TINE Release 4.0 News

(Sept 4, 2009: That was the month that was !)

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

Bug fixes & embellishments …

- VxWorks:
  - Work-around (Motorola Boards)
    - Don’t allow configuration file read-outs for “late” registered modules!
      - Potential deadlock between File Access (via RSH) and “full” socket buffer space on other sockets (server socket, globals socket).

Bug fixes & embellishments …

- Java:
  - New archive retrieval methods:

    getArchivedData(…,**boolean** extendedRange)

    - Adds 2*ArchiveHeartbeat to each end of time range
  - Check for ‘exports.csv or fec.xml’ in Equipment module constructor by default (was optional call).
    - Allows (additional) property registration via config file.
      - e.g. redirect local history call for “Temperature.HIST” to “/PETRA/HISTORY”

Bug fixes & embellishments …

- C Library:
  - Allow optional compile switches:

    -Dtine_decorated_constants

    -DTINE_DECORATED_CONSTANTS
    - re-#defines upper/lower case constants in header files to avoid ‘collisions’ with other packages
      - e.g. STL has ‘out_of_range’ as does ‘tine.h’

        tine.h: ‘out_of_range’ -> ‘tdc_out_of_range’
      - e.g. WINDOWS has ‘CF_NULL’, ‘CF_TEXT’ as does ‘tine.h’

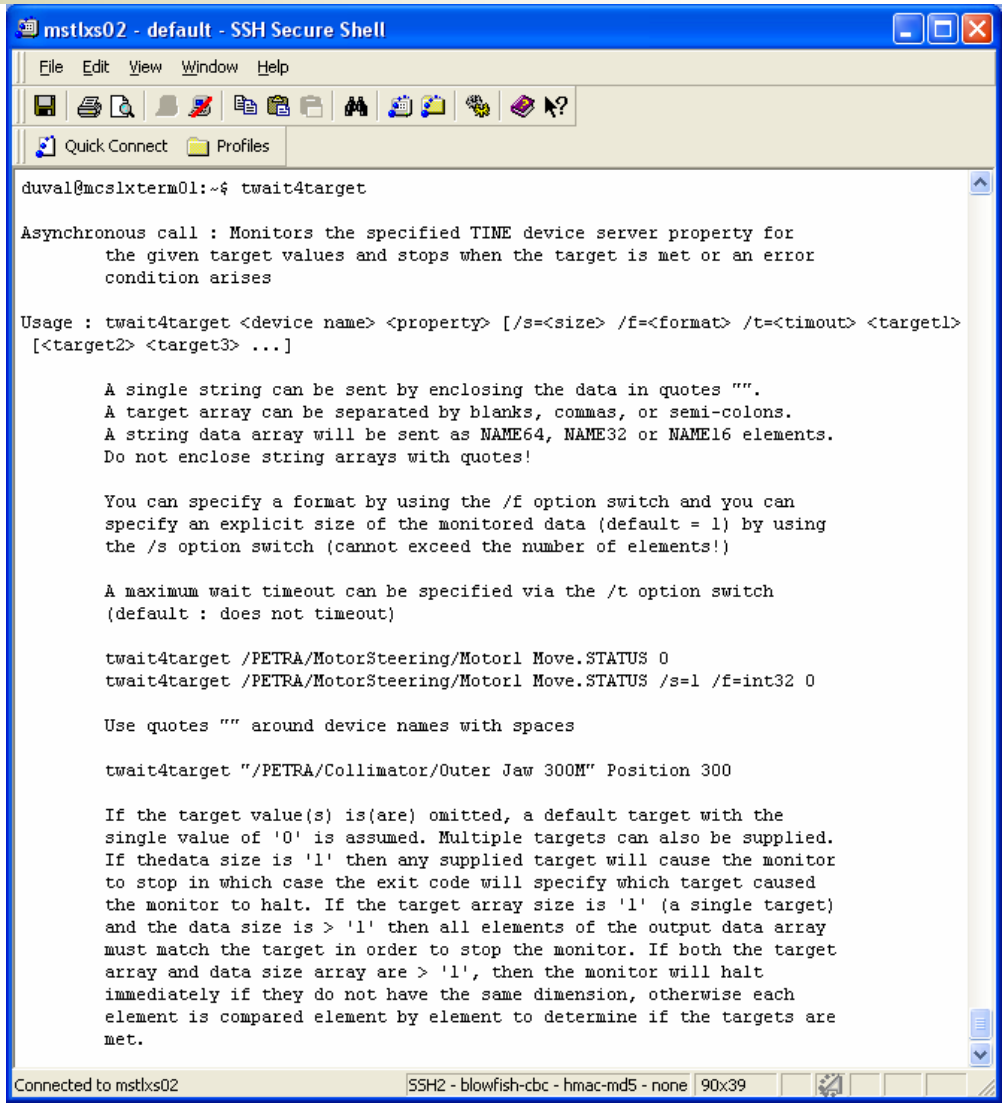
        tine.h: ‘CF_NULL’ -> ‘TDC_CF_NULL’

