TINE Release 4.0 News

(Aug 7, 2009: That was the month that was !)

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

Bug fixes & embellishments …

- **LabView:**
  - e.g lvTineSrvPushSingle.VI
    - Sets the data time stamp to time of the ‘push’
    - Certain cases: want to be able to directly assign the timestamp in the call (e.g. reference trace, etc.)
    - Solution: allow the ‘prpSchedule’ flag to pass a designated timestamp
      - If prpSchedule < 0x10 then schedule to all subsystems (normal case)
      - If prpSchedule >= 0x10 && <= 0x1000 then look at ‘scope’ (0x10 = CA_HIST, 0x20 = CA_ALARM, 0x200 = CA_NETWORK)
      - If prpSchedule > 0x1000 then DON’T schedule but instead use the value as UTC timestamp (seconds resolution, not milliseconds) to assign the ‘pushed’ data.
  - The other lvTineSrvPushXXX VIs follow the same logic.
  - Note in passing: Engineer from Westinghouse Rail Systems (UK) evaluating TINE and using LabView 7.1. Ideally he would ask questions to the tineforum but this is not yet possible => emails with a ‘CC list’. But those with LabView expertise PLEASE get a tineforum account and subscribe to the LabView topics!

Bug fixes & embellishments …

- VxWorks:
  - When spawning a task:
    - Stay with the same priority (unless you have a good reason not to) and …
    - kernelTimeSlice(sysClkRateGet()/100);
      - Otherwise there is NO round-robin preemption!

Remarks: Scheduling Data

- TINE Kernel has a scheduler which calls registered contracts according to the designated polling interval.
  - 1 or more clients with contract at ‘1000’ msecs -> contract scheduled at 1000 msec intervals
    - multiple contracts synchronized on the same time boundary (delivery packing)
  - 1 client with contract at 2000 msec + client with contract at 3000 msec -> contract scheduled at 1000 msec intervals
- Acquisition mode CM_TIMER (CM_POLL)
  - Client receives data at the requested polling rate (latency no greater than the polling interval)
- Acquisition mode CM_DATACHANGE (CM_REFRESH)
  - Checks data on the server at the requested polling rate
  - Client receives data upon data change (zero-tolerance but can also apply a client-side tolerance to suppress unwanted notifications).
  - (Watchdog link provides immediate notification if the server goes down).

Remarks: Scheduling Data

- The server programmer can also ‘call’ the scheduler to effect immediate delivery of a given property to any listening clients.
  - SystemScheduleProperty()
    - Can be called regularly (e.g. video system with external trigger of frame grabber)
    - Can be called when ‘conditions are ripe’ (e.g. the alarm system notices a new alarm)
  - Note:
    - The kernel’s scheduling rules are still in force!
    - But the ‘time of last delivery’ is reset following a call to SystemScheduleProperty().
  - If property “ABCD” is ‘scheduled’ then ANY client with ANY persistent link to “ABCD” will receive a notification regardless of requested polling rate!

Remarks: Scheduling Data

- What if the client ONLY wants the ‘scheduled’ events?
  - Attach a Link with the default polling interval of ‘1000’ msec
  - Use acquisition mode CM_EVENT

Remarks: Archive Retrieval

- Case 1: quasi-dynamic multi-channel arrays (e.g. PT100 temperature sensors)
  - Note: Archive system requires ‘fixed-length’ data records for rapid data retrieval.
  - More sensors this week than there were last week: What to do?
    - Either disable last weeks archive records and make new ones for this week (ugh!, please NO!) or
    - Use a fixed length records with enough capacity to handle later additions (YES!).
  - Proposed systematics to improve display:
    - Empty elements should have a ‘device name’ of e.g. ‘Reserved#88’, ‘Reserved#89’, … etc. filling out the reserved capacity. KEY: device name begins with ‘Reserved’.
    - Applications such as the Multi-Channel Analyzer will trap device names beginning with ‘Reserved’ and collapse the array prior to display

Remarks: Archive Retrieval

- Case 1: quasi-dynamic multi-channel arrays (e.g. PT100 temperature sensors)
  - e.g. 841 Magnet temperature sensors
  - Register property “Temperature” to deliver 900 values
  - Register device names for devices 842 -> 900 with ‘Reserved#842’, etc.
  - The ARCHIVER will then always acquire a record length of 900 float values.
    - Note: a change in the device list will automatically register with the central archiver!

