

仿真相关操作步骤

新定义芯片具有仿真功能，在进行仿真之前需要先安装 RD_KEIL_Setup 插件，该插件可以在新定义官网（www.rdsmcu.com）下载，安装时 keil 插件会自动检索 keil 编译器安装目录，将插件安装在 keil 对应目录下；新定义的仿真工具是 RD LINK PRO，支持所有芯片的仿真。

本文档主要包含以下内容，客户可以根据需求自行查阅：

一、keil 仿真设置

二、仿真操作

- 1、设置/删除断点
- 2、进入/退出仿真
- 3、全速运行至断点(Run)
- 4、单步跟踪(Step)
- 5、跨步运行(Step Over)
- 6、运行至光标处(Run to Cursor Line)
- 7、复位(Reset)

三、查看和修改变量

- 1、使用 Watch 窗口查看和修改变量
- 2、使用 Memory 查看和修改变量
- 3、查看和修改 SFR

一、keil 仿真设置

- 1、 打开 Keil 工程文件, 点击快捷图标“Target Option”, 在“Target Option”界面中选择“Debug”, 点选“Use”, 并从下列中选取“RD 8051 Driver”, 同时, 勾选“Run to main()”;

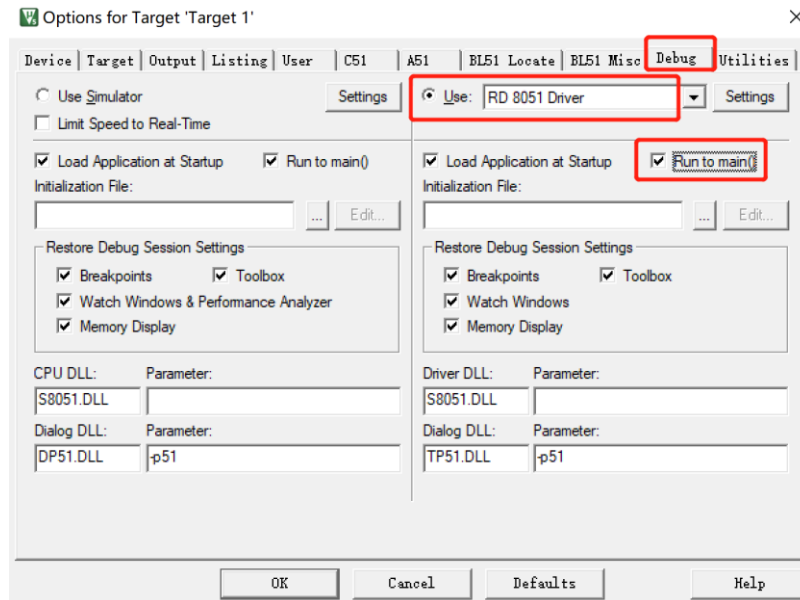


图 1.1

- 2、 点击“Utilities”, 在“Use Target Driver for Flash Programming”中选择“RD 8051 Driver”, 在点击“Settings”, 进入“烧录 Option 信息界面”, 如下图:

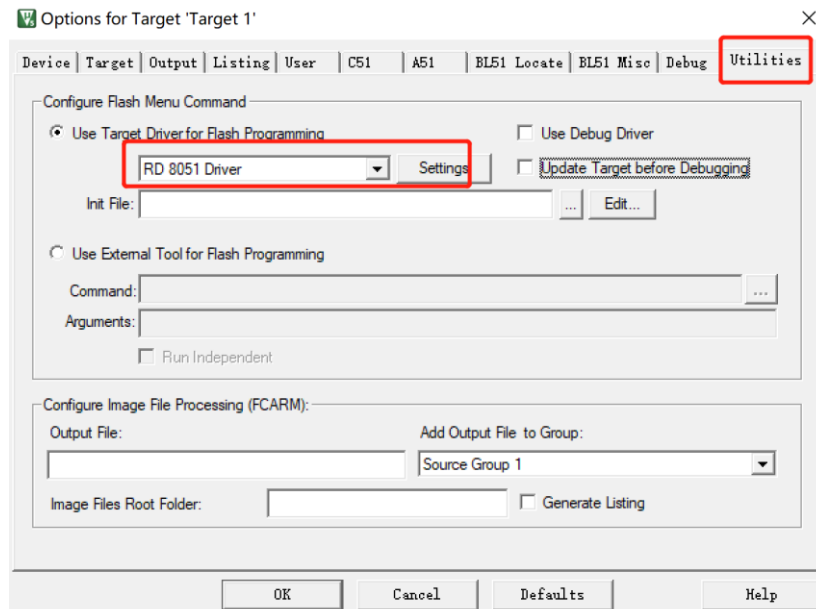


图 1.2

- 3、 再点击“Settings”, 进入“烧录 Option 信息界面”, 如下图:

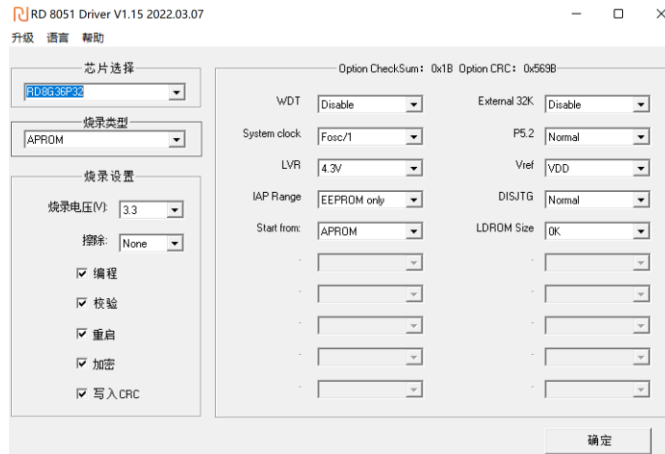


图 1.3

4、烧录 Option 信息配置：

- a) 芯片选择：选择预烧录或仿真的 IC 名称；
- b) 烧录设置：包括擦除，编程，校验，重启，加密以及写入 CRC；
- c) 烧录选项：根据程序设计进行相应的设置；
- d) 升级：用于库文件的升级；
- e) 帮助：版本的相关信息；