Bug fixes & embellishments …

- C Library:
  - UNIX builds: tcycler.c added to libtinemt.so
    - parallel to tine32.dll (i.e. tinemt.dll)
    - includes SystemSetCycleTimer() for use in .NET and python.
  - SetSizeDeviceCapacity()
    - allows post-initialization increases of the registered device space
      - if devices are added at runtime and the registered capacity needs to increase (GENS needs this)
  - SetSystemSchedulePropertyLazy() available for possible use of scheduling within ISRs where a ‘quick’ exit is required.
    - marks the property for scheduling for the next ‘normal’ pass of the scheduler (~Clock Tick)

Bug fixes & embellishments ...

- Scripting: twait4target()



Bug fixes & embellishments …

- Release 4.0.11
  - Next will be 4.1.0
  - Remaining: including non-fixed-length format types (CF_STRING, CF_IMAGE, CF_SPECTRUM) in structures and archive.
- But, as of this release:
  - Enforced Multi-Channel Array acquisition is in play !
    - Registered Multi-Channel Array properties require contracts to access the entire multi-channel array

Bug fixes & embellishments …

- How it works:
  - Client asks for 1 element of multi-channel array
  - Handshaking takes place:
    - Server informs client of size of array and which index refers to the desired element.
  - Multiple links (e.g. get all 500 vacuum pumps individually) collapse to a single link, with minimal impact on the server.
  - What could go wrong?

Bug fixes & embellishments …

- What if the property is registered as a multi-channel array of length “1”?
- Handshaking:
  - Client asks for 1 element
  - Server says, “No ask for all of them”
  - Client asks for 1 element
  - Etc.
- Server-side bug repaired!
  - Handshaking only takes place if registered array length > 1 !

Bug fixes & embellishments …

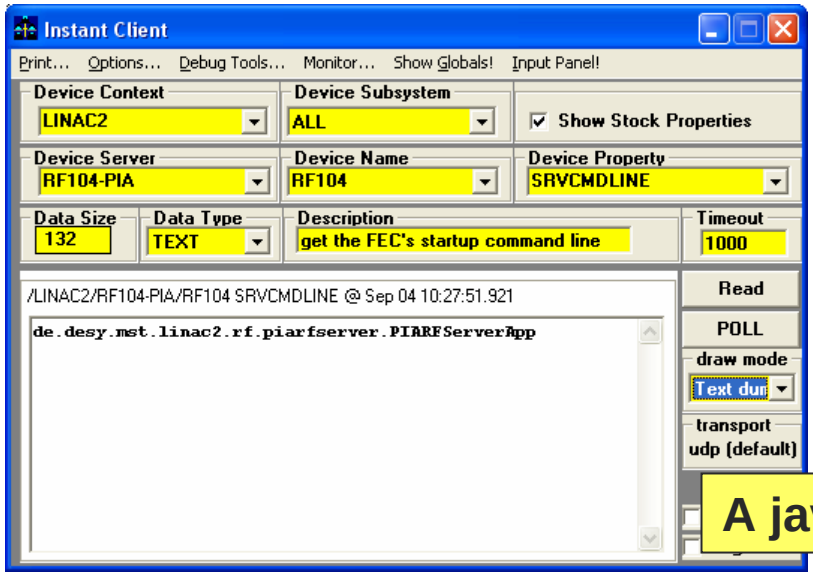
- MS-DOS
  - Now compiles in Large Memory Model
  - Uses compile switch
    - –DSMALL_TINE_LIB
  - Largely tests the consistency of a build with this compile switch.

