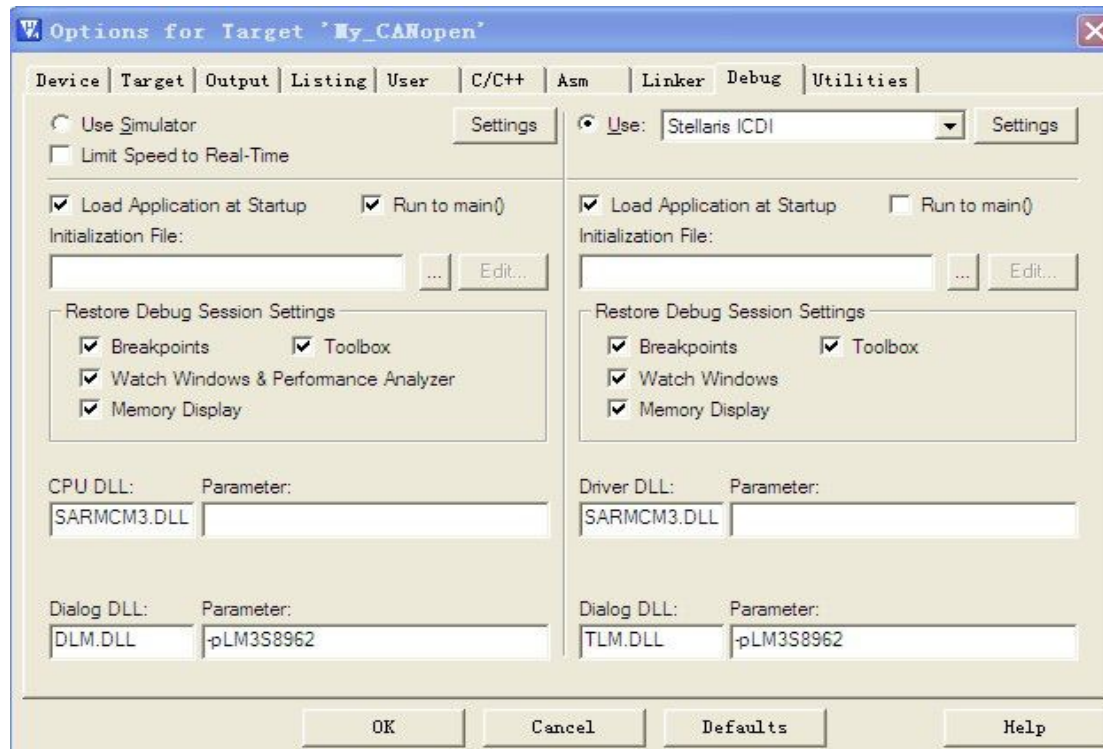


CANopen 中 Timer.h 和 can.h 与库头文件中的重名了，需要修改 CANopen 中的文件名，而且 CANopen 源文件中引用该头文件的地方也需要修改过来，改名加一个 CANopen 前缀。driverlib.lib 文件可能需要更新，之前遇到没有 CANbitset()函数吧！更新一下就可以了。

要用 ICD1 设置断点调试，需要对工程进行设置，之前是自己新建的工程，默认是选择左边的 Use Simulator，至少调试时提示不能写某地址吧！选择右边的后就可以调试了，得记住！



Keil 里全局查找 Message 时，在.h 文件中没有找到，其实是由 Message 的定义的吗，估计没有包含进入工程吧！我们只是包含了 h 文件的目录，并没有将其加入工程；这个真需要注意！可以将头文件添加进工程

需要实现的函数

setTimer()

getElapsedTime()

UNS8 canSend(CAN_PORT notused, Message *m)注意，之前原形弄错了，致使浪费了一点调试时间，程序不能发送心跳报文，

当然还有定时器中断以及 CAN 接收中断处理，

row_number 和 last_timer_raw 都是 16 位无符号整型，last_timer_raw = -1; row_number=0;

if (row_number == (TIMER_HANDLE)last_timer_raw + 1) last_timer_raw++;

