Minutes of LHC-CP Link Meeting 25

Subject : LHC Controls Project

Date : 18th June, 2002

Place : 864-2-B14

Participating Groups :

LHC-ACR	no representative,
LHC-ECR	no representative,
LHC-IAS	J. Brahy, C-H Sicard,
LHC-ICP	A. Hilaire, F. Rodriguez-Mateos
LHC-MMS	no representative,
LHC-MTA	no representative,
LHC-VAC	R. Gavaggio,
PS-CO	B. Frammery, F. DiMaio,
SL-AP	no representative,
SL-BI	no representative,
SL-BT	E. Carlier,
SL-CO	A. Bland,
SL-HRF	E. Ciapala,
SL-MR	R. Billen,
SL-MS	no representative,
SL-OP	M. Lamont,
SL-PO	Q. King,
ST-MA	no representative.

Others : R. Lauckner (Chair),
M. MarcZukajtis, M. Peryt (Logging Project),
B. Puccio (Machine Interlocks),
P. Gayet (Core Team),

M. Tyrrell (Alarm Project).

Distribution : Via LHC-CP website: **http://cern.ch/lhc-cp**
Notification via: **lhc-cp-info@cern.ch**

Agenda :

1. Matters arising from Previous Meeting	
2. LHC-CP News	
3. Time Stamping	A. Bland
4. FGC Gateway	Q. King
5. AOB	

1. Matters arising from Previous Meeting

R. Billen showed a summary of the global features of the Electrical Networks Monitoring and Control system. The new system is replacing the Micene system employed at LEP. Embedded controllers (SEPAM) push time stamped data up into the control system. This is currently exported via an Electrical Networks SCADA (ENS) system to a UNIX fileserver. A History Data Server will be introduced later this year based on Oracle and this will open the interface to the future LHC logging system. P. Gayet commented that it would be useful to have similar information from the Cooling and Ventilation Control System.

ACTION: R. BILLEN, P. SOLLANDER

R. Lauckner reported that A. Daneels had held a plenary meeting for clients and suppliers concerned with the LTI controls infrastructure. The meeting had provoked many exchanges of information and cleared misunderstandings. Minutes are being prepared.

After the meeting P. Sollander pointed out that ST-MA is not responsible for the implementation of any public address system for the LHC. The group was asked was for a cost evaluation of two different solutions, this has been done.

2. LHC-CP News

R. Lauckner reported briefly from the 3rd Joint Controls Project Workshop (JCOP) that took place on 5th and 6th June. Many projects are under way at CERN and other labs using PVSS which has become the most prominent feature of the JCOP. There was a report concerning a new JCOP initiative to develop the Detector Safety Systems (DSS), a project led by B. Flockhart. A collaboration with the Machine Protection team might be fruitful. There was a strong request to launch the 2nd phase of the LHC Data Interchange Project. R. Lauckner has discussed this with W. Salter and a mandate is being prepared. It will include:

- Review the report from phase 1 (now 2 years old)
- Elaborate the system requirements
- Evaluate solutions based on products already in use at CERN
- Build a pilot for the end of 2002.

This would offer a communications tools that would be ready for possible use during the QRL reception tests.

Q. King mentioned that the Interface Specification for the CTX1 card is now in the LHC baseline and is following the approval process. Many comments had been received before formal approval started so he does not expect much more feedback.

E. Ciapala reported that the Signals working group is now meeting regularly every 2 weeks. He also said that the supply of function generators from PO to RF for the LHC is being actively discussed. Clarification is still needed concerning the functionality of the LHC timing system beyond the hardware information supplied in the Engineering Specification. R. Lauckner will arrange for this topic to be covered at a future SLTC meeting.

