



The SNS General Time Timestamp Driver

Sheng Peng & David Thompson



Why create a new timestamp driver?



- **SNS has a timing system that supports hardware timestamps and we use those timestamps to correlate data.**
- **Existing timestamp support (in drvTS) calls the same time source for event timestamps (in record processing) as well as for scan task and I/O task scheduling.**
- **No time source is 100% reliable, we wanted to provide for selection and use of an alternate source if the primary source is unavailable.**
- **The built in EPICS time interface in iocClock.c does not provide for redundancy if a primary time source fails.**

The problem



- At SNS we noticed on at least two occasions that some of our IOCS were not scanning records and that some vxWorks tasks had unusual delays in the task display. This required some expensive reboots.
- The second time that we saw it happen we were ready and captured enough data to track the problem down to the GPS receiver in the timing system.
- GPS keeps time in 1024 week “epochs”, our receiver jumped to an epoch far into the future and stayed there for a couple of minutes. When it jumped back many of our IOCs had drvEtherlp tasks in progress processing data and these got caught by turning a large negative time jump into a large delay. **(The receiver was RMA-d for a firmware update and has behaved since.)**
- During normal timing system operation noise can be experienced on the RTDL causing transient problems.

The problem



- **Even NTP is not perfect: We had one workstation that had a bad clock. It ran at about half time and made 15 second jumps to catch up. This caused it to mis-report “beacon anomalies”**
- **Time itself is not monotonic! (UTC time at least)**

The mean solar day was “exactly” 86400 SI seconds in 1820, but in recent years it has been as much as 86400.002 seconds. A leap second is added every few years to adjust UTC. Since the earth is running slower than UTC then UTC must be set backwards by a second. The difference is about _ second now.

(<http://tycho.usno.navy.mil/leapsec.html>)

What is generalTime?



- It is a time source manager
- It always selects the “best available” time source, depending on whether an event time or the current time is requested.
- It supports hardware time sources i.e. timing system.
- It provides two default time source: NTP & Ticks via `clock_gettime()`.
- Can be extended by just adding time drivers as normal epics drivers to the system.
 - » We are planning a driver to support on-board clock chips for full stand alone IOC

Best available is the one that has the highest priority and is not returning ERROR. The driver must be able to know it is wrong.

API (extending iocClockRegister)



