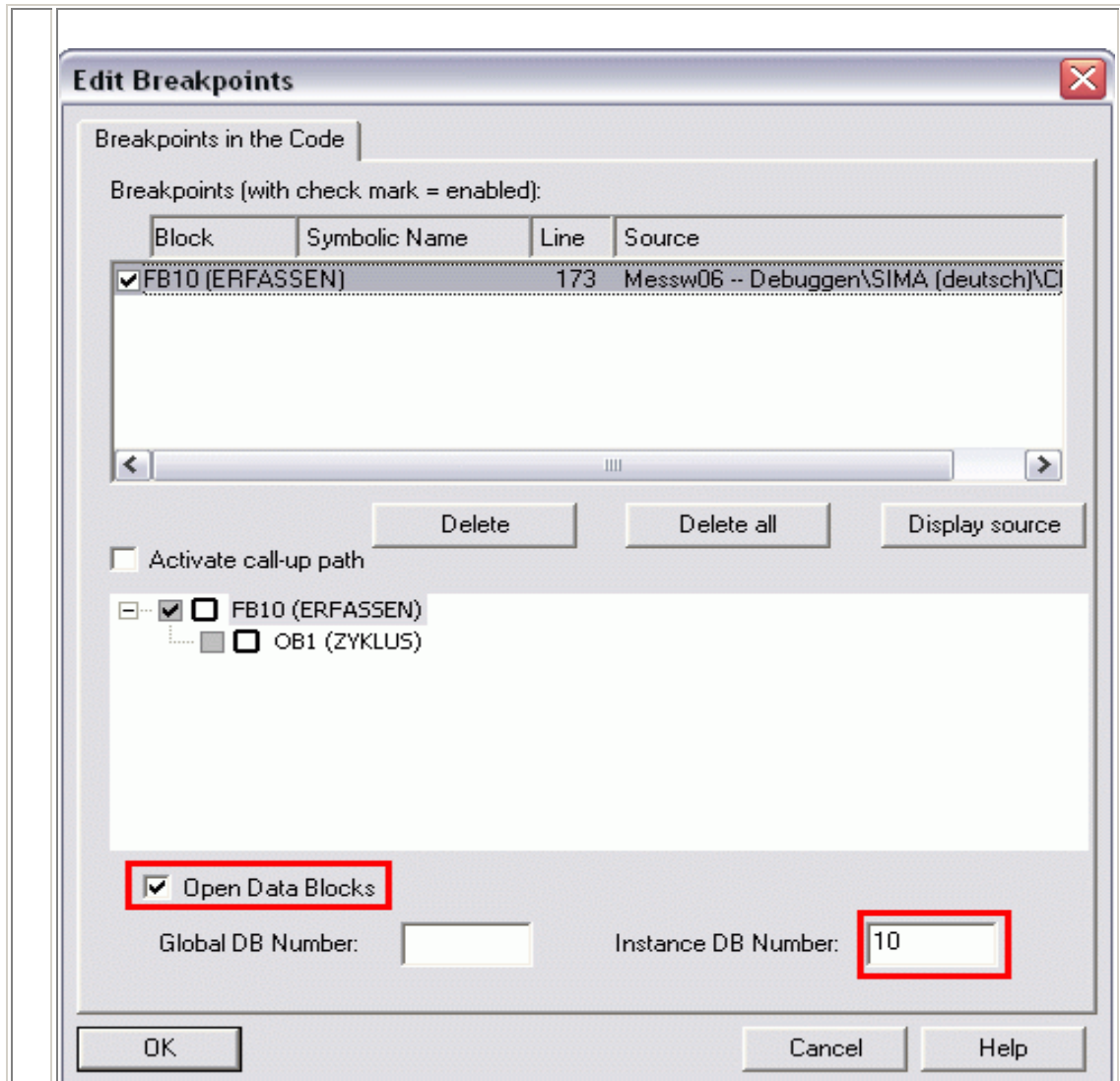


图 03

带有背景数据的功能块被调用时，上面的设置激活断点。

- 9 通过 "Tools > Settings..." 选择"Format"项可以个性化调试窗口，能够改变字体以及调试窗口颜色，并能够为变量在状态改变及特定变量定义不同颜色。