注意：擦除选项有 **None**、**Sector**、**All**，其中 **None** 为不擦除，**Sector** 为只擦除程序区，**All** 为全区域擦除，芯片在烧录前必须要进行擦除，否则可能会出现数据烧录错误导致校验失败的情况。

5、点击 keil 软件中的“编译”按钮，编译通过后，点击“LOAD”按钮将程序下载到芯片中，然后点击“debug”按钮，进入仿真模式，如图：1 为编译；2 为 LOAD 键；3 为进入仿真按钮。

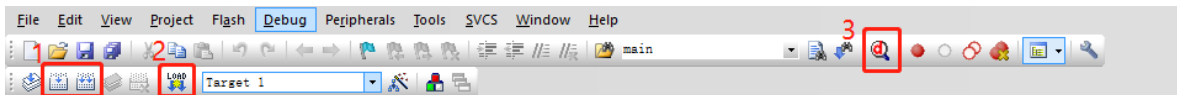


图 1.5

注意：如果芯片选择中未查找到需要的 IC 型号，需要点击升级，进行 MCU 库升级；

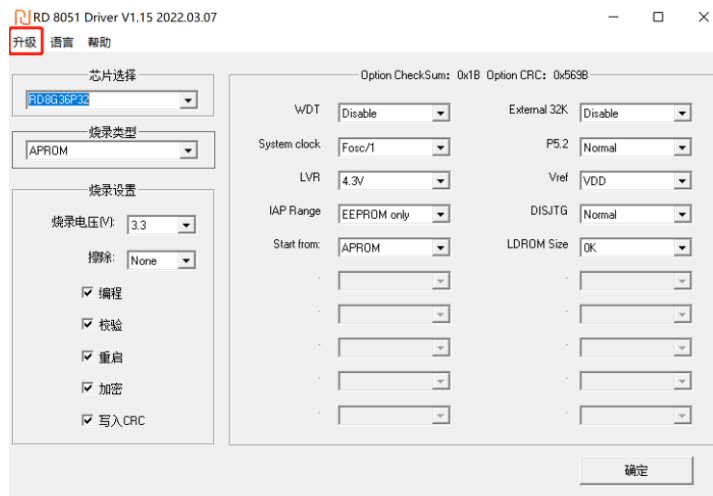


图 1.5

二、仿真操作

1、设置/删除断点

断点设置：在预设断点的源码程序行双击鼠标左键，或者按下“F9”按键，或者点击快捷图标“Insert/Remove breakpoint(“Debug”按钮的右侧)”，当该行左侧会出现红色块标记出现即设置成功。

断点取消：在预取消的断点前，双击鼠标左键，或者按下“F9”按键，或者点击快捷图标“Insert/Remove breakpoint(“Debug”按钮的右侧)”，当该行左侧会出现红色块标记消失即设置成功。

要求：进入仿真前，可以预设好断点，也可进入仿真后设置/删除断点，但在仿真运行过程中不可设置/删除断点，需要将程序停下来之后再设置/删除断点。

2、进入/退出仿真

当程序 Download 通过，点击快捷图标“start/stop Debug Session”按钮，或按下快捷键“Ctrl+F5”或者使用菜单“Debug>start/stop Debug Session”如下图所示。当正确进入时，“D”按钮为凹陷状态，表示打开；再次点击，退出调试，“D”按钮为平滑状态，表示关闭。进入仿真调试状态，工具栏会增加调试相关的菜单，下图从左向右依次是复位、运行、停止、单步、跨步、跳出当前函数、运行至光标处、下一状态、命令窗口、反汇编窗口、符号窗口、系统寄存器窗口、调用堆栈窗口、观察窗口、内存窗口、串口窗口等。上述工具栏的菜单在当前的“Debug”菜单栏中均可找到。



图 2.1

3、全速运行至断点(Run)

由于在进入仿真状态前，已经预先设置好了断点，点击快捷图标“Run”按钮或者“F5”，程序全速运行遇到断点停止，如下图：

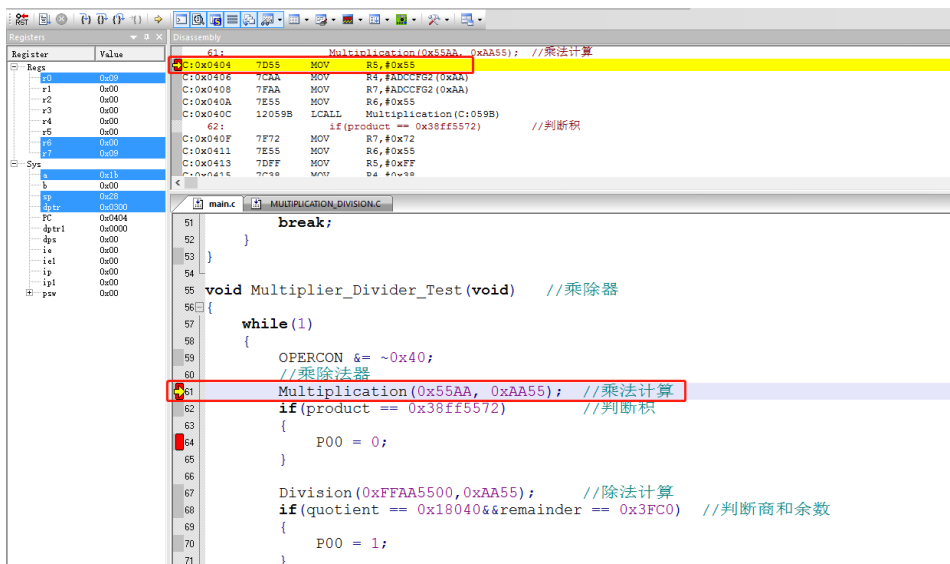


图 2.2

上图中，全速运行至第一个断点处，可以看出当程序停止到断点位置时，断点处左侧出现黄色调试箭头，指向当前的停止行。如果打开反汇编窗口(Disassembly)，同样会出现黄色调试箭头，并指向当前程序的

