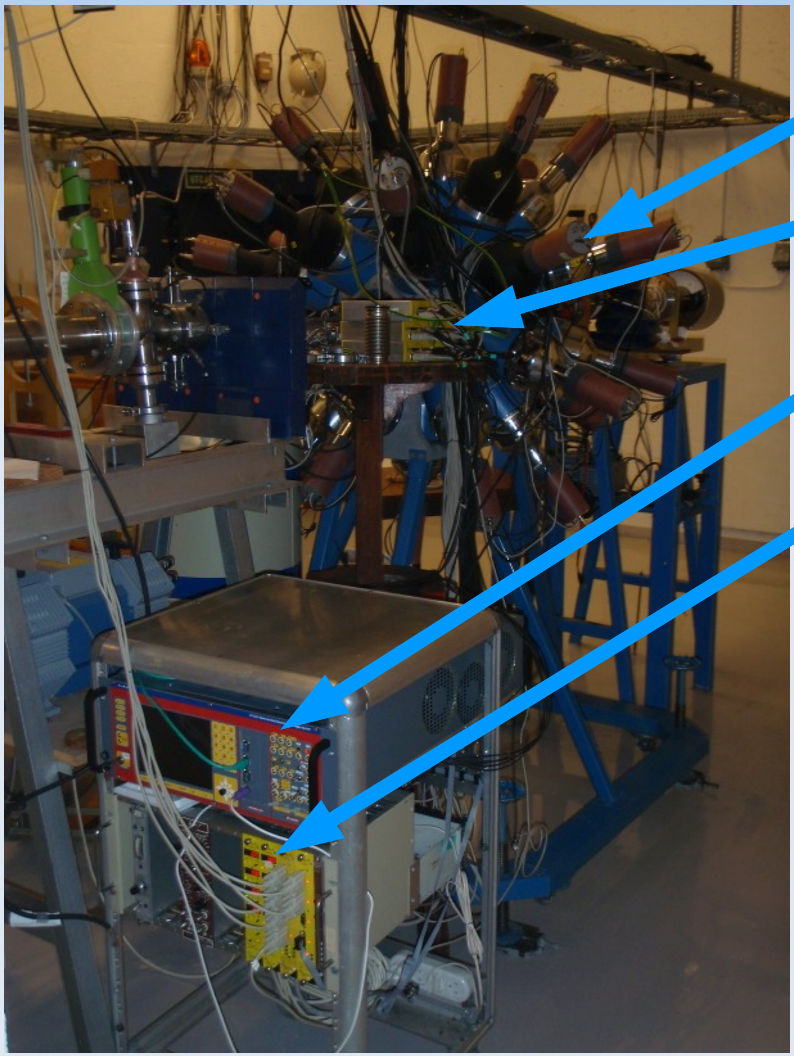
Short ACQ Introduction

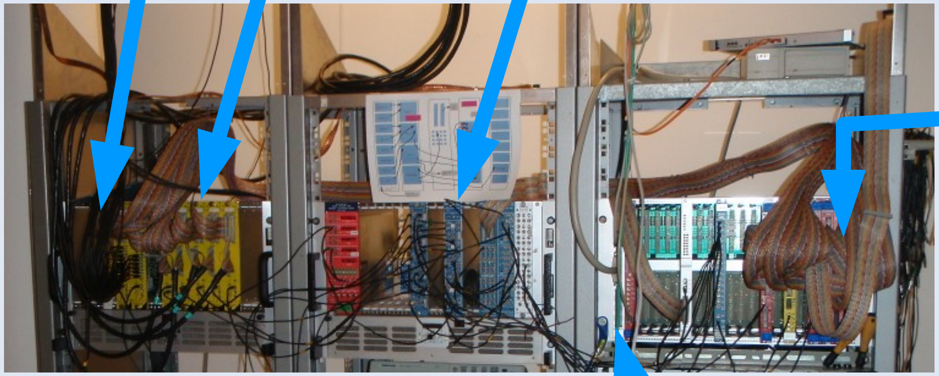
- experimental setup
  - detectors, electronics, data acquisition
- event structure
  - file format, data types
- sorting into ROOT files
  - batch files, spectra, calibration

