TINE Release 4.0 News

(March 6, 2009: That was the week that was !)

“What a long, strange trip it’s been ….”

TINE Kernel : Recent Bug Fixes

- Lossy or busy Network => packet loss
  - Problem with 'retrying' a contract with large data set which returns fewer bytes than requested
  - Note: ‘retries’ will return the last results (if the contract is still available).
  - (Thank you Juergen Maass)
- Problem with 'packed' contract requests containing 'extended string space'
  - Packed : more than 1 at the same time
  - Extended string space: those cdi calls with devName = "dev1,dev2,dev3,....“
  - (Thank you Markus Walla)

TINE 4.0.9: News

- Latest Libs (tine32.dll, tine64.dll) work on Vista, Server2008
  - New winsock kernel from MicroSoft behaves differently with certain calls (e.g. ‘bind’).
  - -> problem with multiple clients on same host fixed!
- New aliases on Central Archive Server !
  - /PETRA/HISTORY/<device> [property]

    can be accessed as

    /PETRA/HISTORY/<device> [serverOfOrigin.propertyOfOrigin]
  - e.g. /DESY2/HISTORY/D2-1:2A [VacPressureAve.D2]

    ->

    /DESY2/HISTORY/D2-1:2A [IEVAC_D2.D2VacPressAve]
  - If an application is accessing

    /DESY2/IEVAC_D2/D2-1:2A [D2VacPressAve]

    then the centrally stored data can always be accessed as

    /DESY2/HISTORY/D2-1:2A [IEVAC_D2.D2VacPressAve]
  - Note: There are still too many cases where property “ALL_DATA” (!) needs to be given a useful name on the central archive server.