ACTION R. LAUCKNER

R. Lauckner announced that CERN have sent a fax from T. Lagrange to ETM with a copy of the PVSS contract amendment.

The schedule and main topics for the next LHC-CP meetings are:

2/7	FGC Gateways, QPS Controls Requirements, Controls for Machine Interlocks	Q. King, H. Milcent, B. Puccio
16/7	QRL Baseline, Experience from RF test stands	A. Daneels, L. Arnaudon

3. Time Stamping A. Bland

A. Bland explained that UTC is a well-established tool for time keeping in computers with support on the operating systems of interest for LHC Control. The supporting tools in PLCs are still to be investigated. It is not subject to daylight saving or longitudinal variations. However it does suffer occasional discontinuities due to the addition of leap seconds. GPS time differs from UTC because it does not keep synchronism with the Earth's rotation via leap seconds. However GPS receivers can supply UTC.

ANSI C includes the time() and ctime systems calls. The time() function returns a 32 bit integer of elapsed seconds since 1st January 1970, ignoring leap seconds. The ctime() call will convert this to a string expressing time of day applying time zone and daylight savings offsets. These results are the same on UNIX, LynxOS and Windows 2000. More precise times are supported in different formats on these systems. Java provides a 64 bit long with the elapsed time in milliseconds since 1st January 1970.

Oracle 8i has the DATE data type to store date and time up to second resolution. There is no support for daylight saving or time zone. Users submit and receive dates as strings although simple arithmetic can be used to convert ANSI C 32 bit integers. UTC storage and retrieval is straightforward. Oracle 9i introduces the TIMESTAMP data type which is an extension of DATE to provide nanosecond resolution. Another two related data types are introduced in 9i with nanosecond resolution as well as with time zone support. The TIMESTAMP data type is interesting for LHC data.

Excel documentation explains conversion between dates and numbers. It is then easy to convert the result of an ANSI C 32 bit integer to an Excel date when you know the offset from local time. However there is no support for daylight saving or time zone.

A. Bland explained that computers keep time with a precision of 1-10 milliseconds using the NTP. He pointed out that a hardware solution is required to achieve 1 millisecond or better accuracy. NTP is available on UNIX, LynxOS and Windows2000. NTP is not (yet) a standard for PLCs.

He concluded with the recommendation that UTC should be used for time stamping. He mentioned various string and binary formats for representation. If precision information is to be included then he recommended the NTP structures. Time stamping should be done at source using hardware solutions for accuracy below 1 millisecond and NTP for general synchronisation.

R. Billen commented that the handling of date and time in Java is a nightmare with a large amount of information available on the Web pointing out the pitfalls. He was also of the opinion that, in the framework of the LHC Controls, the storage of UTC time in Oracle is more advisable than local time.

Asked about wrapper functions for LHC user's A. Bland stressed that he would prefer to supply recommendations rather than embark on supporting solutions with the limited resources.

4. FGC Gateways Q. King

This presentation was postponed until the next meeting

5. AOB

There was no other business.

Long-Term Actions	**People**
Underground Control Rooms requested	R. Lauckner
Establish Post-mortem sub-project	R. Lauckner
Clarify Middleware Services to be used by LHC-CP	Core Team

