Current state of my work to develop a **Digital Positron Lifetime Measurement** for EPOS

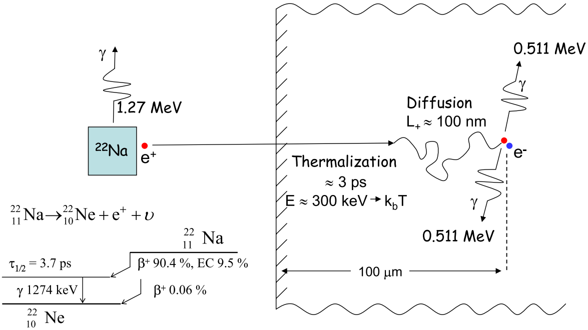
Arnold Krille

Fachbereich Physik, Martin-Luther-Universität Halle-Wittenberg

July 12, 2006

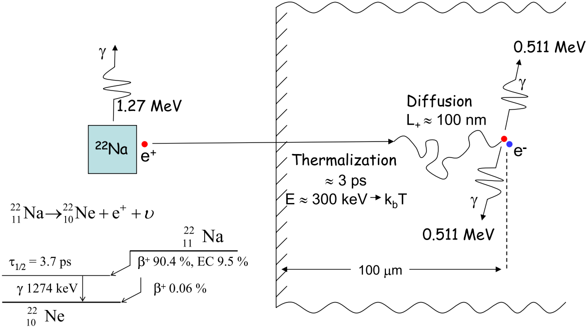
Introduction & Overview

What happens at PA? How is the LT measured?



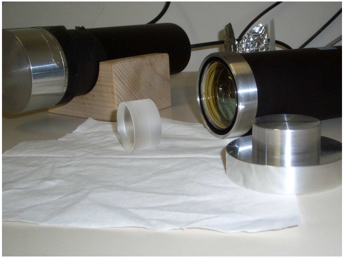
- Positron annihilates with electron, 2 γ-quanta of 511keV are emitted
  $\rightarrow$ End of positron lifetime

What happens at PA? How is the LT measured?



- Positrons emerge in β+-radiation, for example in Na^{22} At the same time a gamma-quantum of 1MeV is emitted $\rightarrow$ Start of positron lifetime

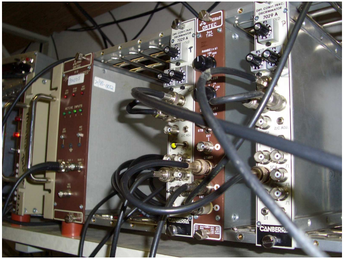
The analog way: Basics I



Scintillator Converts γ-ray into "'visible"' light of 220nm

Photomultiplier Photons trigger (single) electrons, which are gathered, focused and multiplied. Result: a current and a voltage-peak

The analog way: Basics II



Constant Fraction Disc. Sets a window for the impuls-height and syncs the output-pulse to a certain point of the signals edge.

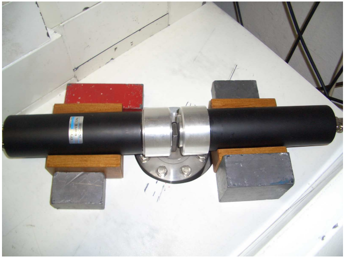
Coincidence Logical AND to only measure on good impuls-pairs.

Time to Amplitude Conv. Output impuls has height proportional to time between start- and stop-signal.

The analog way: 2γ-Setup

Simpliest method:

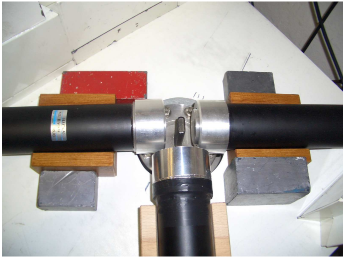
- one photomultiplier tube for the start-quantum
- one photomultiplier tube for the stop-quantum
- tubes headed directly at each other
- Apart from the window with source and sample a Pb-shielding should be used to prevent reflections.