这里必须要进行强制类型转换，否则右边为 65536，经过测试确认的。

编译错误：

1. SDO 中发现 SDO_BLOCK_SIZE 常量未定义，在 SDO.c 中增加定义#define SDO_BLOCK_SIZE 10
2. CO_Data 结构体中用到了未定义的 ObjDict_obj100C，故在对象字典中增加定义
3. Link 错误：比如未实现 void setTimer(TIMEVAL value); TIMEVAL getElapsedTime(void); 等

等。这个是因为没有实现上述函数，其实实现这几个函数是移植的关键：
在

SDO 测试结果截图，读取对象字典对象索引 0x1017 的第 1 个数据，

```
/* index 0x1017 :   Producer Heartbeat Time. */
UNSI16 ObjDict_obj1017 = 0x2710; /* 10000 */
ODCallback_t ObjDict_Index1017_callbacks[] =
{
    NULL,
};
subindex ObjDict_Index1017[] =
{
    { RW, uint16, sizeof (UNSI16), (void*)&ObjDict_obj1017 }
};
```

选择CAN路数: 1

发送格式: 正常发送

☒ 每次发送单帧 ☐ 每次发送 10 帧 ☐ 帧ID每发送一帧递增

帧类型: 标准帧

帧ID: 00000601

数据: 40 17 10 00 00 00 00 00 ...

发送

启动CAN

复位CAN

帧格式: 数据帧

发送次数: 1 次

每次发送间隔: 1000 ms

停止

序号	传输方向	第几路CAN	时间标识	帧ID	帧格式	帧类型	数据长度	数据
00000090	接收	1	0x00bdee44	0x00000701	数据帧	标准帧	0x01	05
00000091	接收	1	0x00ba3782	0x00000701	数据帧	标准帧	0x01	05
00000092	接收	1	0x00be80e0	0x00000701	数据帧	标准帧	0x01	05
00000093	接收	1	0x00bec9fe	0x00000701	数据帧	标准帧	0x01	05
00000094	接收	1	0x00bf133c	0x00000701	数据帧	标准帧	0x01	05
00000095	接收	1	0x00bf5c7a	0x00000701	数据帧	标准帧	0x01	05
00000096	接收	1	0x00bf5b8	0x00000701	数据帧	标准帧	0x01	05
00000097	接收	1	0x00bfef6	0x00000701	数据帧	标准帧	0x01	05
00000098	接收	1	0x00c03834	0x00000701	数据帧	标准帧	0x01	05
00000099	接收	1	0x00c08172	0x00000701	数据帧	标准帧	0x01	05
00000100	接收	1	0x00c0cb46	0x00000701	数据帧	标准帧	0x01	05
00000101	接收	1	0x00c11484	0x00000701	数据帧	标准帧	0x01	05
00000102	接收	1	0x00c15dc2	0x00000701	数据帧	标准帧	0x01	05
00000103	接收	1	0x00c1a700	0x00000701	数据帧	标准帧	0x01	05
00000104	发送	1	无	0x00000801	数据帧	标准帧	0x08	40 17 10 00 00 00 00 00
00000105	接收	1	0x00c1e198	0x00000581	数据帧	标准帧	0x08	4b 17 10 00 10 27 00 00
00000106	接收	1	0x00c1f03e	0x00000701	数据帧	标准帧	0x01	05
00000107	接收	1	0x00c2397c	0x00000701	数据帧	标准帧	0x01	05
00000108	接收	1	0x00c282ba	0x00000701	数据帧	标准帧	0x01	05
00000109	接收	1	0x00c2cbf8	0x00000701	数据帧	标准帧	0x01	05
00000110	接收	1	0x00c31536	0x00000701	数据帧	标准帧	0x01	05
00000111	接收	1	0x00c35fd0	0x00000701	数据帧	标准帧	0x01	05
00000112	接收	1	0x00c3a846	0x00000701	数据帧	标准帧	0x01	05
00000113	接收	1	0x00c3f186	0x00000701	数据帧	标准帧	0x01	05

23 00 18 01 01 02 00 00//改写索引 0x1800 子索引 0x01 即 TPDO1 对应的帧 ID 为 0x00000201
40 00 18 01 00 00 00 00//读取索引 0x1800 子索引 0x01 即 TPDO1 对应的帧 ID
下图测试结果显示正确实现了 SDO 读写功能