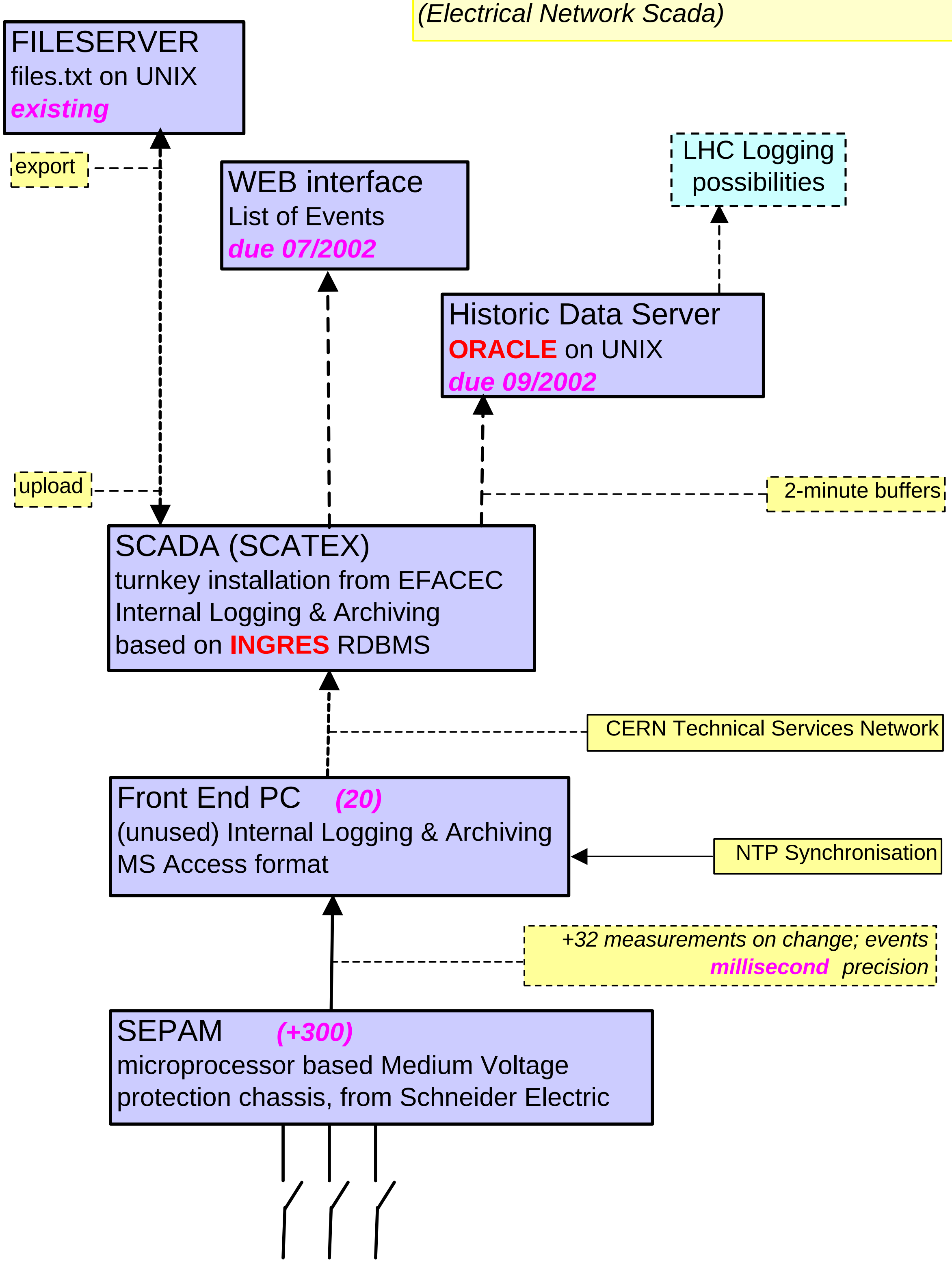
Reported by R. Lauckner

From discussion with:
Soren POULSEN
Georges BURDET

ENS DATA

R. BILLEN
14-JUN-2002

Context: *Electrical Networks Monitoring and Control is being moved from an in house system designed for LEP (MICENE, Front Ends, HP-UX) era to the ENS (Electrical Network Scada)*



FILESERVER
files.txt on UNIX
existing

export

upload

WEB interface
List of Events
due 07/2002

LHC Logging possibilities

Historic Data Server
ORACLE on UNIX
due 09/2002

2-minute buffers

SCADA (SCATEX)
turnkey installation from EFACEC
Internal Logging & Archiving
based on **INGRES** RDBMS