The analog way: 3γ-Setup

Most advanced setup:

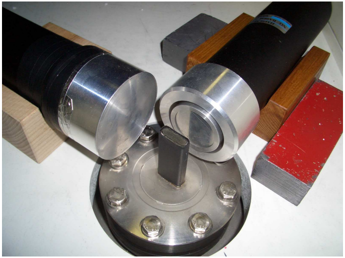
- two tubes for the two stop-quanta facing head-to-head
- one tube for start-quantum vertical to stop-tubes
- “good” pulse only when both stops in time after start-quantum
- \+ less noise, better results



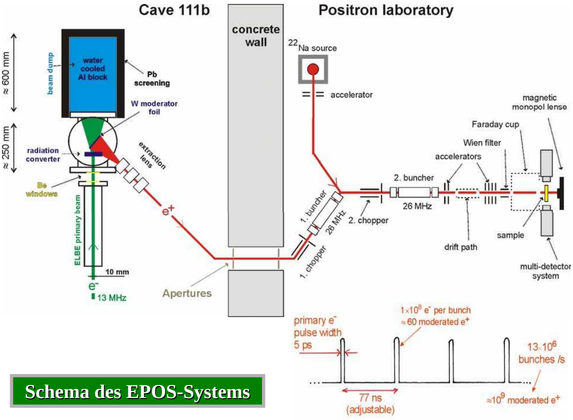
The analog way: 90°-Setup

Mix of 2γ- and 3γ-Setup:

- one tube for start, one tube for stop
- tubes vertical to each other (90°)
- \+ no stop-quanta in start-tube
- \- not as efficient as 2γ-Setup



Detectorsystem for EPOS



Detectorsystem for EPOS

- 8 photomultipliers
  + 2 Ge detectors
- opposing detectors connected to one digitizer
- start-signal from beam
- stop-signal from PM's, either in coincidence (less background noise) or indepedent (faster measurement)

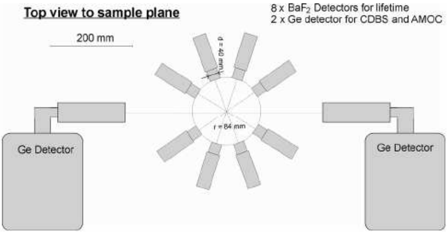
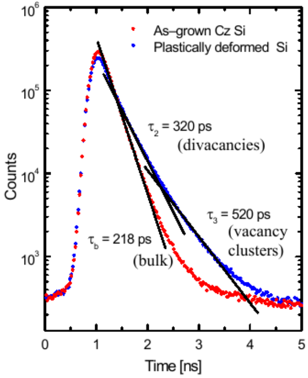
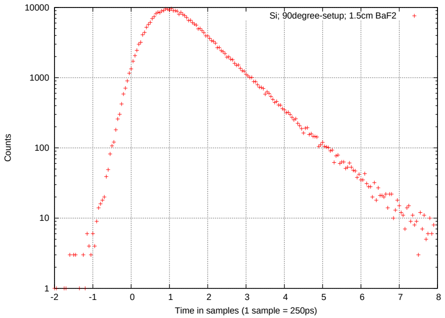


Figure: Planned detector setup

The Resulting Spectra

Resulting spectra look like these:





Basic Digital Way

	Digitizer
+	Minimum-Interpolation
+	Windows (Disc.)
+	Constant Fraction
$\sum$	Spectrum

Hardware | Software

Basic Digital Way

	Digitizer
+	Minimum-Interpolation
+	Windows (Disc.)
+	Constant Fraction
$\sum$	Spectrum

Hardware

Software

Basic Digital Way: Starting the demo

Lets see if we can get some results from ^{60}Co within the next $\sim$ 20 minutes (or until this talk ends).

EODE (EPOS Offline Data Evaluation) will be used to evaluate sampled pulses.

Digital Systems in Detail

First important fact

No need to have dedicated start- and stop-tubes.
Simple coincidence for pulses to reduce background-noise is enough.

Second important fact

Moore’s Law also applies to ideas.

Digital Systems in Detail

First important fact

No need to have dedicated start- and stop-tubes.
Simple coincidence for pulses to reduce background-noise is enough.

Second important fact

Moore's Law also applies to ideas.