PC 地址。

当进入仿真状态后，程序默认是停止在 0x00 地址，即 PC 指 0x00，第一次全速运行至断点，将从该地址一直运行至第一个断点才停止，相比其他仿真操作，该操作的执行速度是最快的。

4、单步跟踪(Step)

单步跟踪是指，每次执行一行程序，执行完成该行程序后停止。

- ① 点击快捷图标“Step”按钮或者按下快捷键“F11”，程序会单步运行一次。需要注意的是，默认情况下，“Disassembly”窗口打开或者关闭，在 C 源程序窗口中，单步运行过程会有差异，建议在运行 Step 的过程中，保持“Disassembly”窗口为打开状态。
- ② “Disassembly”窗口的打开/关闭。点击快捷图标“Disassembly Window”，凹陷状态为打开，此时出现“Disassembly”窗口，并反汇编当前的程序行；平滑状态为关闭，“Disassembly”窗口消失。

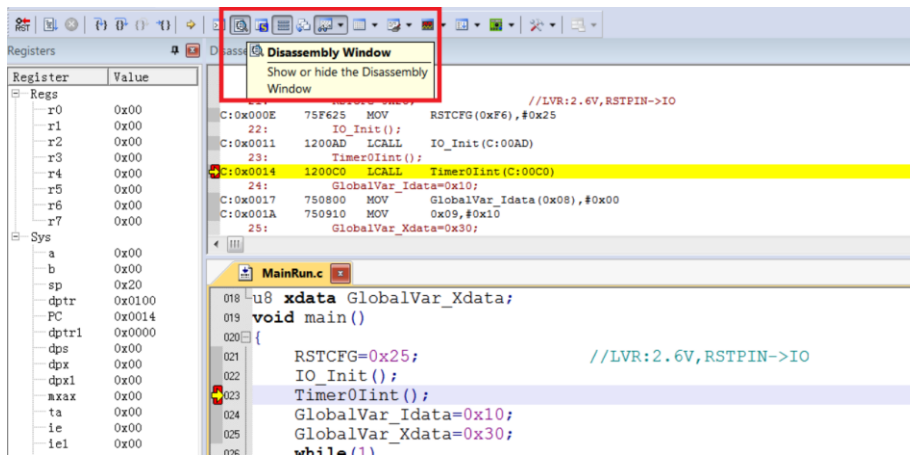


图 2.3

- ③ “Disassembly”窗口打开下运行 Step，当前程序停止在函数体 Timer0Init()，按下 F11 键，执行当前黄色箭头指向的程序行，然后箭头指向下一行。执行完成的结果如下图所示，不断按下 F11，程序将逐行执行下去。当“Disassembly”窗口打开时，默认是执行反汇编行，如果需要执行源码行，只需要鼠标左键选中一次源码界面即可。