CERN Technical Services Network

Front End PC ***(20)***
(unused) Internal Logging & Archiving
MS Access format

NTP Synchronisation

+32 measurements on change; events
millisecond *precision*

SEPAM ***(+300)***
microprocessor based Medium Voltage protection chassis, from Schneider Electric

Timestamping Revisited

Presentation to the LHC-CP

Basically a repeat of the talk given at the Third LHC-CP Workshop (http://lhc-cp.web.cern.ch/lhc-cp/Workshop/2002-03/Slides/Session2/9ab.pdf) concentrating on why UTC is good for us plus more details on Java and Oracle from Ronny Billen, NTP and PLCs

Alastair Bland, 18th June 2002

Summary

- UTC, leap seconds
- GPS
- Time Formats:
  - Text
  - Unix
  - Windows
  - Java
  - Oracle
  - Excel
- How Computers keep time
- Network Time Protocol
- PLCs
- Recommendations

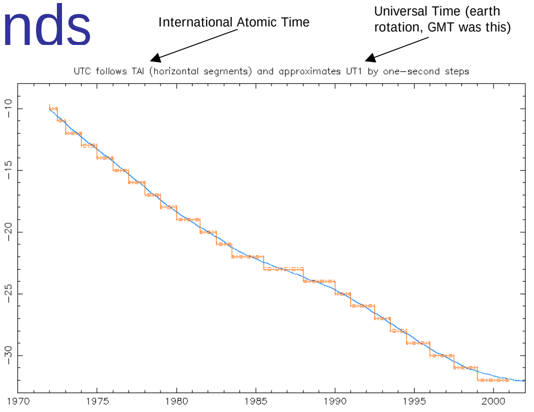
UTC

- Coordinated Universal Time (UTC) is the current term for what was commonly referred to as Greenwich Meridian Time (GMT). Zero hours UTC is midnight in Greenwich England, which lies on the zero longitudinal meridian.
- UTC is frequently used in computers for counting time passing as it is the same at any point on Earth.
  => so **all** correctly set up computers using this have the **same** UTC
  => For Unix/Linux/LynxOS, everywhere on Earth, the same time is output with:
  `date –u`
- It does not hop back and forth twice a year in Geneva
- People do not like looking at it – except in Morocco!
  - So tools are provided by most computer systems for people to input or output a date in local time

---

Leap Seconds

- A leap second is added about every 18 months
- The leap seconds are effectively forgotten so knowing the UTC time now you can calculate how many seconds have passed since, for example the 1st of January 1970 UTC, without knowing how many leap seconds have been added
- The decision to introduce a leap second in UTC to deal with the poor time keeping of the Earth is the responsibility of the International Earth Rotation Service at the Observatoire de Paris.



- No leap second will be introduced in UTC on 30 June 2002
- All leap seconds so far have been positive and in December or June

GPS (1)

- Global Positioning System (GPS) receivers give accurate time
  - 1 microsecond resolution and precision or better
- Lots of hardware solutions
  - Standalone (serial I/O plus pulse per second), VME bus, PCI bus
  - With IRIG-B output for distribution over long distances (for example in tunnels)
  - With freeze input for 100% hardware based time stamping
  - Pulse per second and programmable pulse outputs
  - Cable delay correction
- Already used in
  - PS: fed into the Timing System via the Master Timing Generator
  - SPS: Network Time Protocol Distribution to Work stations and Front Ends
  - LEP: time stamping Beam Dump requests and RF diagnostics (via IRIG-B)
  - String2: time distribution over WorldFIP to Power Converters and Magnet Protection systems
- Lock on to GPS satellites after power on can take many minutes