Digital Systems in Detail

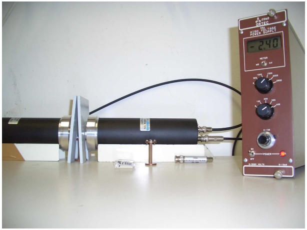
First important fact

No need to have dedicated start- and stop-tubes.
Simple coincidence for pulses to reduce background-noise is enough.

Second important fact

Moore's Law also applies to ideas.

Hardware & System Setup: The Analog Part



- Current testing includes two tubes either in 90°- or 180°-Setup.
- Tubes are directly connected to digitizers, no external amplification is applied.
- For better S/N-Ratio an external coincidence (logical AND) is used to trigger data acquisition only on good pulse-pairs.

Hardware & System Setup: Server and Clients I

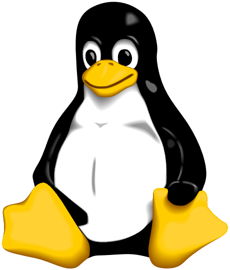
One server: Several tasks

Many clients: One task

Hardware & System Setup: Server and Clients II

One server to rule them all:

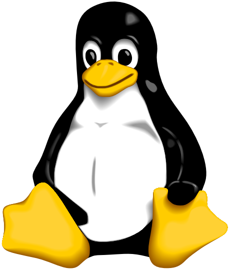
- Controls the environment
- Controls the experiment
- Stores the final data
- Provides the OS for the diskless clients



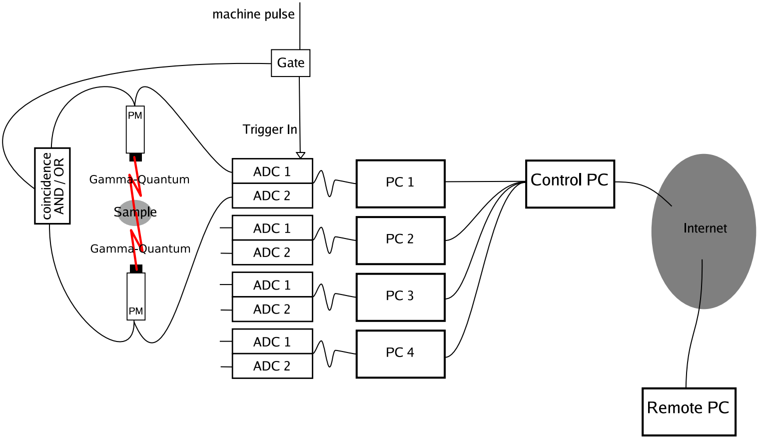
Hardware & System Setup: Server and Clients III

"Dumb" clients:

- Boot via network,
  no disk, no graphics, no keyboards
- Each connected to an Acqiris digitizer
  (2 channels, 4GS/s, 8bit) to gather data
- Either the acquired pulses are stored on the server and evaluated offline
  (for testing)
- or the pulses are evaluated online and only results are saved to disk / sent to the server (into a database).



Hardware & System Setup: All together



Software

What is needed?

- Data storage
- Data evaluation
- Environment control
- Graphical interface that is easy-to-use
- Network-transparency for everything

Software: Using existing parts

Using existing apps and libraries simplifies things:

Data storage There are already good databases available: MySQL, PostgreSQL

Web-Interface Apache+PHP will probably be enough to implement a webinterface for EPOS.

User-Interface Trolltech's Qt provides easy ways to create nice-looking GUIs.
A plus: There are classes for network, database access, threads, etc. in the toolkit too.
See www.trolltech.com for details.

Software: Using existing parts

Using existing apps and libraries simplifies things:

Data storage There are already good databases available: MySQL, PostgreSQL

Web-Interface Apache+PHP will probably be enough to implement a webinterface for EPOS.

User-Interface Trolltech's Qt provides easy ways to create nice-looking GUIs.
A plus: There are classes for network, database access, threads, etc. in the toolkit too.
See www.trolltech.com for details.

Software: Using existing parts

Using existing apps and libraries simplifies things:

Data storage There are already good databases available: MySQL, PostgreSQL

Web-Interface Apache+PHP will probably be enough to implement a webinterface for EPOS.