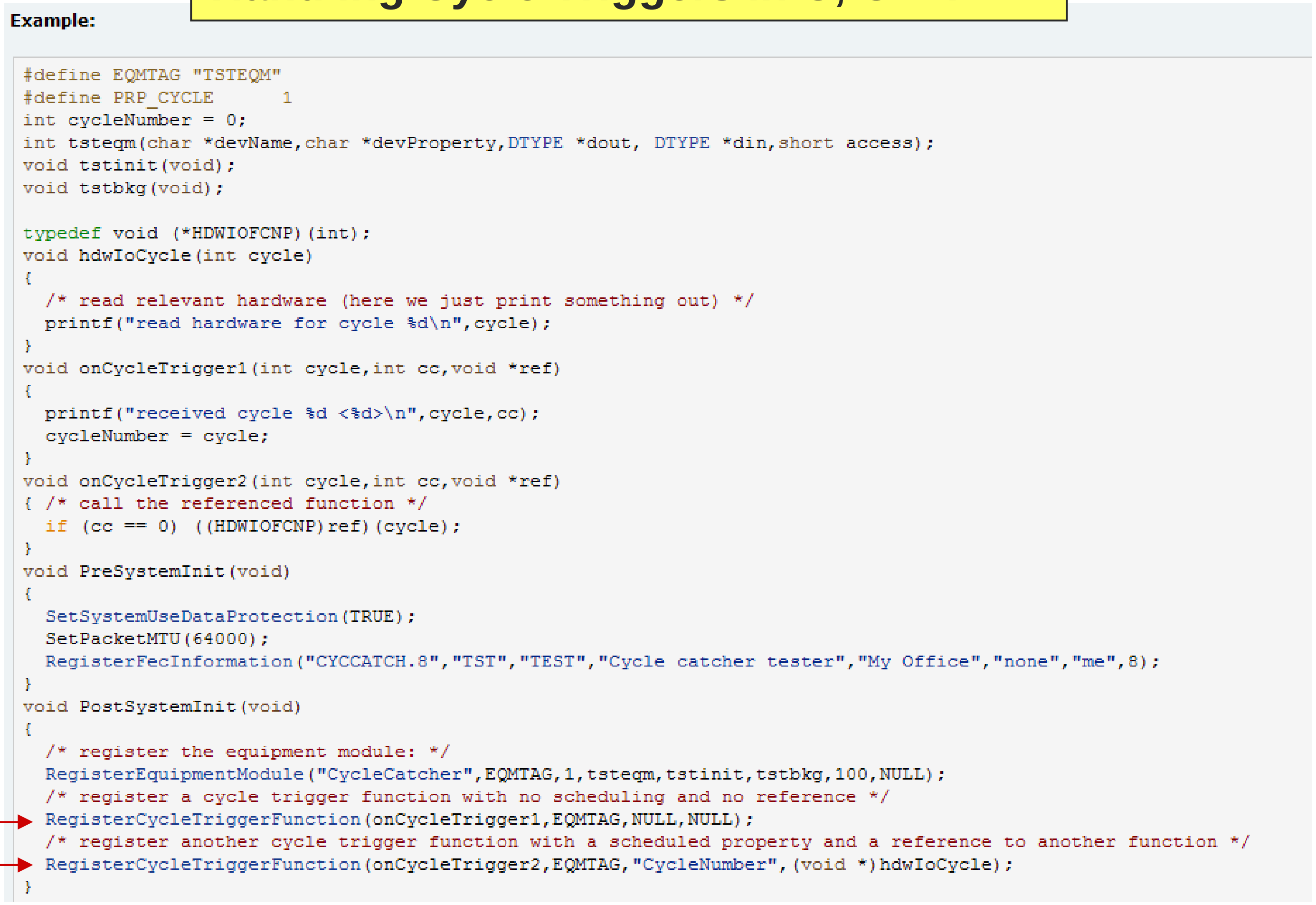
TINE 4.0.9: News

- Most command line tools now take optional switches to explicitly set the
  - Size
  - Format
  - Timeout
- TINE web page now has updated documentation on
  - Stock Properties
  - Meta Properties
  - Central Alarm Server (+ configuration)
  - Central Archive Server (+ configuration)
- TINE Forum now has several active users and several ‘threads’ !

PLEASE try it out!

TINE 4.0.9: Cycle Triggers

Handling Cycle Triggers in C, C++:

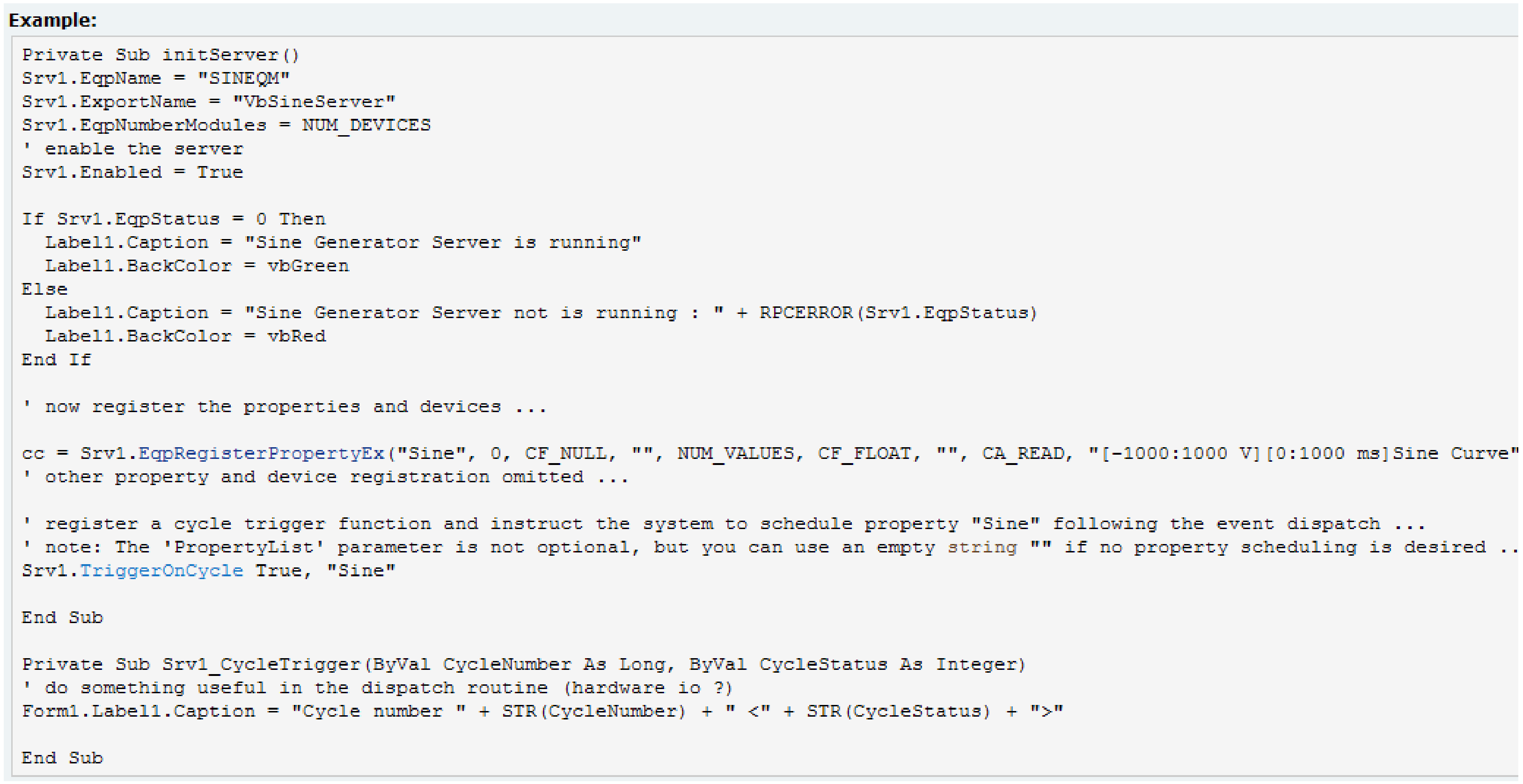


Example:

```c
#define EQMTAG "TSTEQM"
#define PRP_CYCLE      1
int cycleNumber = 0;
int tsteqm(char *devName,char *devProperty,DTYPE *dout, DTYPE *din,short access);
void tstinit(void);
void tstbkg(void);

typedef void (*HDWIOFCNP)(int);
void hdwIoCycle(int cycle)
{
  /* read relevant hardware (here we just print something out) */
  printf("read hardware for cycle %d\n",cycle);
}
void onCycleTrigger1(int cycle,int cc,void *ref)
{
  printf("received cycle %d <%d>\n",cycle,cc);
  cycleNumber = cycle;
}
void onCycleTrigger2(int cycle,int cc,void *ref)
{ /* call the referenced function */
  if (cc == 0)  ((HDWIOFCNP)ref)(cycle);
}
void PreSystemInit(void)
{
  SetSystemUseDataProtection(TRUE);
  SetPacketMTU(64000);
  RegisterFecInformation("CYCCATCH.8","TST","TEST","Cycle catcher tester","My Office","none","me",8);
}
void PostSystemInit(void)
{
  /* register the equipment module: */
  RegisterEquipmentModule("CycleCatcher",EQMTAG,1,tsteqm,tstinit,tstbkg,100,NULL);
  /* register a cycle trigger function with no scheduling and no reference */
  RegisterCycleTriggerFunction(onCycleTrigger1,EQMTAG,NULL,NULL);
  /* register another cycle trigger function with a scheduled property and a reference to another function */
  RegisterCycleTriggerFunction(onCycleTrigger2,EQMTAG,"CycleNumber",(void *)hdwIoCycle);
}
```