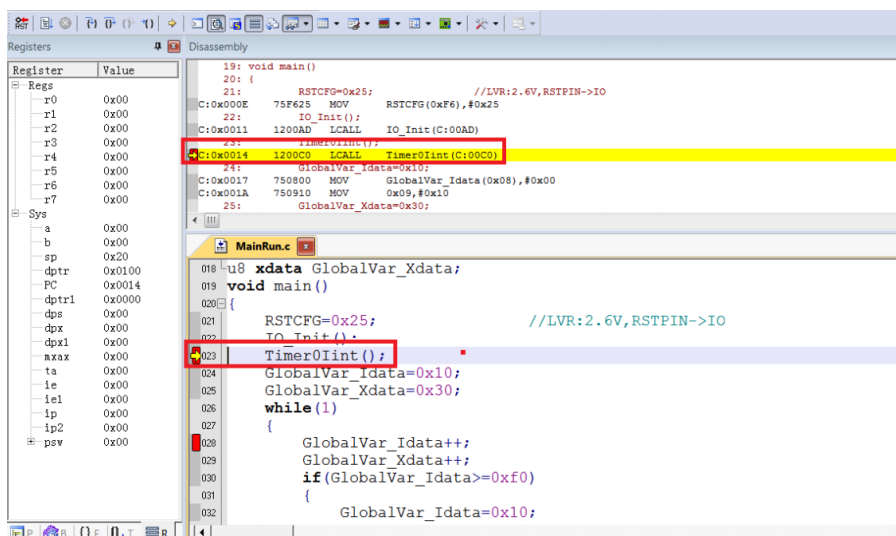


图 2.4

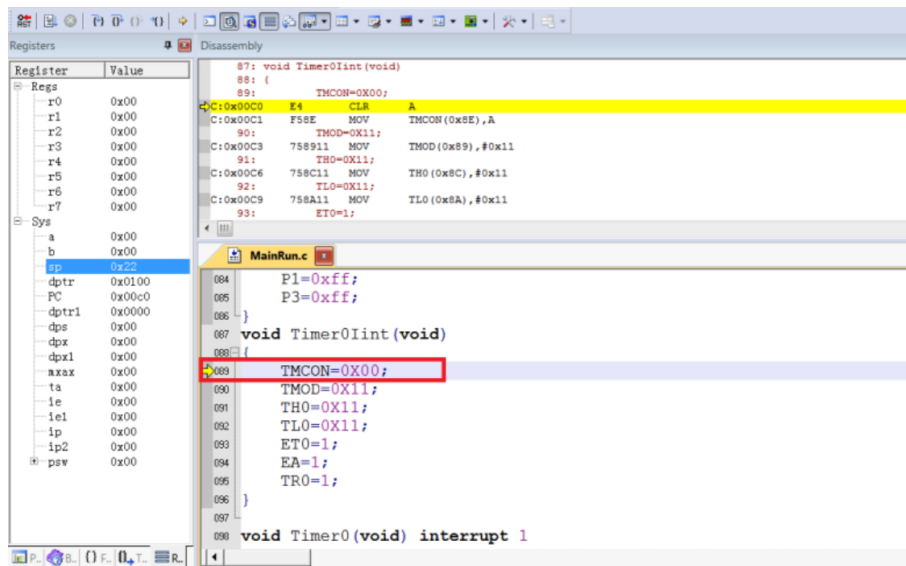


图 2.5

5. 跨步运行(Step Over)

当程序执行至子函数或者汇编中的 CALL/LCALL 时, 该操作不会进入子函数单步执行, 而是将子函数全速运行, 停在下一指令处。

注意:

- ①当程序执行到的位置不是子函数时, 该操作与单步跟踪结果相同;
- ②当子函数内存在断点时, 程序会优先停在断点处。

点击快捷图标“Step Over”按钮或快捷键“F10”操作即可。

以当前断点停止在函数体 Timer0Init()为例, 按下 F10 键, 调试光标不进入函数内部, 而是全速执行完该函数, 然后黄色箭头指向下一行, 如下图:

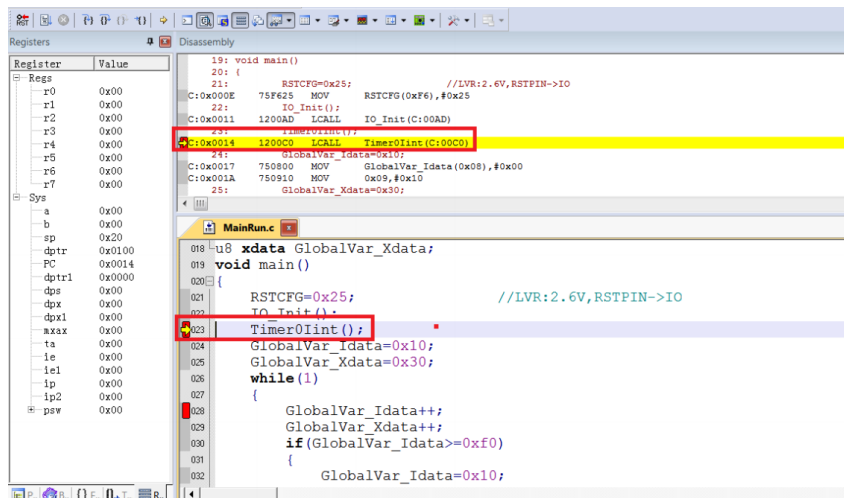


图 2.6

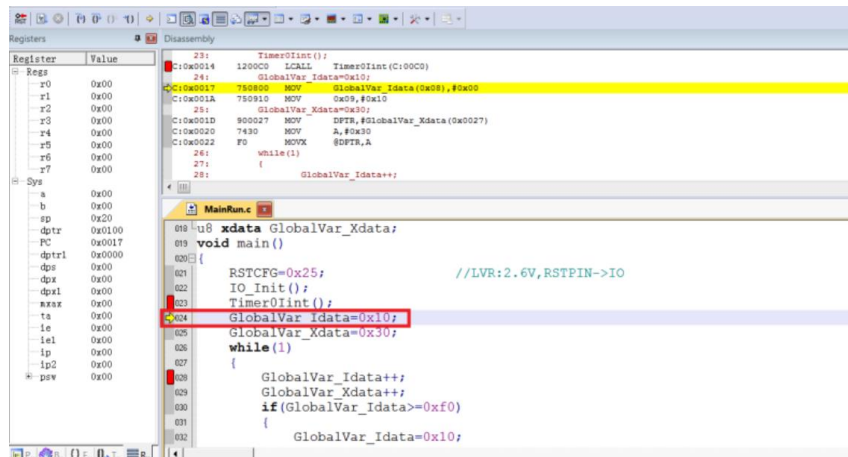


图 2.7

继续按下 F10，观察反汇编窗口，其运行结果与单步跟踪相同，如下图：

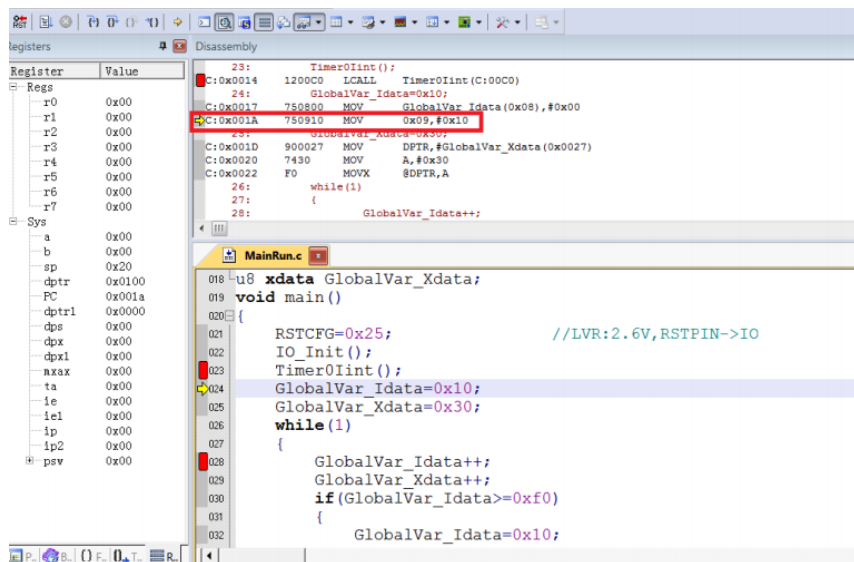


图 2.8

6、运行至光标处(Run to Cursor Line)