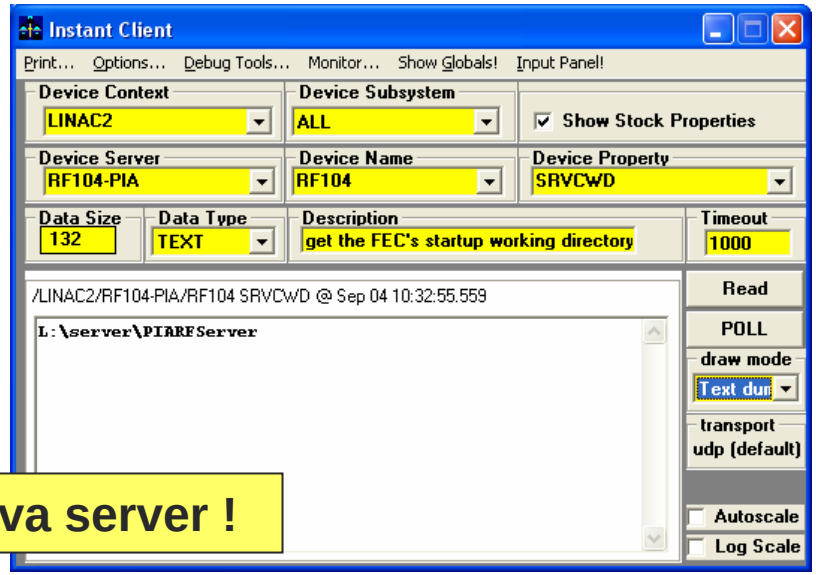
New Features (cmd line)

- Determining the command line and the working directory from the TINE library.
  - Straightforward in WINDOWS and UNIX
  - Difficult in Java
    - Can sometimes discover which class called Main.
    - How to determine the startup parameters and JVM switches ?
  - Difficult in VxWorks
    - Can discover the Task which launched SystemInit().

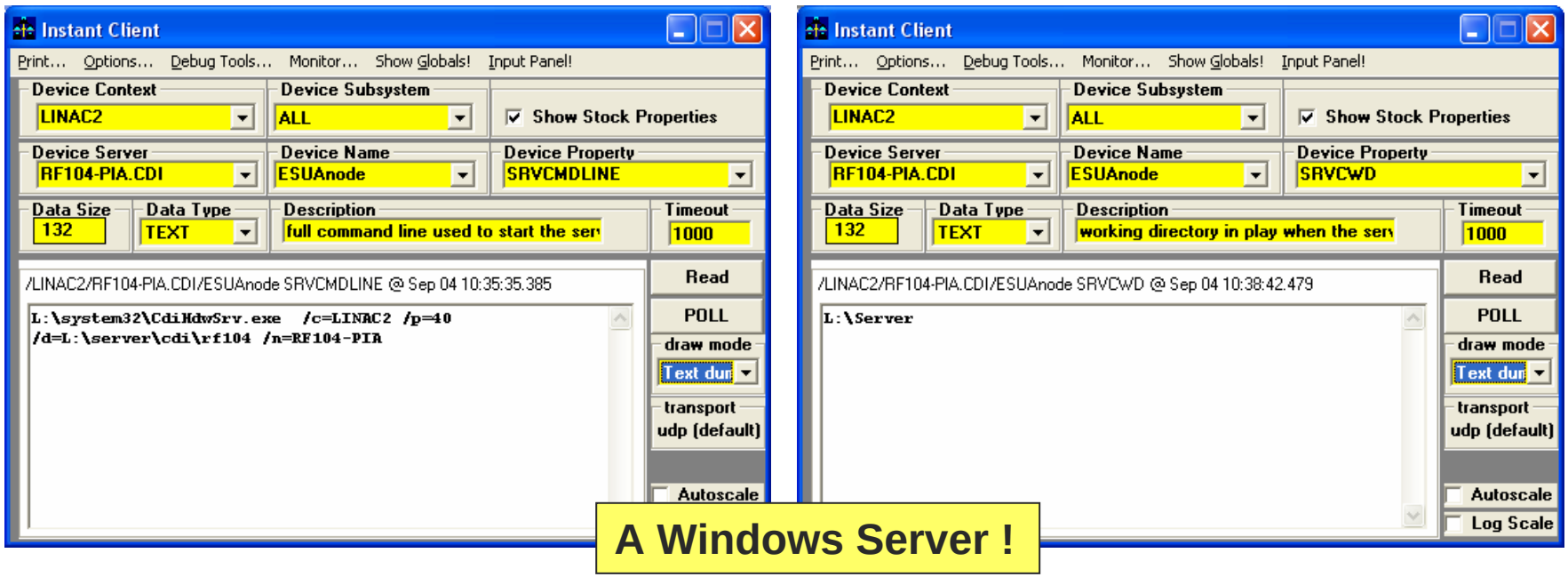
New Features (cmd line)