TINE 4.0.9: Cycle Triggers

Handling Cycle Triggers in VB with srv.ocx

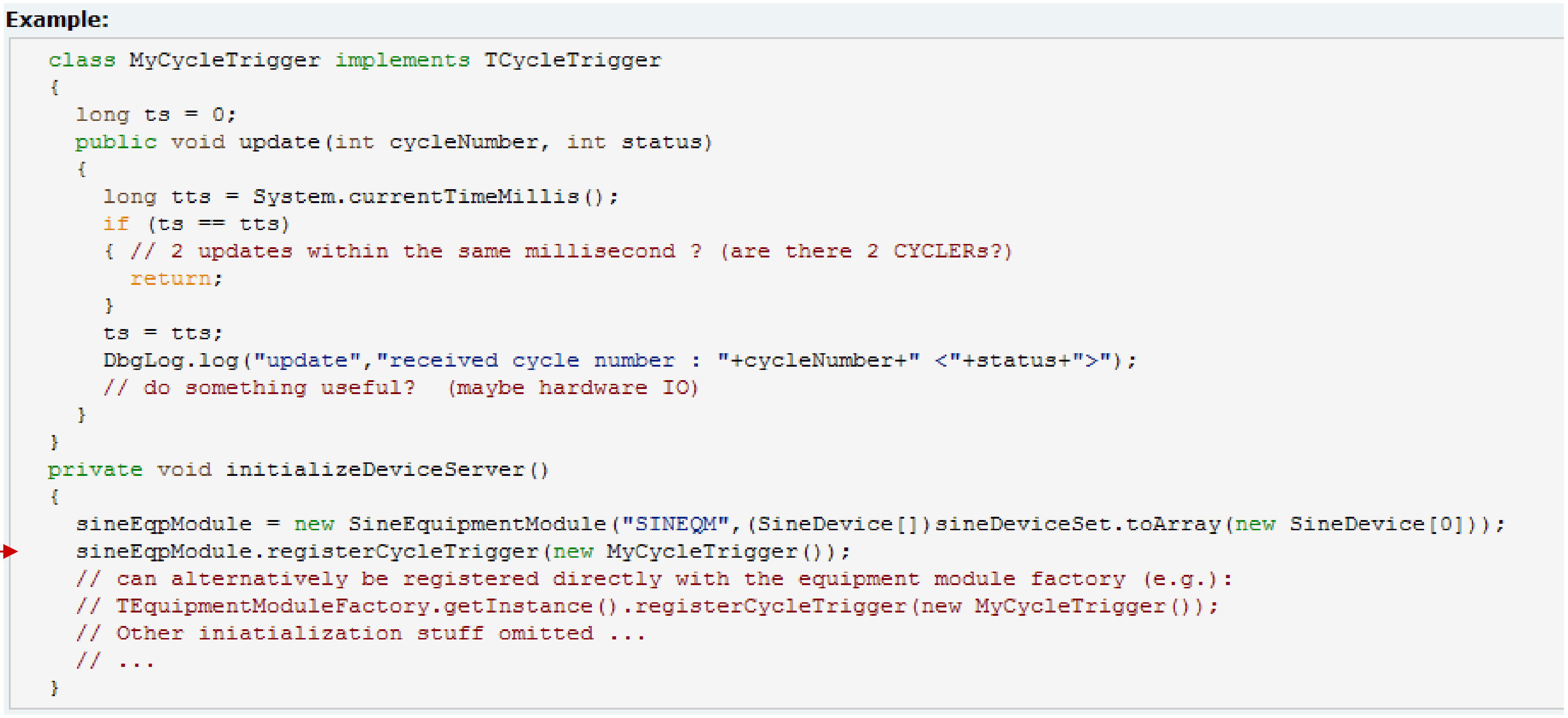


Example:

```
Private Sub initServer()
Srv1.EqpName = "SINEQM"
Srv1.ExportName = "VbSineServer"
Srv1.EqpNumberModules = NUM_DEVICES
' enable the server
Srv1.Enabled = True

If Srv1.EqpStatus = 0 Then
  Label1.Caption = "Sine Generator Server is running"
  Label1.BackColor = vbGreen
Else
  Label1.Caption = "Sine Generator Server not is running : " + RPCERROR(Srv1.EqpStatus)
  Label1.BackColor = vbRed
End If

' now register the properties and devices ...

cc = Srv1.EqpRegisterPropertyEx("Sine", 0, CF_NULL, "", NUM_VALUES, CF_FLOAT, "", CA_READ, "[-1000:1000 V][0:1000 ms]Sine Curve"
' other property and device registration omitted ...

' register a cycle trigger function and instruct the system to schedule property "Sine" following the event dispatch ...
' note: The 'PropertyList' parameter is not optional, but you can use an empty string "" if no property scheduling is desired ..
Srv1.TriggerOnCycle True, "Sine"

End Sub

Private Sub Srv1_CycleTrigger(ByVal CycleNumber As Long, ByVal CycleStatus As Integer)
' do something useful in the dispatch routine (hardware io ?)
Form1.Label1.Caption = "Cycle number " + STR(CycleNumber) + " <" + STR(CycleStatus) + ">"

End Sub
```