当断点间仿真操作通过单步跟踪或者跨步操作效率较低时，又希望直接运行至源程序行时，可通过运行至光标处(Run to Cursor Line)来完成；

该操作可通过按下快捷图标“Run to Cursor Line”或者快捷键“Ctrl+F10”实现；

以下图为例，希望程序直接停止在 Timer0Init()函数内的最后一行，选择使用运行至光标操作；

定位光标：鼠标左键点击一次预定位的程序行，出现蓝色箭头表示该行被选中，如下图 2.9：

注意：预定位的程序行一定是从当前黄色箭头开始能够运行到的，否则运行至光标处操作无效。

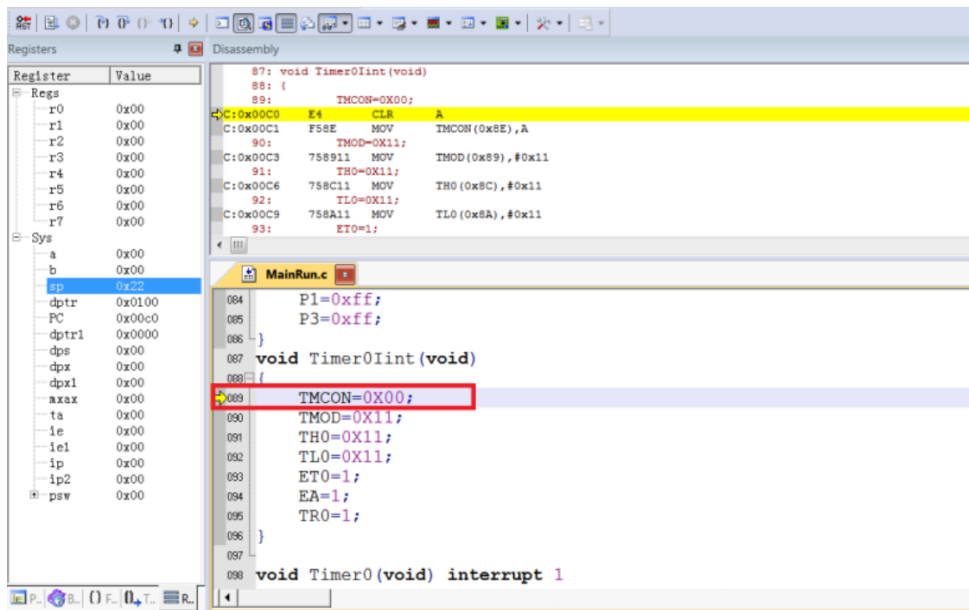


图 2.9

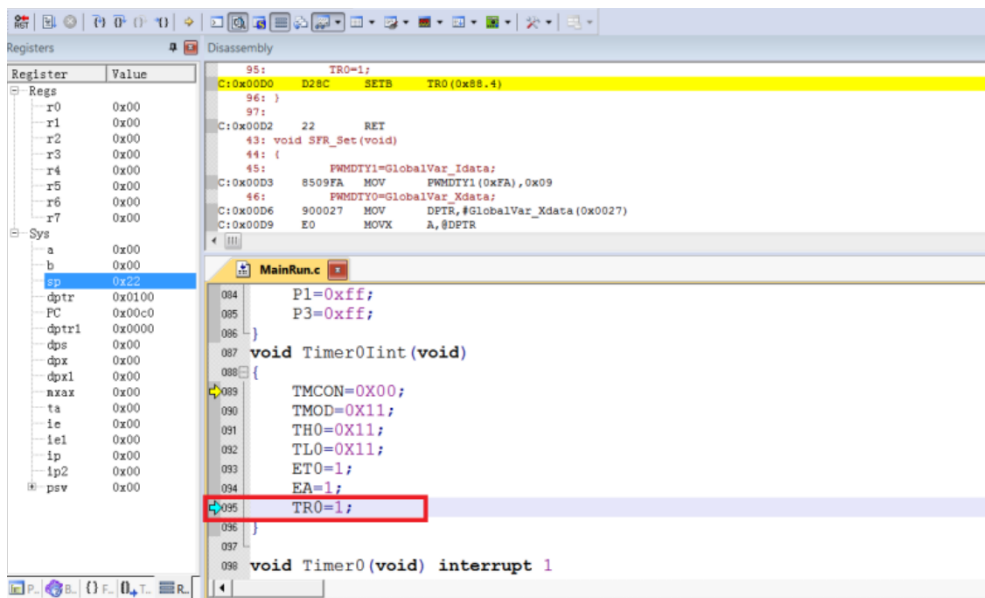


图 2.10

按下 Ctrl+F10，程序全速运行至光标处停止，定位光标处变为黄色箭头，如下图：