- New Stock Properties
  - “SRVCMDLINE” (command line)
  - “SRVCWD” (current working directory)



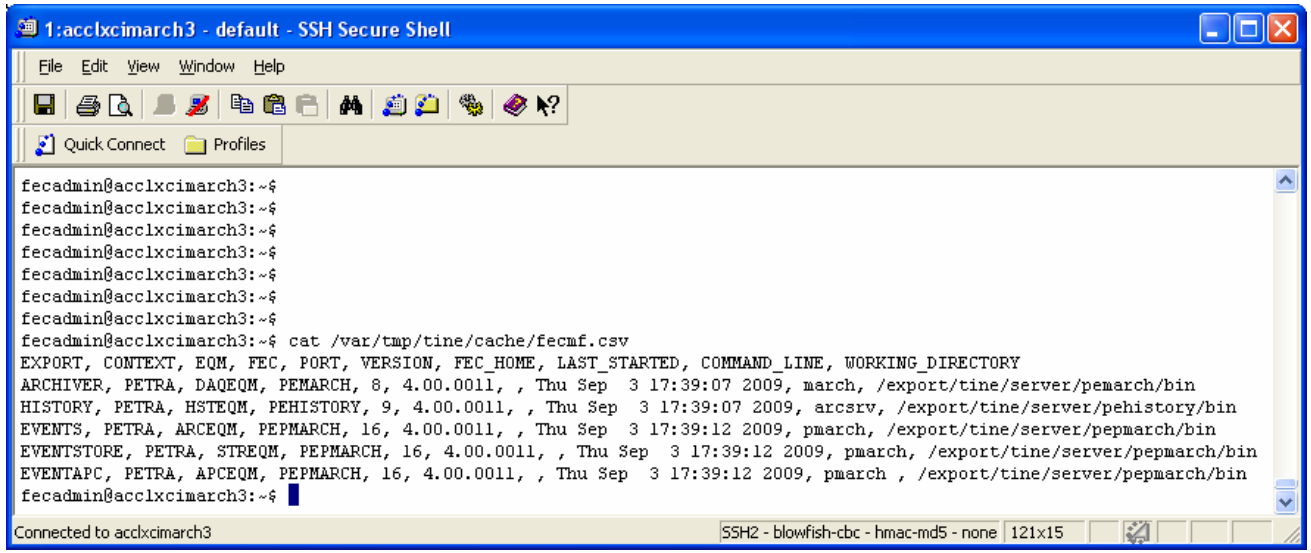


New Features (cmd line)



New Features (cmd line)