Experimental Setup



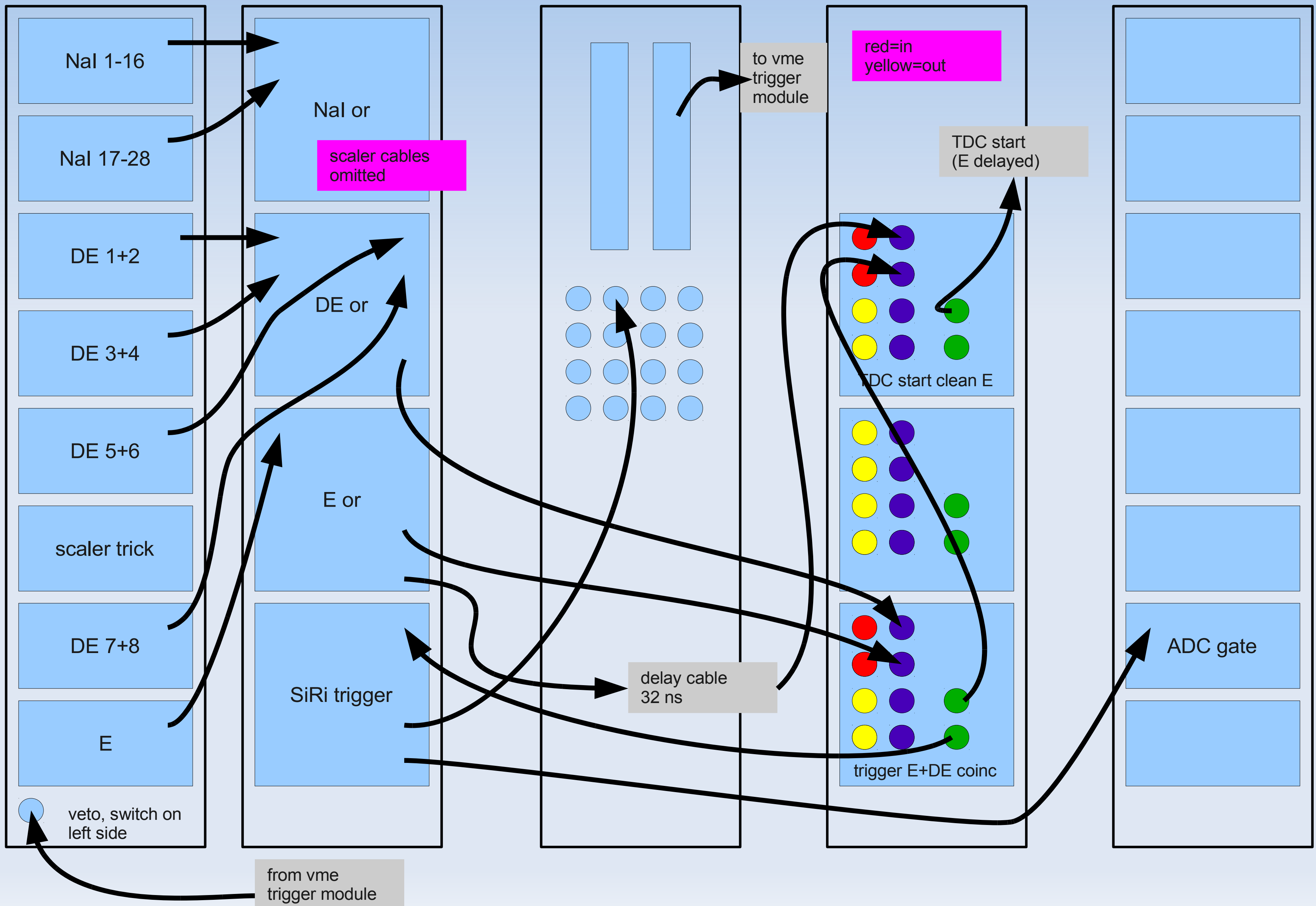
- CACTUS
- SiRi preamplifiers
- CACTUS high voltage
- SiRi high voltage