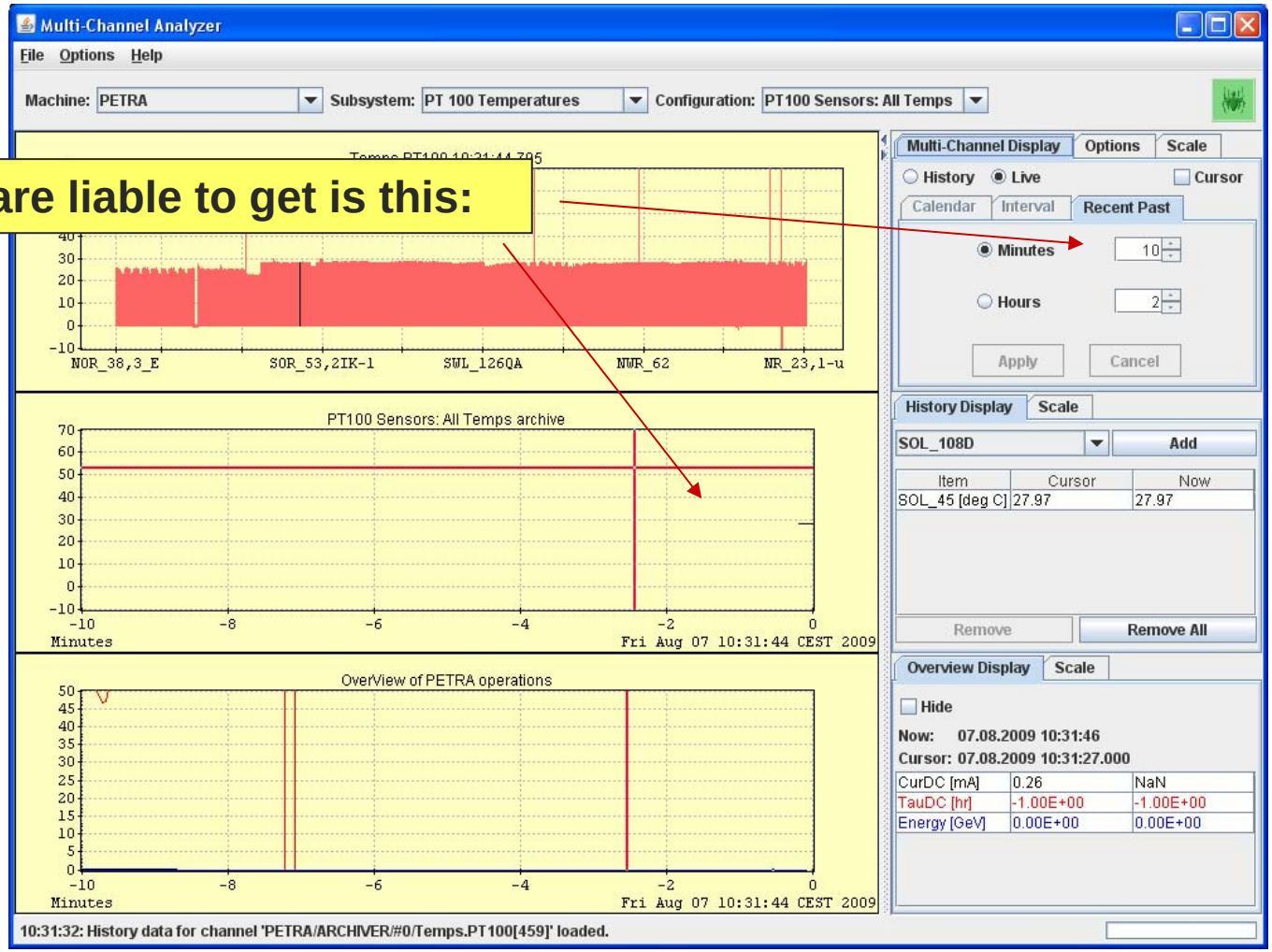
Remarks: Archive Retrieval

- Case 2: Time range
  - Archive call for a value over a time range provides ‘startTime’ and ‘stopTime’.
  - Data returned are those points found between