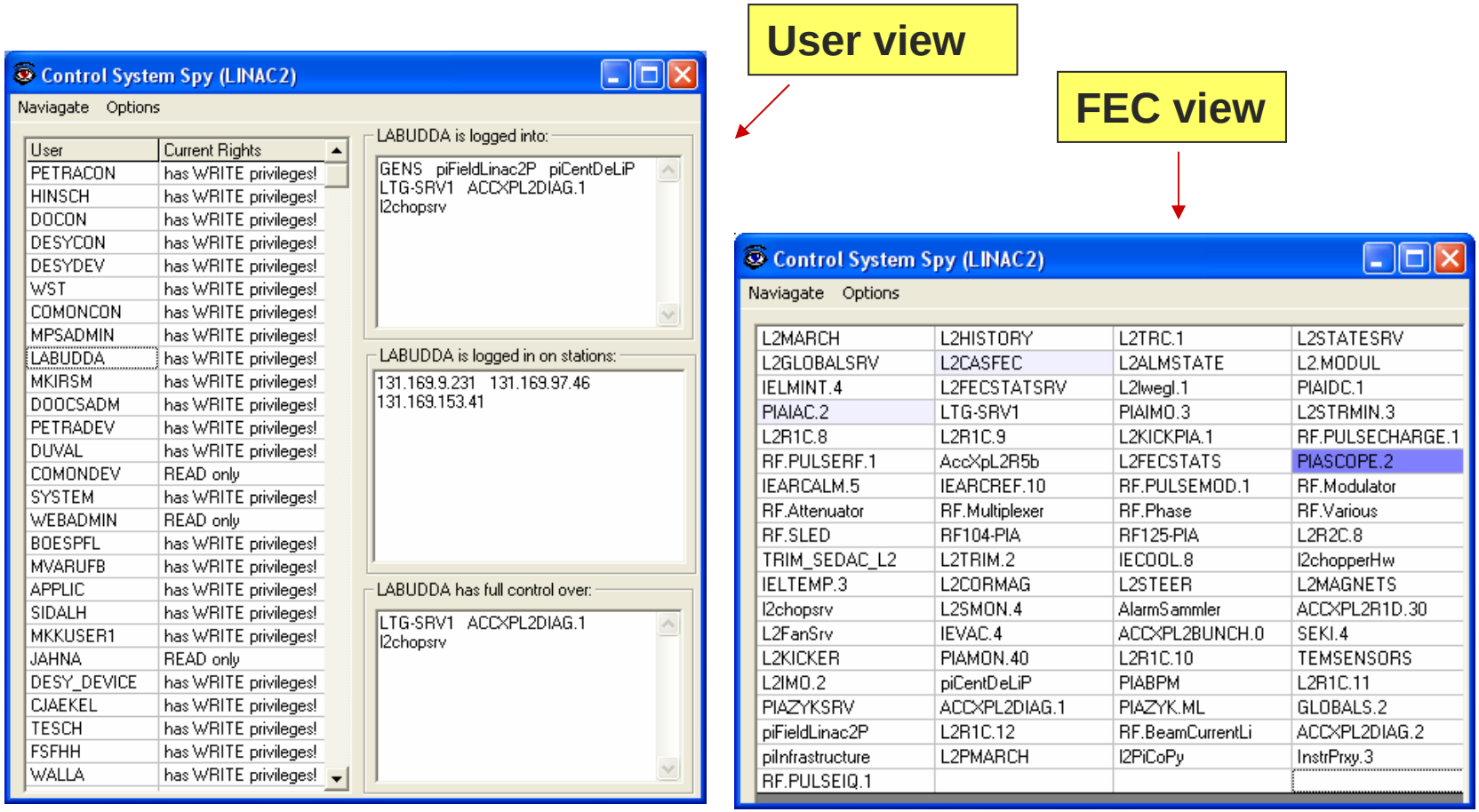
- New information found in the server manifest (fecmf.csv)
  - SystemDrive:\tine\cache\
  - /var/tmp/tine/cache/



New Features (env)

- New Environment Variables
  - FEC_LOCATION
    - e.g. set FEC_LOCATION=Bldg 20 Rack 2B
    - Supersedes input from config file or API
    - If set should be valid for the location of anything running on the host !
  - <EQM>_SUBSYSTEM
    - e.g. set GRAEQM_SUBSYSTEM=VIDEO
    - Supersedes input from config file or API

Central Services: CSSPY



.NET and mono

- Minimal client and server interface available for .NET applications !

e.g. VB.NET client

```
Public Sub updatePlot()
  Dim i As Integer
  bunProfCurve.Clear()
  For i = 0 To 78
    dataY(i) = currBuffer(i)
    bunProfCurve.Add(dataY(i))
  Next
  bunchProfile.Refresh()
End Sub
Public Sub lcb(ByVal lnk As tine.TLink)
  If bunchProfile.InvokeRequired Then
    bunchProfile.Invoke(New MethodInvoker(AddressOf updatePlot))
  Else
    updatePlot()
  End If
End Sub
Dim cb As TLink.TLinkCallback

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
  Dim id As Integer
  cb = New TLink.TLinkCallback(AddressOf lcb)
  tdt = New TDataType(currBuffer)
  tnull = New TDataType(IntPtr.Zero)
  currLink = New TLink("/PETRA/BunchScope/Bunch-1", "BunchTrace.SCH", tdt, tnull, tine.Access.CA_READ)
  id = currLink.Attach(tine.Modes.CM_DATACHANGE, cb, 1000)
End Sub
End Class
```

