

keil 的信号函数用于模拟和测试串行 IO，模拟 IO，端口通讯等重复发生的外部事件。

信号函数以关键字 **signal** 开头。在函数中必须调用 **twatch** 用于延时，不然 keil 会陷入死循环。**twatch** 函数的参数表示要延时的 CPU 执行周期，比如 **twatch (1000000)** 表示一个信号函数延时 1000000 个 CPU 周期。

信号函数受到如下约束：

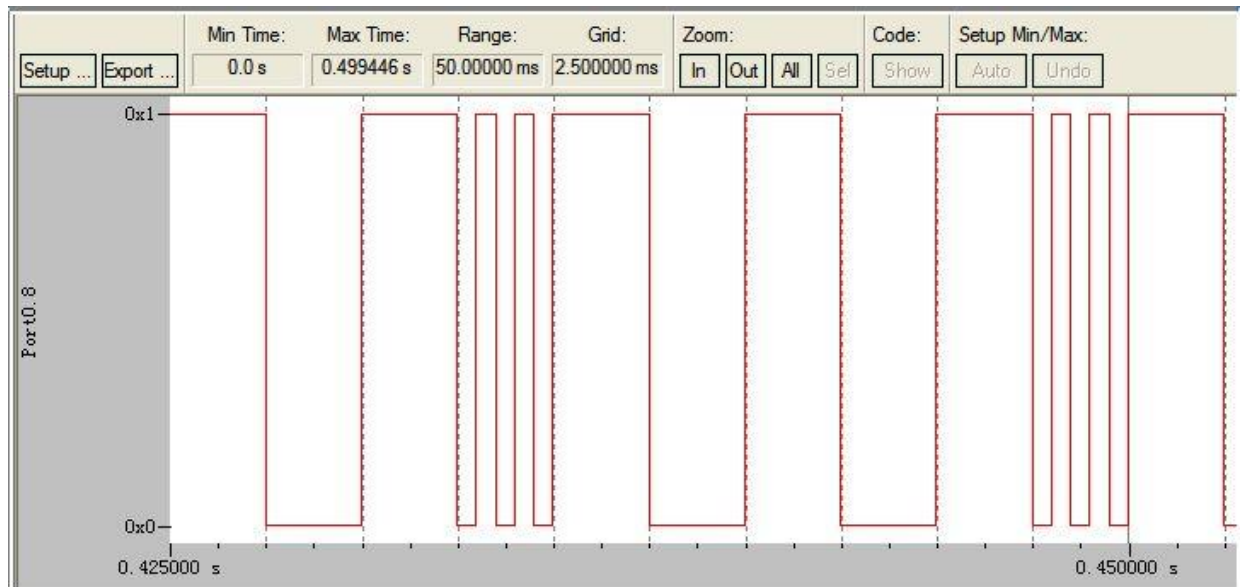
- 函数的返回值类型必须为 **void**。
- 函数最多只能有 8 个参数。
- 信号函数可以调用其他重定义函数和用户函数。
- 信号函数之间不能相互调用。
- 信号函数可以被用户函数调用。
- 信号函数必须调用 **twatch** 至少一次。如果信号函数从不调用 **twatch**，则目标程序将得不到时间执行。而且由于不能使用 **Ctrl+C** 中断信号函数，在这种情况下 μ Vision3 将进入死循环。

1.建立函数

keil 中有内建的函数编辑器，通过 **Debug--Function Editor** 来打开。打开函数编辑器时需要输入一个文件名或者打开一个由 **Options for Target--Debug Initialization File** 指定的文件。

2. 编辑例子。

下面的例子在 **LPC2114** 的 **P0.8** 引脚上输入一个如下图所示的方波。



[cpp] [view plaincopy](#)

```

1. // 在 200HZ 的方波的低半周期中增加两个频率为 1K 的波
2. //
3. //
4. signal void test(void)
5. {
6.     while(1)
7.     {
8.         PORT0 |=0x100;          //P0.8 置高
9.         twatch(11059200*4/400); //延时 2.5ms(200HZ)
10.        PORT0 &=~0x100;         //P0.8 置低
11.        twatch(11059200*4/400); //延时 2.5ms(200HZ)
12.
13.        PORT0 |=0x100;          //P0.8 置高
14.        twatch(11059200*4/400); //延时 2.5ms(200HZ)
15.        PORT0 &=~0x100;         //P0.8 置低
16.        twatch(11059200*4/400); //延时 2.5ms(200HZ)
17.
18.
19.        PORT0 |=0x100;          //P0.8 置高
20.        twatch(11059200*4/400); //延时 2.5ms(200HZ)
21.        PORT0 &=~0x100;         //P0.8 置低
22.        twatch(11059200*4/2000); //延时 ms(800HZ)
23.        PORT0 |=0x100;          //P0.8 置高
24.        twatch(11059200*4/2000); //延时 2.5ms(400HZ)
25.        PORT0 &=~0x100;         //P0.8 置低
26.        twatch(11059200*4/2000); //延时 2.5ms(800HZ)
27.        PORT0 |=0x100;          //P0.8 置高

```

```
28.      twatch(11059200*4/2000);    //延时 2.5ms(400HZ)
29.      PORT0 &=~0x100;            //P0.8 置低
30.      twatch(11059200*4/2000);    //延时 2.5ms(800HZ)
31.
32.  }
33. }
```

3.编译

在信号函数编辑界面上，点击 **Compile** 按钮，编译通过与否都会有相应提示。

4. 调用信号函数

编译通过后，在软件仿真时，在控制窗口（**Command** 窗口）中输入如下命令：

```
test（）
```

注：**test** 为信号函数名。

5.查看空闲信号函数

使用 **signal state** 命令

6. 删除信号函数（以 **test（）** 为例）

signal kill test 或者 **kill func test**