Electronics



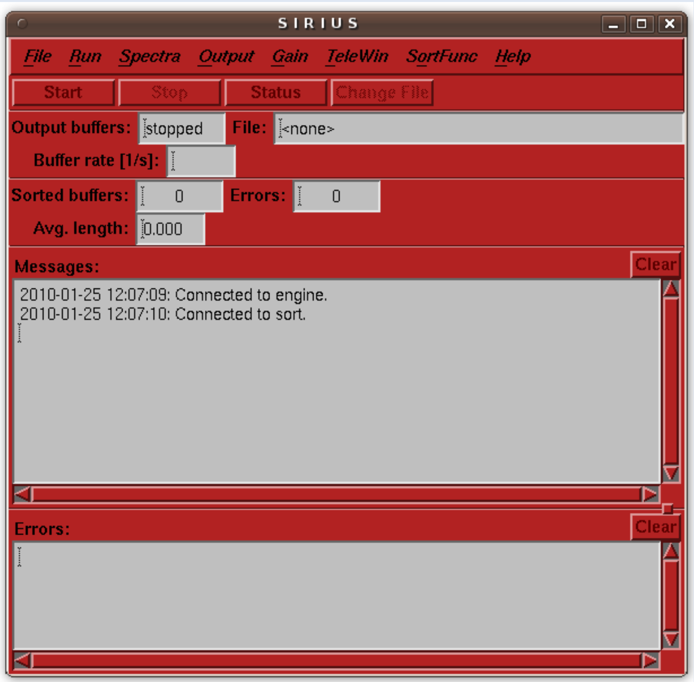
- CACTUS amplifiers
- SiRi amplifiers
- trigger logic
- ADCs and TDCs
- VME computer

Trigger Logic



DATA Acquisition

- VME computer (bobcat):
  - eventbuilder collects data from ADCs, ...
- Linux PC (tiger):
  - `usb-engine` receives data from VME computer and writes to disk
  - `acq_master` controls engine
  - `rupdate` shows spectra
  - `mesytec-rc.py` adjusts amplifier settings



Event Structure

32 24 16 8 0

C0	00	00	n

32 30 23 16 0

0	b_1	c_1	v_1
0	b_2	c_2	v_2
	...	...	
0	b_n	c_n	v_n

32 24 16 8 0

C0	00	00	m

... ...

32 24 16 8 0

80	00	00	00

- each buffer is 128kB large
- event header: C0 00 00 nn
- data:
  - b = box identifier, 7 bit
  - c = channel number, 7 bit
  - v = value, 16 bit
- next event's header
- ...
- end of buffer: 80 00 00 00

Box and Channel Numbers

- TPU, E, ΔE should always be present
- NaI **should** have t and E at the same time
- clock and scaler
  - every ~1000/4000 events
  - in pairs of 2 channels

TPU	0x00	0
clock	0x01	16, 17
scaler	0x02	0..31
NaI time 0..27	0x10	0..31
NaI energy 0..27	0x24	0..31
E	0x21	0..31
ΔE 0..3	0x22	0..31
ΔE 4..7	0x23	0..31

Event Data (1)

- TPU: which trigger – presently unused information
- E back detetor chips:
  - guard rings: 0, 2, ..., 14
  - detectors: 1, 3, ... 15
- ΔE: front detectors
  - 0..7 for back detector 0
  - 8..15 for back detector 1
  - ...

Event Data (2)

- NaI time:
  - time from “back over threshold” until “NaI over threshold”
  - channels 0..27
- NaI energy:
  - channels 0..27

Other Data

- clock:
  - seconds since 1.1.1970 in 2*16 bit
  - every ~1024 events
- scalers:
  - ΔE1+2: scaler 0 = channels 0+1 (high 16 bit + low 16 bit)
  - NaI 0..15: scaler 1 = channels 2+3
  - NaI 16..27: scaler 2 = channels 4+5
  - E: scaler 3 = channels 6+7
  - every ~4096 events

Sorting-Program for ROOT

- copy `userroot.h, userroot.cpp, Makefile, user_routine_root.cpp`
  - `user` directory in `.../sirius/src/sort`
- modify `Makefile` and, if necessary, `user_routine_root.cpp`
- run `make` to compile `offline_sort`
- run `offline_sort XY.batch` to sort
- look at spectra in ROOT

