

EECS 223:

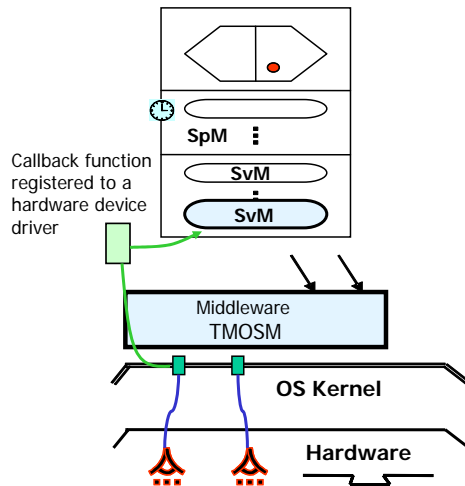
**Introduction to
Non-TMO-to-TMO OnewaySR
Programming**

Programming I/O Activities in TMO -- Pt. 2

05.4.22 | 1



**Basic styles of writing reactive control –
RC2: SvM invoked Indirectly by a Device Driver**



- To write an SvM which is **invoked upon occurrence of a specified signal from the underlying OS kernel**.
- Not broadly applicable.
 - The device driver must be built to use a callback function.
 - Often, ISRs (and DPC target functions) of peripherals do not seem to provide necessary hooks.
 - So, often necessary to modify an ISR to schedule an invocation of (or a DPC for) a callback function which will in turn invoke the SvM.

05.4.22 | 2



OneWaySR from Non-TMO threads to TMO

- An API for calling from non-TMO functions (e.g., callback functions) for SvMs in TMOs.
- Both non-TMO functions and TMOs to be called should be on the same node with an IP address.
- TMOsL provides a class **CGate_4_NonTMO** which has a member function serving as an API for calling an SvM.

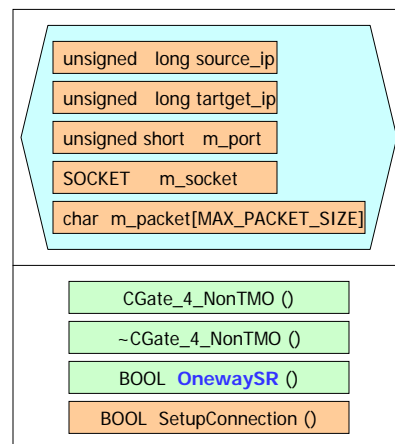
05.4.22 3



CGate_4_NonTMO

```
class CGate_4_NonTMO
{
public:
    CGate_4_NonTMO ();
    ~CGate_4_NonTMO ();
    BOOL OnewaySR (
        TCHAR * target_tmo_name,
        TCHAR * target_svm_name,
        void* ParamPtr, int ParamSize);
private:
    BOOL SetupConnection (unsigned short port);
    unsigned short m_port;
    unsigned long source_ip;
    unsigned long target_ip;
    SOCKET m_socket;
    char m_packet [MAX_PACKET_SIZE];
};
```

- Used to instantiate a gate for the TMO world in the local node



orange private green public

05.4.22 4



CGate_4_NonTMO in TMOSL

- Used to instantiate a gate for the TMO world in the local node
- Object method

```
BOOL OneWaySR (TCHAR * target_tmo_name, TCHAR *
               target_svm_name, void* ParamPtr, int ParamSize);
```

- `target_tmo_name`
[in] name of a target TMO
- `target_svm_name`
[in] name of a target SvM to be triggered
- `ParamPtr`
[in] pointer to a parameter-message to be sent
- `ParamSize`
[in] size of the parameter-message which ParamPtr is pointing to

* Note that there is **no ORT** parameter. -- This is different from the case of OneWaySR () used inside the TMO world.

Return value

- **TRUE** is returned if the function succeeds.
- Otherwise, **FALSE** is returned.

05.4.22

5



Example

TestNonTMOonewaySR.h

```
typedef struct _ParamStruct
{
    TCHAR content[200];
    int size;
} ParamStruct;

class TMO1: public CTMOBase
{
public:
    TMO1 (TCHAR * TMO_name, TCHAR *
          RMMC_name, tms TMO_start_time);
private:
    //The SvM to call
    int SvM1 (ParamStruct *);
};
```

TestNonTMOonewaySR.cpp

```
TMO1::TMO1 (TCHAR * TMO_name, TCHAR *
            SvM_name, tms TMO_start_time)
{
    SvM_RegistParam svm_regist_param;
    _tcscopy (svm_regist_param.name, SvM_name);
    // Register svm along with the registration
    // parameter
    BOOL result = RegisterSvM ( (PFSvMBody)
                               SvM1, & svm_regist_param);

    // fill out tmo registration parameters
    TMO_RegistParam tmo_regist_param;
    _tcscopy (tmo_regist_param.global_name,
             TMO_name);
    tmo_regist_param.start_time =
        TMO_start_time;

    // register this TMO to the TMOSM
    result = RegisterTMO (& tmo_regist_param);
}
```

05.4.22

6



Example (cont.)

TestNonTMOOnewaySR.cpp (cont')

```
int TMO1::SvM1 (ParamStruct * param)
{
    //Print out the message
    TMOSLprintf (_T("Receive a message:: %s
issued at %l64d\n"), param,
SRTimestamp);
    return 1;
}
```

- Since a non-TMO program cannot utilize TMO SM services, the sending timestamp cannot be generated.

==> Inside the SvM function body, a call for `GetSRTimestamp ()` must not appear.

```
//Non-TMO thread
void ThreadRoutine (void)
{
    //create a communication gate for NonTMO
    CGate_4_NonTMO gate;

    //Prepare the date to be sent
    TCHAR buf [200];
    _tcscpy (buf, _T("This is message to be sent from
NonTMO thread to TMO method.\n"));

    ParamStruct param;
    param.size = sizeof (ParamStruct);
    _tcscpy (param.content, buf);

    //Send out the Data
    gate.OnewaySR (_T("TMO1"), _T("SvM1"), (void*)
& param, sizeof (ParamStruct), tm4_DCS_age
(GetCurrentDCSage () + 50 * 1000));
}
```

05.4.22

7

UCI
DREAM Lab

Example (cont.)

```
void main ()
{
    StartTMOengine ();
    tms TMO_start_time1 = tm4_DCS_age (3*1000*1000);
    TMO1 tmo1 (_T("TMO1"), _T("SvM1"), TMO_start_time1);
    //Create a non-TMO thread
    HANDLE nonTMOThread = CreateThread
        (NULL, FALSE,
        (LPTHREAD_START_ROUTINE) ThreadRoutine,
        NULL, FALSE, NULL);

    MainThrSleep ();
}
```

05.4.22

8

UCI
DREAM Lab