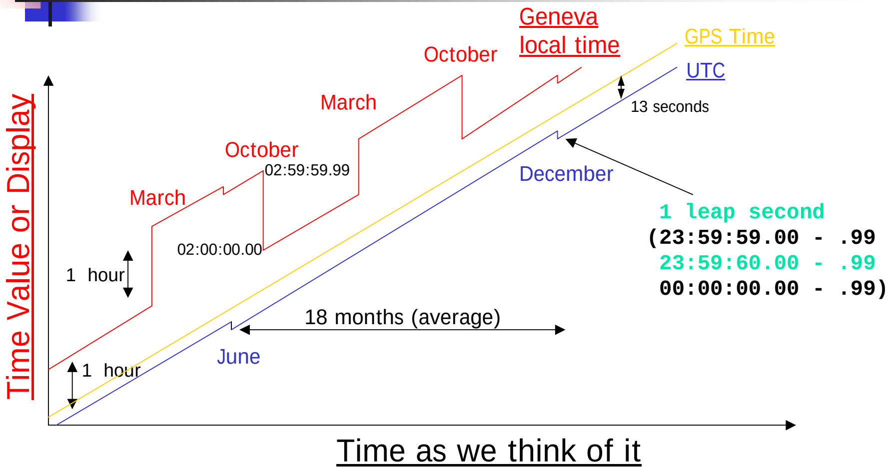
---

GPS (2)

- A GPS receiver normally provides
  - GPS time (does not include leap seconds since 1980)
  - UTC
  - Registers where local time offset can be set, under external software control. To be avoided!
- There will be one source feeding the SPS / LHC MTGs and the NTP distribution
  - However there will be other GPS receivers in the outlying buildings to check that the Timing delays have been correctly set
  - This implies that the time should be sent early to all the outlying buildings so it arrives on time.

We use this

UTC, local time and GPS



1 leap second
(23:59:59.00 - .99
23:59:60.00 - .99
00:00:00.00 - .99)

Not to scale!

Time Formats (Text)

- International Standard ISO 8601 specifies numeric representations of date and time. A long document as usual …
- In summary dates should be like:

YYYY-MM-DD

where YYYY is the year in the usual Gregorian calendar, MM is the month of the year between 01 (January) and 12 (December), and DD is the day of the month between 01 and 31.

- Times should be:

hh:mm:ss

where hh is hours since midnight (00-24), mm is minutes since the start of the hour (00-59), and ss is seconds since the start of the minute (00-60). If the hour value is 24, then the minute and second values must be zero. [The value 60 for ss might sometimes be needed during an inserted leap second]

Time Formats (C library)

- Time is a signed 32 bit integer giving the number of seconds since the 1st of January 1970, not including leap seconds (like our watches). Problems will occur in 2038 when it becomes negative.
- Unix, LynxOS, Windows2000 and MacOS X natively in their C libraries support the time() system call:

```
#include <time.h>
main() {
      time_t t;

      time(&t);
      printf(“%d\n%s”, t, ctime(&t));
}
```

- The output on Unix, LynxOS, Windows2000 and MacOS X (Pierre!) will be like:

```
1016645238
Wed Mar 20 18:27:18 2002
```

Time Formats (Unix)

- On Unix, LynxOS or MacOS X to get 1 microsecond theoretical resolution there is:

```
#include <time.h>
struct timeval t;

gettimeofday(&t,0);
printf(“%d.%06d\n”, t.tv_sec, t.tv_usec);
```

- Or on Unix or LynxOS for 1 nanosecond theoretical resolution:

```
#include <time.h>

struct timespec posix;
clock_gettime(CLOCK_REALTIME, &posix);
printf("%d.%09d\n", posix.tv_sec, posix.tv_nsec);
```

Time Formats (Windows2000)

- On Windows2000 it is easy to get 1 millisecond theoretical resolution:

```
#include <sys/timeb.h>
#include <winsock2.h>

struct _timeb timebuffer;
_ftime( &timebuffer );
printf(“%d.%03d\n”, timebuffer.time, timebuffer.millitm);
```

Time Formats (Java)

- Java has a lot of support for dates. The call that gives results consistent with Unix and Windows2000 is in java.lang.System:

```
public static long currentTimeMillis()
```

- It returns the current time in milliseconds in a 64 bit signed long. Dividing by 1000 gives the the number of seconds since the first of January 1970.
- Watch out for implicit conversions to local time, bugs, undocumented features and other strange behavior of Java methods on Date, Time Timestamp, TimeZone and Calendar objects. Confusion guaranteed!