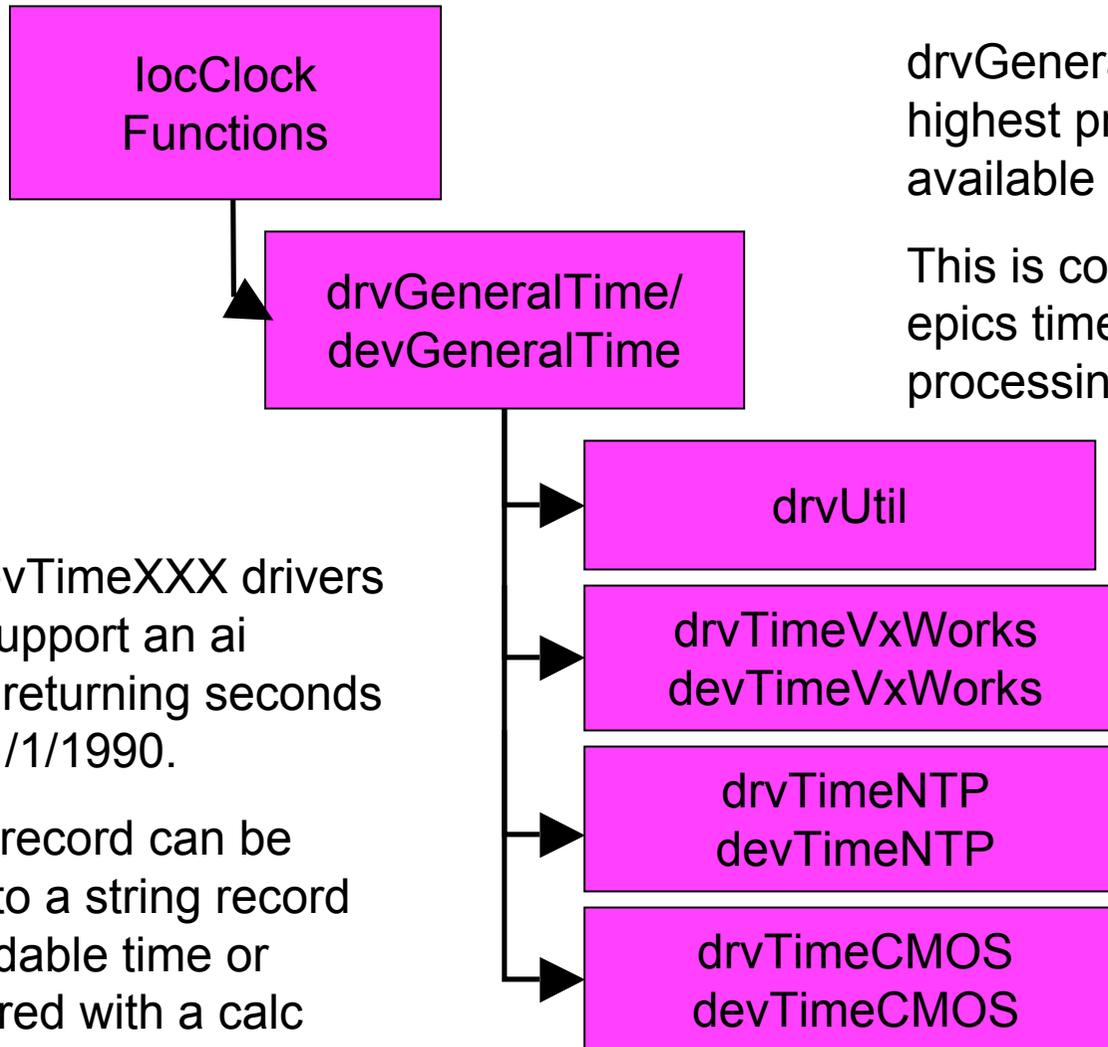
- **General time calls: iocClockRegister().**
- **A time driver calls: generalTimeTpRegister().**
Arguments:
 - » **Tcp_priority: Priority of current time function.**
 - » **getCurrent: Pointer to current time function in your driver.**
 - » **syncTime: A pointer to a function that general time calls to synchronize the time in the driver if needed.**
 - » **Tep_priority: Priority of event time function.**
 - » **getEvent: The event time function in your driver.**

API (Continued)



- **When an EPICS API function requests a current time or an event time, general time calls registered time functions in order of priority.**
- **When a time driver returns OK the search for the time is stopped and that time is returned to the Epics system caller.**
- **If the time function returns ERROR then the next lower priority driver is called.**
- **The lowest priority time is the system time which should always be available, even if it is wrong.**

Details



drvGeneralTime returns the highest priority time source available

This is completely transparent to epics time and record processing functions.

The devTimeXXX drivers must support an ai record returning seconds since 1/1/1990.

The ai record can be linked to a string record for readable time or monitored with a calc record for differences.

General time features:



- **General time is an Epics driver with device support.**
- **The drvGeneralTime report function reports status and a list of time providers.**
- **Device support provides string and ai record support for status of current time providers for both event time and current time lists.**

Utility module (SNS Hardware)



- No longer uses drvTS.
- USES generalTime interfaces.
- The Utility driver has been improved because it no longer has to provide time when the RTDL is bad. The old driver had to implement NTP to provide back up time.
- Returns error if time is bad, so general time can pick next best.
 - » Timing master reboot causes zero timestamp errors in generalTime
 - » The timing master will set a flag on the RTDL when it does not know the time for some reason. This will be passed up to general time.
- The driver has had internal changes to detect and track VME and timing system errors.

Conclusions



- **We would like to see general time replace iocClock in builds for real time OS.**
- **General time is written using OSI calls.**
- **Works the same as iocClock when hardware time is not available, even if hardware time is configured and installed.**
- **drvTimeNtp is derived from iocClock.c**
- **Probably most useful on vxWorks where the basic system time calls do not use NTP.**