.NET and mono

e.g. VB server (some declarations)

```
Imports System.Runtime.InteropServices
Imports time
Public Class Form1
  <StructLayout(LayoutKind.Sequential, Pack:=1, CharSet:=CharSet.Ansi)> _
  Public Structure SineInfo
    Public amplitude As Single
    Public frequency As Single
    Public noise As Single
    Public phase As Single
    Public numberCalls As Integer
    <MarshalAs(UnmanagedType.ByValArray, SizeConst:=64)> _
      Public description() As Char
  End Structure
  Private Const PRP_SINE = 1
  Private Const PRP_AMPLITUDE = 2
  Private Const PRP_FREQUENCY = 3
  Private Const PRP_PHASE = 4
  Private Const PRP_NOISE = 5
  Private Const PRP_INFO = 6
  Private Const NUM_DEVICES = 10
  Private Const NUM_VALUES = (1024 * 8)
  Dim sinbuf(NUM_DEVICES - 1, NUM_VALUES - 1) As Single
  Dim tts As TTaggedStruct
  Dim sineInfoTable(NUM_DEVICES - 1) As SineInfo
  Dim myeqm As TEquipmentModule

  Public Function sineqm(ByVal dev As String, ByVal prp As String, ByVal dout As TDataType, ByVal din As T

    Dim fval As Single
```

.NET and mono

e.g. VB server (some initialization)

```
Public Sub sinbkg()
  Dim i As Integer
  For i = 0 To NUM_DEVICES - 1
    Call updateSine(i)
  Next
End Sub
Public Sub sinini()
  Dim i As Integer, k As Integer
  Dim dsc As String
  For i = 0 To NUM_DEVICES - 1
    sineInfoTable(i).amplitude = 256
    sineInfoTable(i).frequency = 1
    sineInfoTable(i).phase = 0
    sineInfoTable(i).noise = 5.0
    ReDim sineInfoTable(i).description(63)
    dsc = "Sine device " + Str(i) + " at your service"
    For k = 0 To Len(dsc) - 1
      sineInfoTable(i).description(k) = dsc.Chars(k)
    Next
  Next
End Sub
Public Sub sinexi()
  Console.WriteLine("server stopping")
End Sub
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

  TKernel.InitializeServer()

  tts = New TTaggedStruct(sineInfoTable)

  myeqm = New TEquipmentModule("", "SINEQM", 0, AddressOf sineqm, AddressOf sinini, AddressOf sinbkg, 1000, AddressOf

  TKernel.StartServices()

End Sub
End Class
```

.NET and mono

```
Public Function sineqm(ByVal dev As String, ByVal prp As String, ByVal dout As TDataType, ByVal din As TDataType, ByV

  Dim fval As Single
  Dim mcarray(NUM_DEVICES) As Single
  Dim cc As Integer, i As Integer
  Dim devnr As Integer
  Dim prpid As Integer

  devnr = eqm.GetDeviceNumber(dev, prp)
  prpid = eqm.GetPropertyId(prp)
  Select Case prpid
    Case PRP_SINE
      cc = Errors.illegal_read_write
      If ((acc And Access.CA_WRITE) = 0) Then
        cc = dout.PutData(sinbuf, devnr, NUM_VALUES, 0)
        sineInfoTable(devnr).numberCalls = sineInfoTable(devnr).numberCalls + 1
      End If
    Case PRP_AMPLITUDE
      If (din.GetDataArrayLength() > 0) Then
        If ((acc & Access.CA_WRITE) <> Access.CA_WRITE) Then
          cc = Errors.illegal_read_write
        End If
        cc = din.GetData(fval)
        If cc <> 0 Then Exit Select
        sineInfoTable(devnr).amplitude = fval
      End If
      If (dout.GetDataArrayLength() > 0) Then
        For i = 0 To NUM_DEVICES - 1
          mcarray(i) = sineInfoTable(i).amplitude
          cc = dout.PutData(mcarray, NUM_DEVICES, devnr)
          If cc <> 0 Then Exit Select
        Next
      End If
    Case PRP_FREQUENCY
      If (din.GetDataArrayLength() > 0) Then
        If ((acc & Access.CA_WRITE) <> Access.CA_WRITE) Then
```

e.g. VB server (equipment module)

.NET and mono

- C# similar to java (and vb.NET with different syntax)
- Mono demo ….