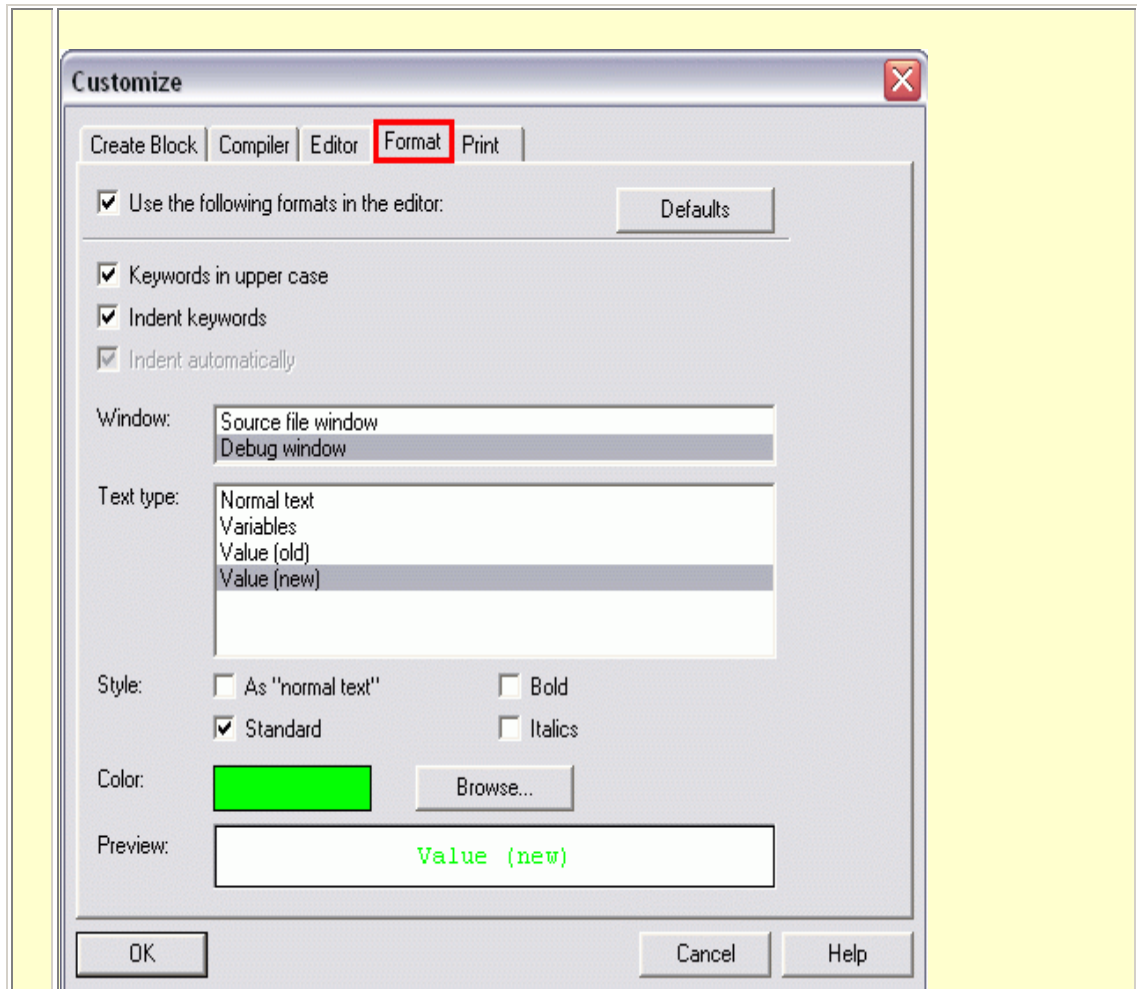


图 04

- 10 通过菜单"Debug > Breakpoints active".激活设置的断点。
- 11 如图 05, 现在可以测试 S7-SCL 程序代码, 程序执行到断点后位置。