TINE 4.0.9: Cycle Triggers

Handling Cycle Triggers in Java

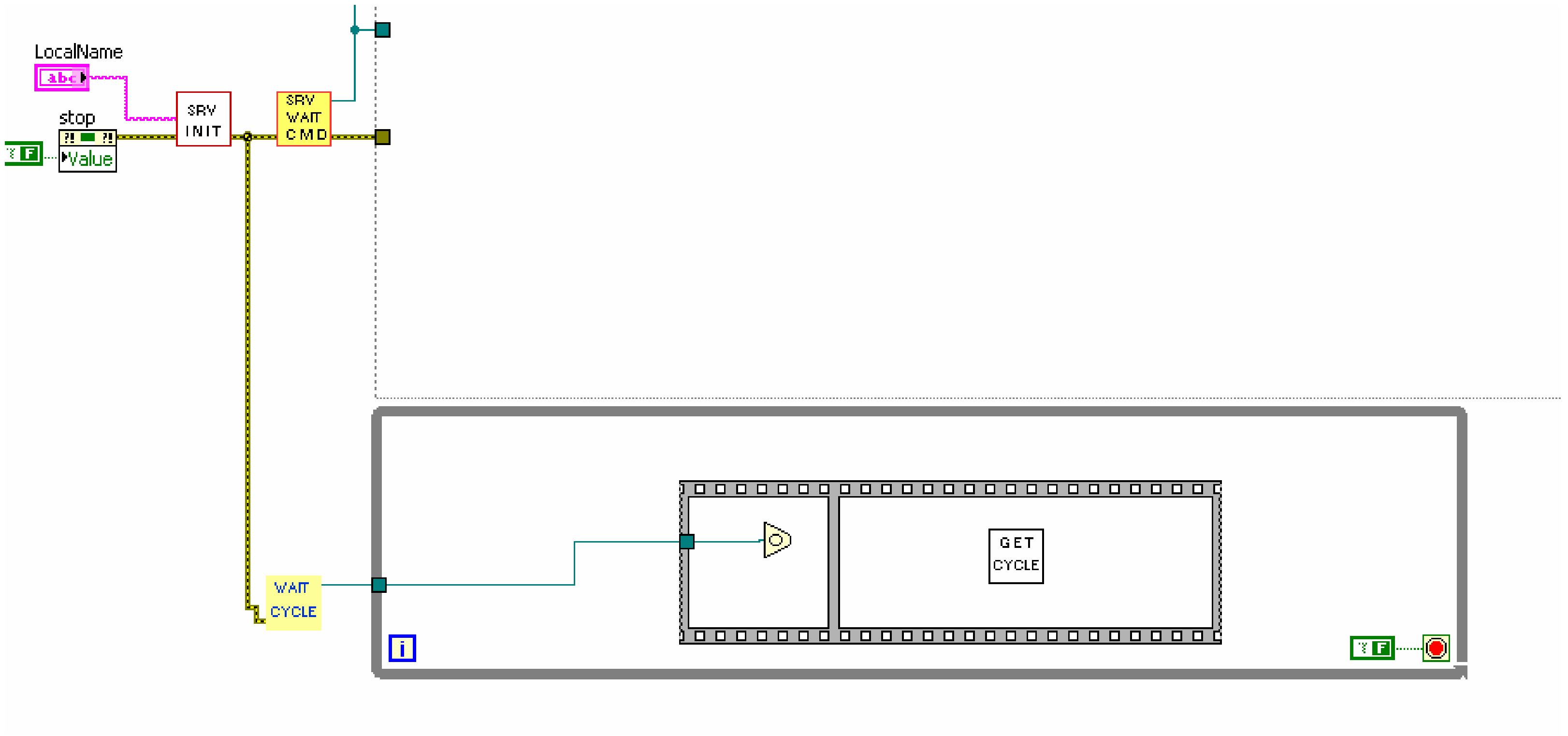


Example:

```
class MyCycleTrigger implements TCycleTrigger
{
  long ts = 0;
  public void update(int cycleNumber, int status)
  {
    long tts = System.currentTimeMillis();
    if (ts == tts)
    { // 2 updates within the same millisecond ? (are there 2 CYCLERs?)
      return;
    }
    ts = tts;
    DbgLog.log("update","received cycle number : "+cycleNumber+" <"+status+">");
    // do something useful?  (maybe hardware IO)
  }
}
private void initializeDeviceServer()
{
  sineEqpModule = new SineEquipmentModule("SINEQM",(SineDevice[])sineDeviceSet.toArray(new SineDevice[0]));
  sineEqpModule.registerCycleTrigger(new MyCycleTrigger());
  // can alternatively be registered directly with the equipment module factory (e.g.):
  // TEquipmentModuleFactory.getInstance().registerCycleTrigger(new MyCycleTrigger());
  // Other iniatialization stuff omitted ...
  // ...
}
```

TINE 4.0.9: Cycle Triggers

Handling Cycle Triggers in LabView



TINE 4.0.10: up and coming …

- 1) Solving the ‘132 MB transfer problem’
  - CM_STREAM transfer doesn’t work beyond a ‘magic number’ of bytes : 132461899
  - (S. Weisse)
- 2) Implementing the ‘multi-channel array’ background logic.
  - Properties registered as multi-channel arrays being accessed ‘pro channel’
    - e.g. Vacuum Pressure, BPM positions, etc. can be obtained with a single contract instead of 300 contracts!
- 3) History calls using CF_HISTORY
  - Allow any format type to be archived and retrieved
  - Allow access to the ‘system stamp’ and ‘user stamp’ (along with the timestamp) stored with the data.
- 3) Variable length formats in structs
  - CF_STRING, CF_IMAGE, CF_SPECTRUM

- Services:
  - ENS deadweight checker
    - Periodically ping all servers and record ‘last alive’ timestamp
    - Remove ‘dead’ entries (e.g. 3 months since ‘last alive’)