序号	传输方向	第几路CAN	时间标识	帧ID	帧格式	帧类型	数据长度	数据
00000301	接收	1	0x03c993ea	0x00000202	数据帧	标准帧	0x01	33
00000302	接收	1	0x03c9ae48	0x00000701	数据帧	标准帧	0x01	05
00000303	接收	1	0x03c9f786	0x00000701	数据帧	标准帧	0x01	05
00000304	接收	1	0x03ca40c4	0x00000701	数据帧	标准帧	0x01	05
00000305	接收	1	0x03ca8a02	0x00000701	数据帧	标准帧	0x01	05
00000306	接收	1	0x03cad3d6	0x00000701	数据帧	标准帧	0x01	05
00000307	接收	1	0x03cb1d14	0x00000701	数据帧	标准帧	0x01	05
00000308	接收	1	0x03cb6652	0x00000701	数据帧	标准帧	0x01	05
00000309	发送	1	无	0x00000801	数据帧	标准帧	0x08	40 00 18 01 00 00 00 00
00000310	接收	1	0x03cbaae0	0x00000581	数据帧	标准帧	0x08	43 00 18 01 02 02 00 00
00000311	接收	1	0x03cbaf90	0x00000701	数据帧	标准帧	0x01	05
00000312	接收	1	0x03cbf8ce	0x00000701	数据帧	标准帧	0x01	05
00000313	接收	1	0x03cc420c	0x00000701	数据帧	标准帧	0x01	05
00000314	接收	1	0x03cc8b4a	0x00000701	数据帧	标准帧	0x01	05
00000315	发送	1	无	0x00000801	数据帧	标准帧	0x08	23 00 18 01 01 02 00 00
00000316	接收	1	0x03ccacb0	0x00000581	数据帧	标准帧	0x08	60 00 18 01 00 00 00 00
00000317	接收	1	0x03cc4488	0x00000701	数据帧	标准帧	0x01	05
00000318	接收	1	0x03cd1dc6	0x00000701	数据帧	标准帧	0x01	05
00000319	接收	1	0x03cd6704	0x00000701	数据帧	标准帧	0x01	05
00000320	接收	1	0x03cdb0d8	0x00000701	数据帧	标准帧	0x01	05
00000321	发送	1	无	0x00000801	数据帧	标准帧	0x08	40 00 18 01 00 00 00 00
00000322	接收	1	0x03cde71c	0x00000581	数据帧	标准帧	0x08	43 00 18 01 01 02 00 00
00000323	接收	1	0x03cdfa16	0x00000701	数据帧	标准帧	0x01	05
00000324	接收	1	0x03ce4354	0x00000701	数据帧	标准帧	0x01	05
00000325	发送	1	无	0x00000801	数据帧	标准帧	0x08	40 00 18 01 00 00 00 00
00000326	接收	1	0x03ce7c2a	0x00000201	数据帧	标准帧	0x01	47
00000327	接收	1	0x03ce8c92	0x00000701	数据帧	标准帧	0x01	05
00000328	接收	1	0x03ced5d0	0x00000701	数据帧	标准帧	0x01	05
00000329	接收	1	0x03cf1f0e	0x00000701	数据帧	标准帧	0x01	05

To **read** the data 0xd0d1... in the server node object dictionary, the client node sends :

The client request :

600 + Serv NodeId	0	40	Index	Sub index	00	00	00	00
----------------------	---	----	-------	--------------	----	----	----	----

The server responds (if success) :

580 + Serv NodeId	0	4F	Index	Sub index	d1	x	x	x
----------------------	---	----	-------	--------------	----	---	---	---

X : undefined. Should be 0

The server responds (if success) :

580 + Serv NodeId	0	4B	Index	Sub index	d1	d0	x	x
----------------------	---	----	-------	--------------	----	----	---	---

X : undefined. Should be 0

The server responds (if success) :

580 + Serv NodeId	0	47	Index	Sub index	d2	d1	d0	x
----------------------	---	----	-------	--------------	----	----	----	---

X : undefined. Should be 0

The server responds (if success) :

580 + Serv NodeId	0	43	Index	Sub index	d3	d2	d1	d0
----------------------	---	----	-------	--------------	----	----	----	----