User-Interface Trolltech's Qt provides easy ways to create nice-looking GUIs.
A plus: There are classes for network, database access, threads, etc. in the toolkit too.
See www.trolltech.com for details.

Software: Writing own parts

Several things have to be written:

- Application framework
- Plugins / Interfaces for evaluation, storage and visualisation
- Network: communications, management, transport

Software: Plugins

Current state: fully functional

Plugins are usable at several parts in varying states:

Evaluation There are several modules to manipulate the data, filter the data and extract variuous information from the data.
The path of evaluation can be changed at runtime, new modules can be used without restarting or recompiling the app.

Software: Plugins

Current state: functional

Plugins are usable at several parts in varying states:

Storage Qt's Model-View-Pattern is used and extended to store the extracted data in models.
The standard table can be used freely in the apps. New table-types have to be added at compile-time.

Software: Plugins

Current state: functional

Plugins are usable at several parts in varying states:

Visualisation The Model-View-Pattern also adds ways to have several different views on models. A View knowing about the abstract Model can display data for several actual implementations of the Model.

Software: Network

Current state: not existing

Several ideas swirl around in my head:

1. One app per client acqiring the data from the digitizer, evaluating the data and sending the results to the server. Uses several threads to parallelize tasks.
   Advantage: easy to implement and to administrate
   Downside: not very extendible, might waste resources

Software: Network

Current state: not existing

Several ideas swirl around in my head:

1. One app per client acqiring the data from the digitizer, evaluating the data and sending the results to the server. Uses several threads to parallelize tasks.
   Advantage: easy to implement and to administrate
   Downside: not very extendible, might waste resources

Software: Network

Current state: not existing

Several ideas swirl around in my head:

1. One app per client acqiring the data from the digitizer, evaluating the data and sending the results to the server. Uses several threads to parallelize tasks.
   Advantage: easy to implement and to administrate
   Downside: not very extendible, might waste resources

Software: Network

Current state: not existing

Several ideas swirl around in my head:

1. One app per client acqiring the data from the digitizer, evaluating the data and sending the results to the server. Uses several threads to parallelize tasks.
   Advantage: easy to implement and to administrate
   Downside: not very extendible, might waste resources

Software: Network

Current state: not existing

Several ideas swirl around in my head:

1. One app per client acquiring the data from...
2. Several apps per client, all connected via network:
   - One app to acqire the pulses and store them in local memory. Server gets notified of the available data.
   - One app (with several threads) to take data from the acquisition-apps, evaluate it and send the results to the server.

Advantage: Distributing work equally on all clients. Even clients without digitizers are possible.

Software: Network

Current state: not existing

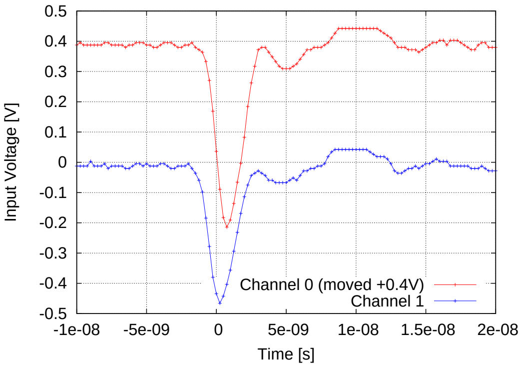
Several ideas swirl around in my head:

1. One app per client acquiring the data from...
2. Several apps per client, all connected via network:
   - One app to acqire the pulses and store them in local memory. Server gets notified of the available data.
   - One app (with several threads) to take data from the acquisition-apps, evaluate it and send the results to the server.

   Advantage: Distributing work equally on all clients. Even clients without digitizers are possible.

Evaluationmethods

Extracting the time-information is the crucial part of the system.



Lets look at different ways to extract the timing-information:

Constant Fraction: **Polynom-Fit**

1. Finding the minimum and interpolating it.
2. Interpolate constant fraction of minimum - baseline

Resolution

Several groups have resolution of 200ps to 250ps.
Own measurements (see Demo) are around 170ps with ^{60}Co.