Time Formats (Oracle 8*i*)

- Oracle has the DATE data type to store the *date-time group* up to second precision.
- The range of an Oracle DATE is from 4712 BC to 9999 AD.
- Oracle uses *strings* to put the *date-time group* into the database and to display dates already in the database. The function TO_DATE() converts to an Oracle date and TO_CHAR() converts back to a string.
- Oracle is unaware of the time zone and the offset between local time and UTC.

```
TO_DATE(’18-JUN-2002 16:00:00','MM-MON-YYYY HH24:MI:SS');
```

- Use of unix_seconds *as such* (with simple date arithmetic) means working with UTC (advisable to baptize in your table “my_table” the column name e.g. “utc_time”), :

```
TO_DATE('01/01/1970 00:00:00','MM/DD/YYYY HH24:MI:SS') + unix_seconds/86400;
```

- Date retrieval with custom formatting, and with local summer time conversion

```
SELECT TO_CHAR(utc_time,'YYYY/MM/DD HH24:MI:SS') + 2/24 FROM MY_TABLE;
```

- UTC Date Storage and Retrieval is straightforward, Local Time is a bit tricky.

Time Formats (Oracle 9*i*)

- TIMESTAMP data type:

  extension of DATE with fractional seconds up to **nanoseconds**. Interesting for LHC data.
- TIMESTAMP WITH TIME ZONE (TSTZ) data type:
  variant of TIMESTAMP that includes a *time zone displacement* in its value

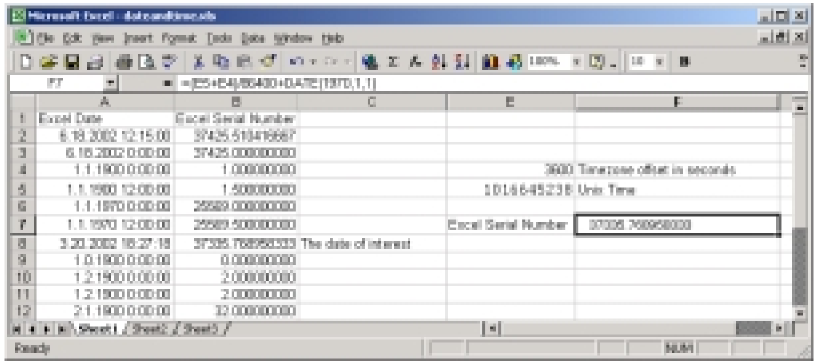
  ```
  TIMESTAMP '18-JUN-2002 16:06:56.665 +02:00'
  ```

  You can replace the UTC offset with the *time zone region* format element.

  ```
  TIMESTAMP '18-JUN-2002 16:06:56.665 US/Pacific PDT'
  ```
- TIMESTAMP WITH LOCAL TIME ZONE (TSLTZ):
  another variant of TIMESTAMP that includes a **time zone displacement** in its value. It differs from TIMESTAMP WITH TIME ZONE in that data stored in the database is normalized to the database time zone.
- The new support for time zones is part of Oracle’s Globalization Support, a feature that is less useful for single time zone organizations like CERN

Excel

- Same problem as Oracle 8i – cannot find local time offset in the language. However a simple macro with 136 constants (2038-1970 years, each with a value for the spring and autumn time change) could be made to do the conversion.