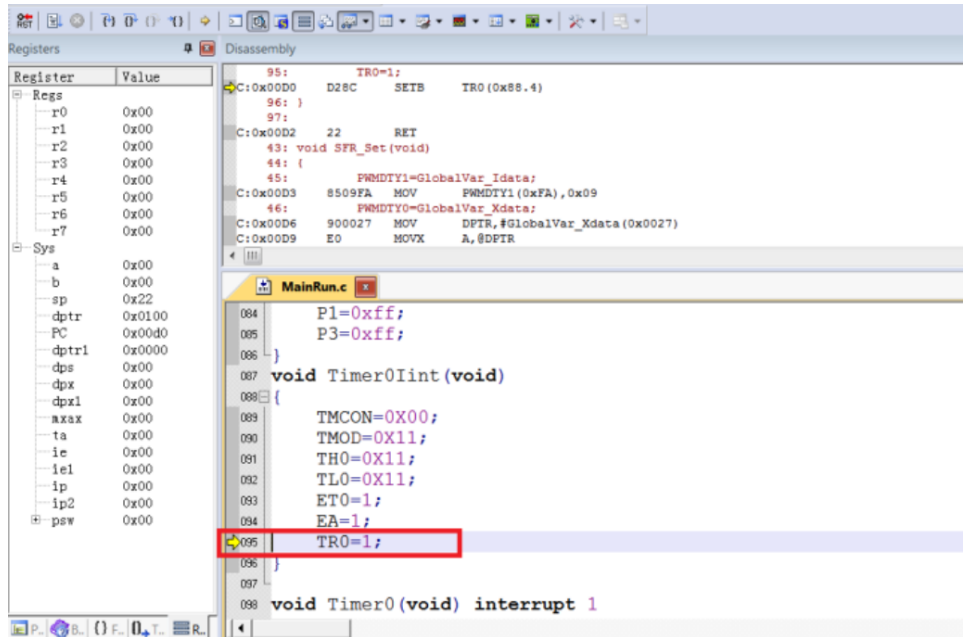


图 2.11

7、复位(Reset)

点击“Reset”按钮，程序重新复位，黄色箭头指向地址 0x00，如下图：

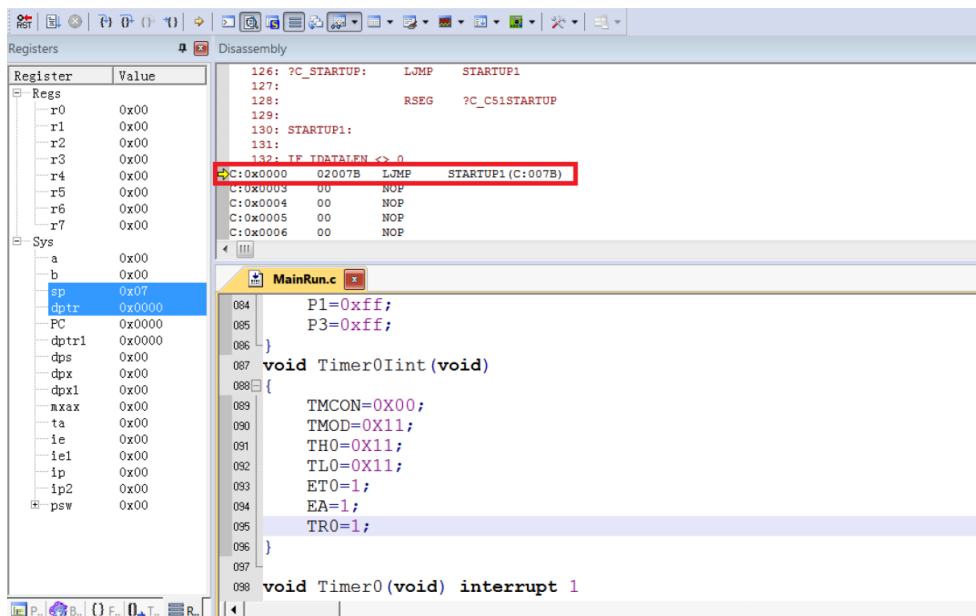


图 2.12

三、查看和修改变量

1、使用 Watch 窗口查看和修改变量

在仿真调试模式下，可通过观察窗口来查看或修改当前的变量。

① 打开 Watch 窗口：

点击快捷图标“Watch Windows”出现 3 个可选择窗口:Locals,Watch1,Watch2,如果这 3 个窗口的带浅黄色背景框,则表示选中,KEIL 界面下方会出现子窗口,否则,点击选择 Watch1 或者 Watch2 的任意一个,Watch 界面就会出现在 KEIL 界面的下方,如图