Constant Fraction: **Gauss-Fit**

1. Apply fit of Gauss-Function: $y = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{s\sigma^2}}$

Resolution

[Aavikko, 2004 NIM A]: 200ps - 220ps

Constant Fraction: **Spline**

1. Smoothing the wave-form
2. Cubic spline interpolation

Resolution

[Saito, 2001] with a 4GS/s digitizer on ^{60}Co: 118ps
[Aavikko, 2004 ACTA] with a 2GS/s digitizer on SiC: 146.7ps

Constant Fraction: **integrals (iCF)**

1. Search for minimum
2. Integrate around minimum
3. Constant fraction with simple polynom-interpolation on integrated pulse

Resolution

According to Prof. Becvar a resolution of $\sim$100ps is reached with ^{60}Co and 4GS/s digitizers.

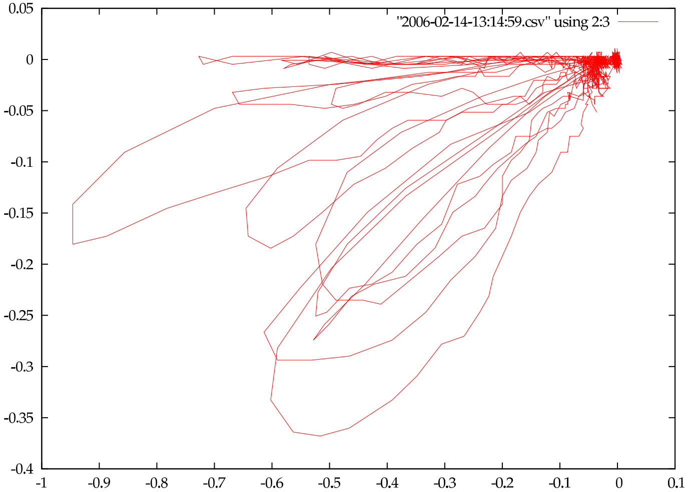
Idea: Deconvolution

Idea from LHC-experiments:

- Measured signal is convolution of δ-peak with resolution-functions of tubes, amplifiers, etc.
- Thus deconvolution of the sampled pulse should return δ-peak.

Might be similar to *Autocorrelation* of both pulses with each other or with general pulse.

Idea: Polar-like plots



Comparing the results

Method	Resolution
Analog measurements	$>$200ps
Polynom-Fit	200-250ps own: 170ps
Gauss-Fit	200ps - 220ps
Smoothing Spline	118ps - 150ps
integral CF	$\sim$100ps

Conclusion: Evaluating the demo

- What is the resolution acquired with the demo-data?
- How long did it take?

Conclusion: Future

What is left to the future?

In general:

- Generally digital systems compare rather well to analog equipment.
  Doing math in software gives way to using new algorithms without paying for new hardware.

For EPOS:

- Different methods of evaluating the pulses have to be tested to find the optimum.
- There is still a lot of software to be written.

Conclusion: Future

What is left to the future?

In general:

- Generally digital systems compare rather well to analog equipment.
  Doing math in software gives way to using new algorithms without paying for new hardware.

For EPOS:

- Different methods of evaluating the pulses have to be tested to find the optimum.
- There is still a lot of software to be written.

Conclusion: Literature, Links, Thanks

Thanks for your attention!

Get the slides at http://positron.physik.uni-halle.de/.

F. Becvar, J. Cizek, I.Prochazka, J. Janotova
The asset of ultra-fast digitizers for positron-lifetime spectroscopy
NIM A 539 (2005) 372-385

J.Nissilä, K.Rytsölä, **R. Aavikko**, A. Laakso, K. Saarinen, P. Hautojärvi
Performance analysis of a digital positron lifetime spectrometer
NIM A 538 (2005) 778-789

R. Aavikko, K. Rytsölä, J. Nissilä, K. Saarinen
Stability and Performance Characteristics of a Digital Positron Lifetime Spectrometer
ACTA PHYSICA POLONICA A, Vol. 107 (2005)

H. Saito, Y. Nagashima, T. Kurihara, T. Hyodo
A new positron lifetime spectrometer using a fast digital oscilloscope and BaF_2 scintillators
NIM A 487 (2002) 612-617