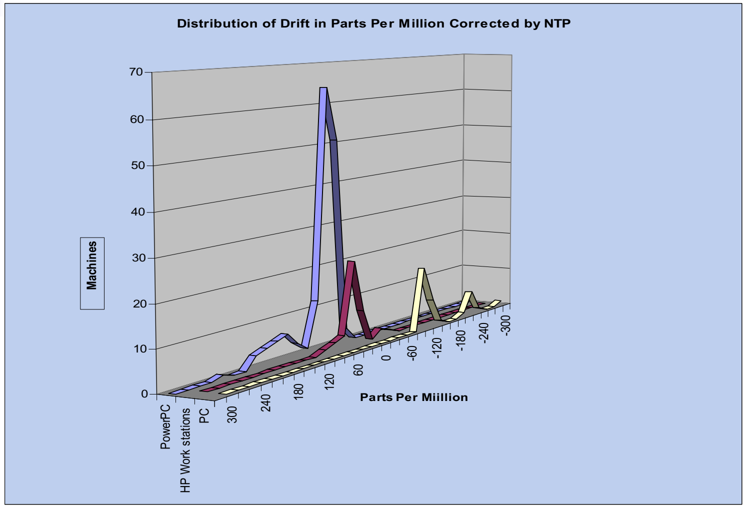
=(E5+E4)/86400+DATE(1970,1,1)

	A	B	C	E	F
1	Excel Date	Excel Serial Number			
2	6.18.2002 12:15:00	37425.510416667			
3	6.18.2002 0:00:00	37425.000000000			
4	1.1.1900 0:00:00	1.000000000		3600	Timezone offset in seconds
5	1.1.1900 12:00:00	1.500000000		1016645238	Unix Time
6	1.1.1970 0:00:00	25569.000000000			
7	1.1.1970 12:00:00	25569.500000000		Excel Serial Number	37336.768958333
8	3.20.2002 18:27:18	37336.768958333	The date of interest		
9	1.0.1900 0:00:00	0.000000000			
10	1.2.1900 0:00:00	2.000000000			
11	1.2.1900 0:00:00	2.000000000			
12	2.1.1900 0:00:00	32.000000000			

How Computers keep time

- When the Operating system is not running the time is kept, typically at 1 second resolution, by a Real Time Clock (RTC) chip driven from a battery. This is used when:
  - Power is off
  - The machine is booting
  - The machine has stopped working!
- When the operating system starts it reads the RTC and uses the time as the starting value for time.
- From then on this time value is incremented, typically every 10 milliseconds, by a periodic interrupt from a timer/counter chip.
- For more time resolution there are processor cycle counters in Pentiums (1 nanosecond resolution for 1 GHz Pentium) and the Decrementer on a PowerPC.
- The quartz crystal oscillators used in computers are chosen for their cheapness and do not keep time as well as your watch – see next slide!

Measured and Corrected Clock Drift in SL Control System



Real Time Timestamping

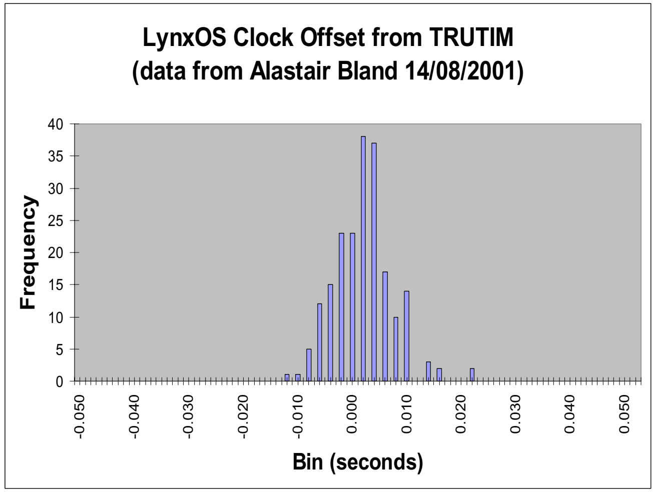
- If in doubt do it in hardware (GPS freeze input <1 microsecond)
- Next best thing is in a real-time OS such as LynxOS or a PLC
  - Bounded time between having an event to timestamp and clock read routine returning (a few 10s of microseconds for highest priority task or in a driver on LynxOS)
- On Unix or Windows2000 systems you may be pre-empted for 10s of milliseconds between getting the event and the clock read routine returning by disk activity, swapping, task switches etc.