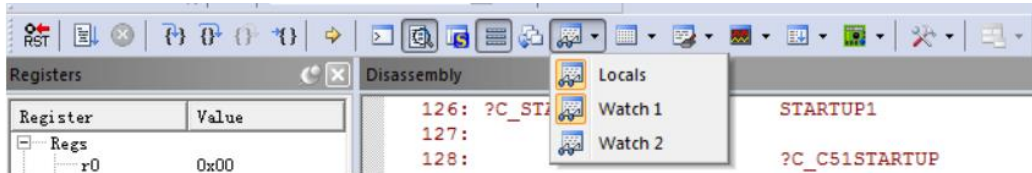


图 3.1

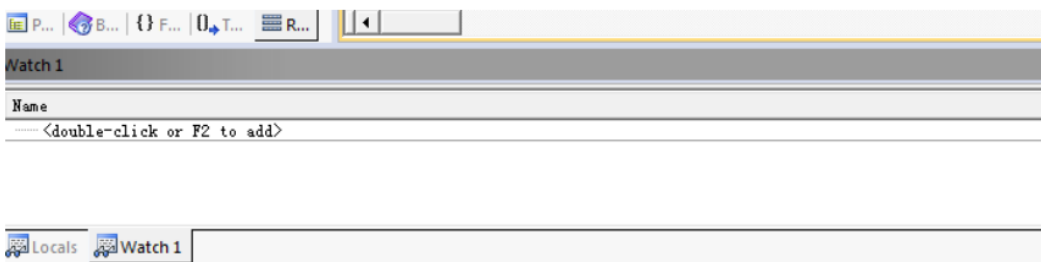


图 3.2

② 填写预查看/修改的变量名：

在“Name”栏下填入要操作的变量名,该变量名必须是源码中存在的,否则无效。此时其对应的“Value”栏会显示出该变量当前的值,如下图

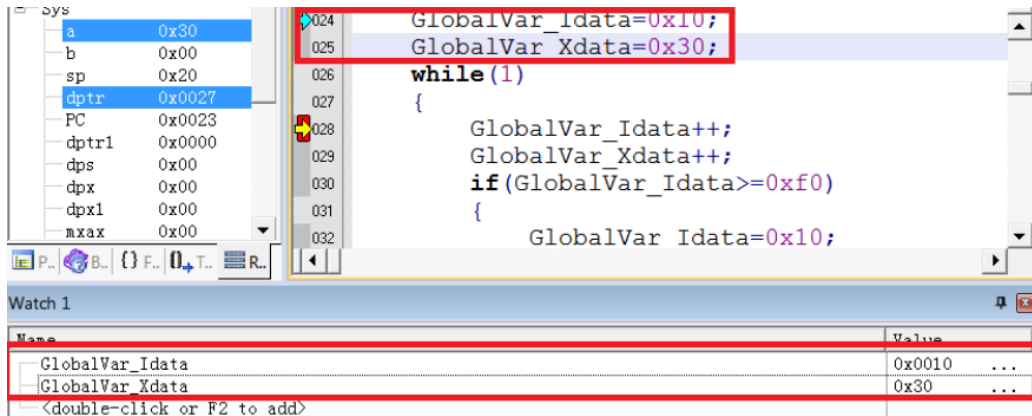


图 3.3

上图中,添加变量 GlobalVar_Idata、GlobalVar_Xdata, Watch1 中观察到该变量的值分别是 0x0010、0x30。另外,在 C 源码界面内,当鼠标移动到变量名的位置时,也会显示当前该变量的值及其类型和地址,如下图所示



图 3.4

③ 修改变量值:

在需要修改的变量对应的“Value”栏中，双击该值，进行修改，修改完成之后，在任意位置单击鼠标左键，当修改成功，该变量一栏显示变为深色，如图。

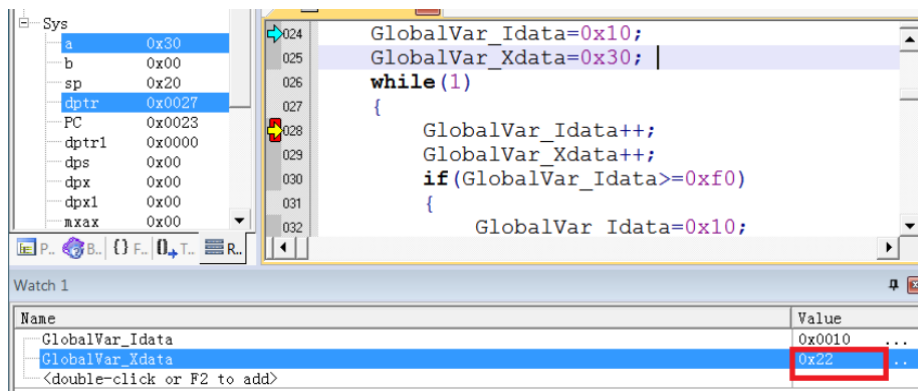


图 3.5

上图中，修改 GlobalVar_Xdata 的值为 0x22。

2、使用 Memory 查看和修改变量

① 打开 Memory 窗口

点“Memory Windows”出现 4 个可选择窗口:Memory1~Memory4，如图 3.2.1，如果这 4 个窗口的带浅黄色背景框，则表示选中，KEIL 界面下方会出现子窗口，否则，点击选择任意一个，Memory 界面就会出现在 KEIL 界面的下方，如图 3.2.2

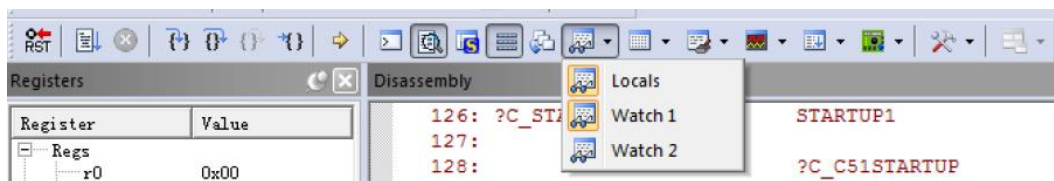


图 3.6



图 3.7

② 通过地址查看修改变量

在 **Memory** 的界面中，**Address** 栏用来输入待显示的存储区起始地址。如果知道当前变量的地址及变量所在的区域时，可通过如下命令完成。**Ram data 区: D: xx; RAM idata 区: I:xx; Ram Xdata 区: X:xx。**当上述命令填写完成，就可以显示从起始地址开始的变量值，同时也可以双击该值进行修改。

例如：**GlobalVar_Xdata** 为 Xdata 区，地址为 **0x27**，使用 **0x27** 为起始地址，查看 Xdata 区域的值，如图。