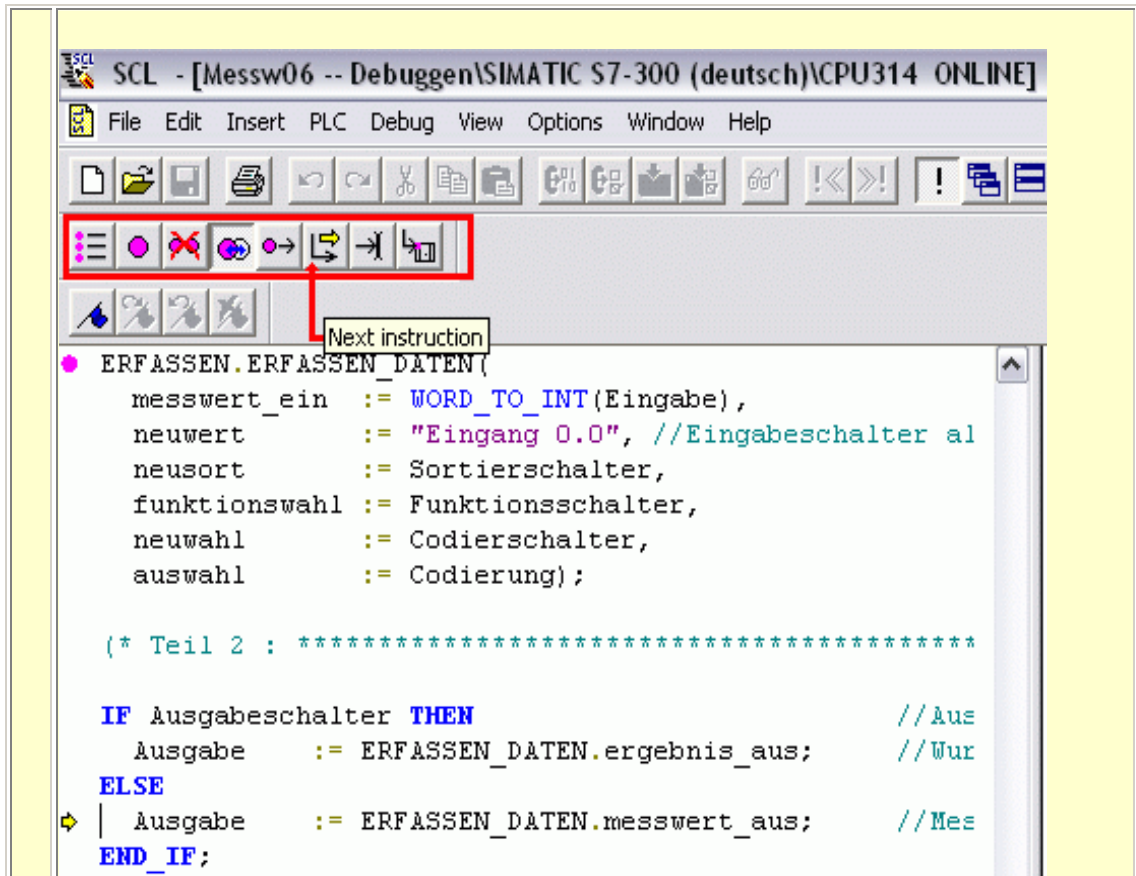


图 05

12 调试器通过执行下面操作退出程序调试。

1. "Debug > Delete all breakpoints",
2. "Debug > Close debug".

**注意:**

重新将 CPU 的操作模式设置到生产模式并取消选项"Create debug info"(通过"Tools > Settings... > Tab: Compiler"), 这降低了内存要求并提高了程序块的执行时间, 同时必须重新编译程序并下载到 CPU 中。

能够通过下面的内容找到更多关于 S7-SCL 的信息

- 关于断点/单步模式测试信息
- 通过断点多步调试
- 单步模式的调试

**关键字:**

故障处理

条目号:30335737 日期:2008-08-27