Network Time Protocol (1)

- The Network Time Protocol has been used since the early 1990s to keep many CERN computers (including the SL Control System computers) synchronized. The Control System synchronizes from the IT division servers IP-TIME-1 and IP-TIME-2 plus TRUTIM, our own GPS/IRIG-B based server.
- More recently it has been available on Windows2000, provided by the W32time service which implements the Simple Network Time Protocol. Since early this year the NICE2000 machines synchronize their time via the Domain Controllers IP-TIME-1 and IP-TIME-2. A great improvement!
- Internally it is based on a 32 bit unsigned long number of seconds starting 1st of January 1900 and a 32 bit fraction. It will overflow in 2036. It has 233 picosecond resolution.

Network Time Protocol (2)

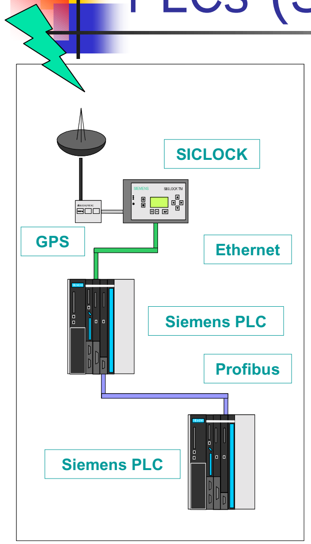
- The chart shows the precision available with the default LynxOS NTP configuration.
- Better results would have been obtained if TRUTIM was in the list of 5 timeservers used!
- All 200 machines running NTP are counted, the data has not been cleaned up



PLCs

- Software in the PLCs can read time, normally with fractions of a second. Usually local time has been used.
- Schneider PLCs can have the time set by MODBUS TCP calls from the outside. They can also synchronize to a Pulse Per Second or Minute hardware input.
- Siemens PLCs also have facilities for hardware inputs to be used for synchronizing the time with great precision.
- Work is under way in LHC/IAS to use NTP or SNTP to get the time in Schneider and Siemens PLC from our usual network sources. However the maximum precision of NTP is limited to about half the network round trip time:
  - 0.5 milliseconds on a fast local network
  - 5 milliseconds when going a long distance through many routers.

PLCs (Siemens)



- Jacky Brahy of LHC/IAS has a GPS receiver supplying time with 1 millisecond precision via ethernet to Siemens PLCs.
- The GPS receiver uses UTC
- The Siclock module receives this UTC over RS485. For the moment it then generates a Sinec H1 packet (one day this will be an SNTP or NTP packet) and sends it with great care to avoid collisions over the ethernet.
  - The packet contains a field representing the timezone offset in ½ hour units.
  - Does this offset change automatically in Spring and Autumn?
- The Siemens PLC sets the local clock with this time, presumably applying the timezone offset.
- The time can be further distributed via Profibus or Ethernet.

Recommendations

- Use one central GPS time source, check it with local GPS receivers wherever possible
- Use UTC time wherever possible
- Possible String formats:
  - Floating point string in XML (nanoseconds needs special parsing)
  - ISO Date string
- Binary formats in network byte order:
  - 64 bit IEEE Double Floating point number (cannot resolve nanoseconds)
  - NTP fixed point 64 bit number (233 picosecond resolution in bit 0)
  - Unix Timeval (1 microsecond resolution)
  - Posix Timespec (1 nanosecond resolution)
- If it is desired to transport timestamp and precision then NTP structures can be used
- Timestamp as close to source of the event/data as possible:
  - Hardware pulses or Machine Timing for precision finer than a millisecond
  - NTP for general computer synchronization