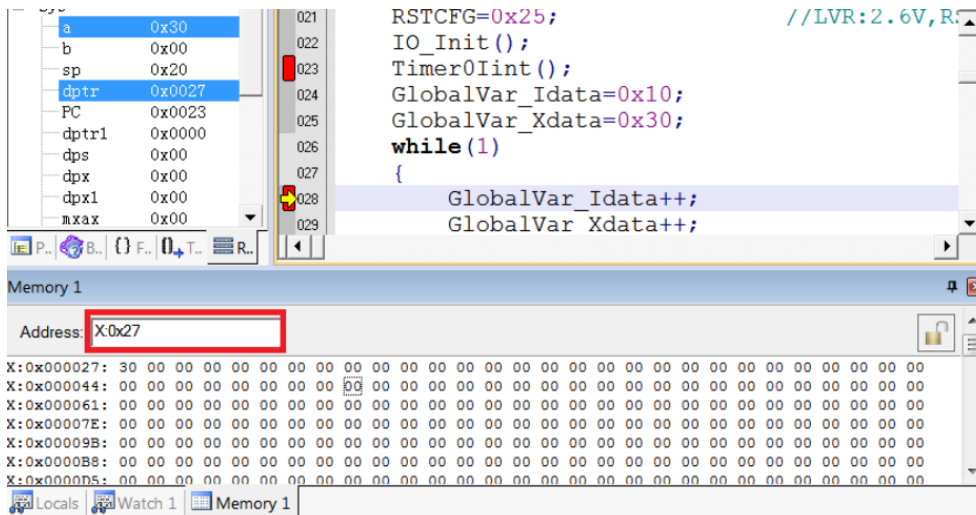


图 3.8

从上图看出，X:0x000027 地址的值为 **0x30**，即 **GlobalVar_Xdata** 的值为 **0x30**。

同样，也可在该位置修改变量的值，修改方式如下：

1.双击 **Memory** 中地址相应的值，进行修改，如图

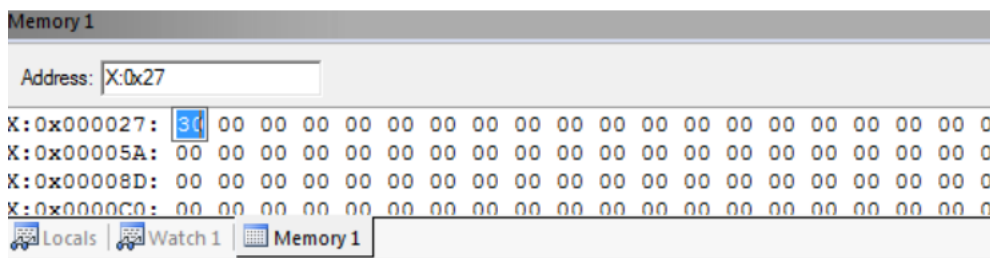


图 3.9

写入新的值，任意位置单击鼠标，完成修改，如图

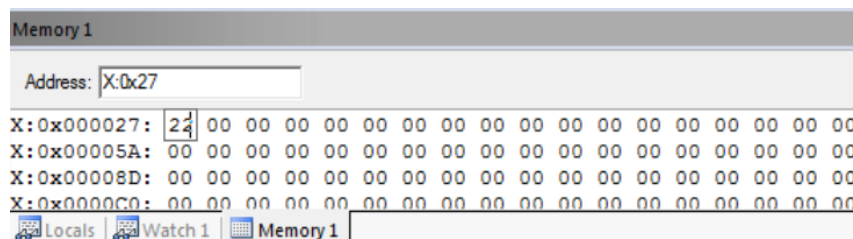


图 3.10

建议：如果对变量地址分配不清楚的情况下，请使用 Watch 窗口的方式。

3、查看和修改 SFR

① 打开 Watch 窗口

与查看和修改变量中打开 Watch 窗口相同。

a) 填写预查看/修改的 SFR 名

在“Name”栏下填入要操作 SFR 名，该 SFR 必须是头文件中存在的，否则无效。此时其对应的“Value”栏，会显示出该 SFR 当前的值，如图

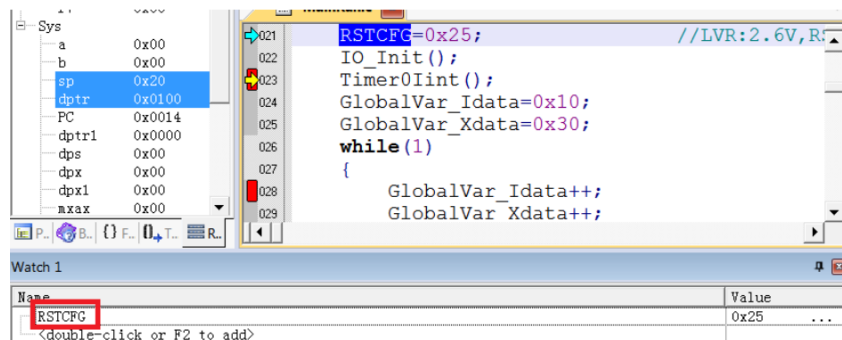


图 3.11

在需要修改的 SFR 对应的“Value”栏中，双击该值，进行修改，修改完成之后，在任意位置单击鼠标左键，当修改成功，该变量一栏显示变为深色，修改结果如图：

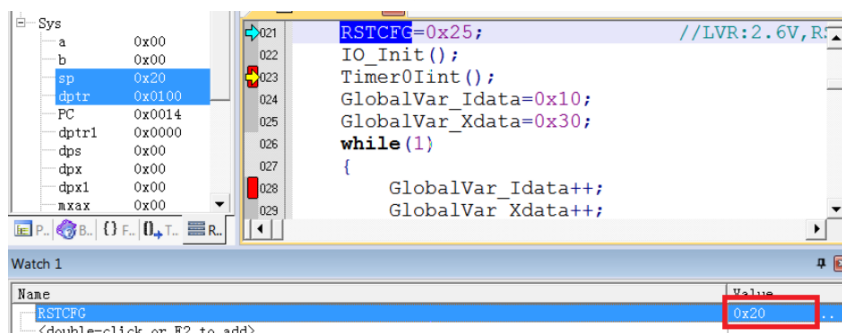


图 3.12