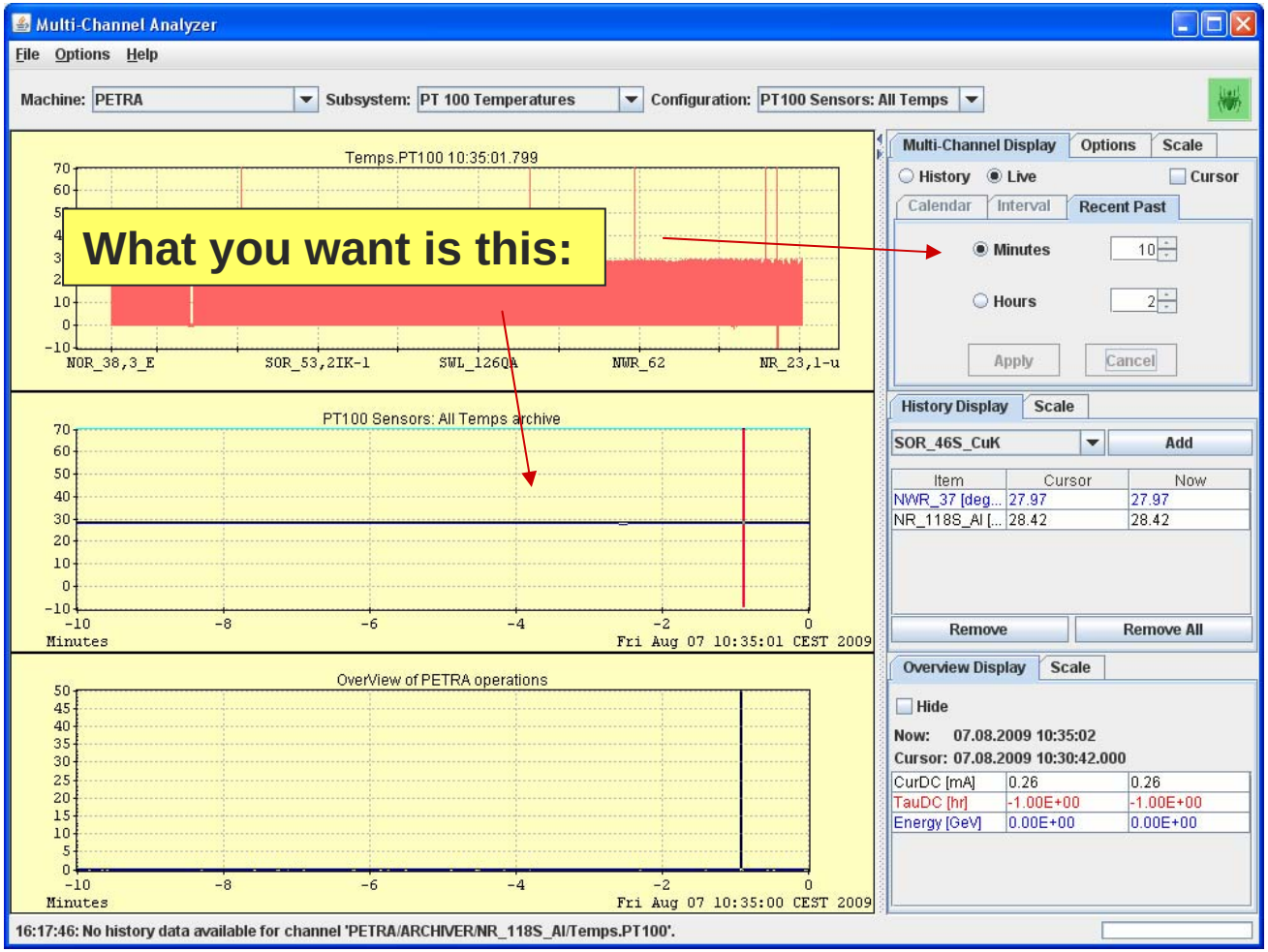
    t >= startTime AND t <= stopTime

Remarks: Archive Retrieval

What you are liable to get is this:



Remarks: Archive Retrieval



Remarks: Archive Retrieval

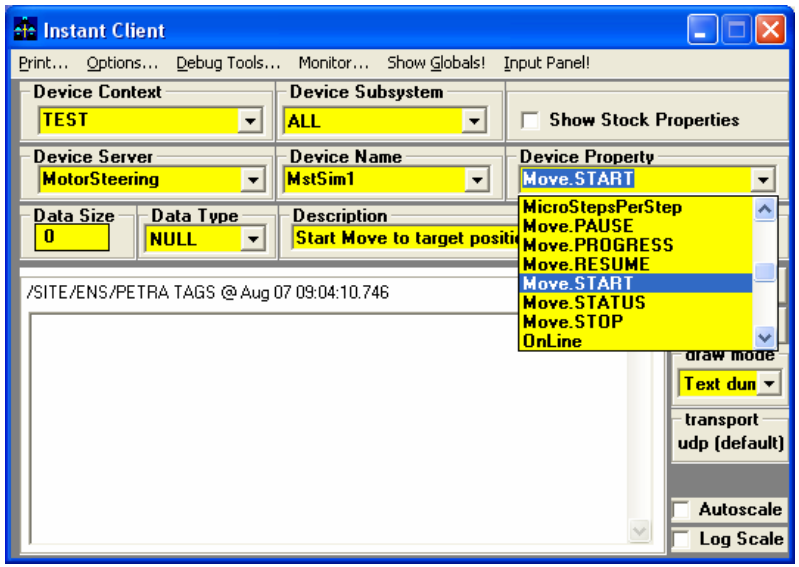
- Case 2: Time range
  - How to achieve this?
    - Archive heartbeat is typically 15 minutes
      - As long as there are active data with a changing timestamp then a data record is archived every 15 minutes.
    - Ask for a time range of

      (startTime – heartbeat) to

      (stopTime + heartbeat)
    - Obtains a point outside the display region
  - Question:
    - Is this entirely up to the application? OR
    - Should the API call to getArchivedData do this automatically?

Proposal: twait4target()

- How to manage server ‘processes’ with scripts?
- Typical Process: Moving a motor to a target position
  - takes some amount of time
  - Follows the ‘process’ paradigm



- **Start Process with property ‘Move.START’**
- **Monitor Process with property ‘Move.STATUS’**
- **When ‘Move.STATUS’ returns ‘0’ then the process is finished.**

Proposal: twait4target()

- This is easy to code with a High-Level Language.
- How to do it with a script?

  tsend /TEST/MotorSteering/MstSim1 Move.START

  - will start the move, but we want to avoid ‘polling’ the status in a script!

Proposal: twait4target()

- Need a single call which will only complete if the ‘move’ has finished or an error occurs.
- e.g.

  twait4target /TEST/MotorSteering/MstSim1 Move.STATUS /value=0

- Possible script:

  tsend /TEST/MotorSteering/MstSim1 Move.START
  twait4target /TEST/MotorSteering/MstSim1 Move.STATUS /value=0
  if "%errorlevel%" == "1" …. (something didn’t work …)