Batch Files for Sorting

- `# ...` = comment
- `gain file ...` = load gainshift from file
- `parameter ...` = set parameter value
- `outfile ...` = set output filename
- `data file ...` = read event file

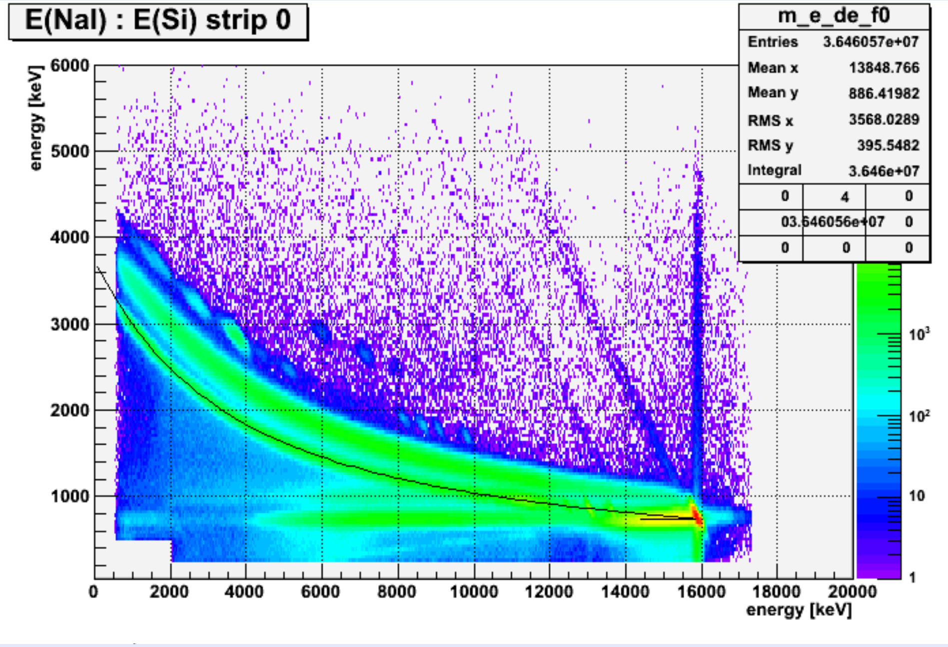
```
# batch file for 92Zr before TDC range change
gain file gainshifts.dat
parameter tnai_corr_enai = 12.985 30447.6 -58.8935 -4.43951e-03
parameter tnai_corr_esi  = 12.641 -1.08751e+05 -516.198 -9.95697e-05
parameter ex_from_ede    = 16551.9 -0.969197 -9.57725e-07 \
   16540.8 -0.968965 -9.66113e-07 \
   16528.9 -0.968702 -9.74982e-07 \
   16515.9 -0.968367 -9.8629e-07 \
   16502 -0.967961 -1.00006e-06 \
   16487.1 -0.96749 -1.01574e-06 \
   16471.1 -0.966949 -1.03386e-06 \
   16453.9 -0.966319 -1.05469e-06
#parameter ex_corr_exp     = 0 1    0 1    0 1    0 1    0 1    0 1    0 1    0 1
parameter ex_corr_exp    = -116.2561 1.0124 \
 -122.8256 1.0118 \
 -120.0651 1.0097 \
 -120.0082 1.0078 \
 -118.2018 1.0051 \
 -114.3768 1.0019 \
 -109.4958 0.9984 \
 -104.1663 0.9945
outfile tmp/sorted_92zr_pp.root
data file datafiles/sirius-20090607-131232.data.gz
...
```

ROOT Spectra

- E:ΔE for all strips:
  - m_e_de_b0f0, ..., m_e_de_b7f7, ...
- NaI time:Nai energy
  - individual: m_nai_e_t_01, ..., m_nai_e_t_28
  - all summed up: m_nai_e_t
  - corrected: m_nai_e_t_c
- Nai time:E back
  - for each back dectector: m_siri_e_t_0, ..., m_siri_e_t_7
  - all together: m_siri_e_t

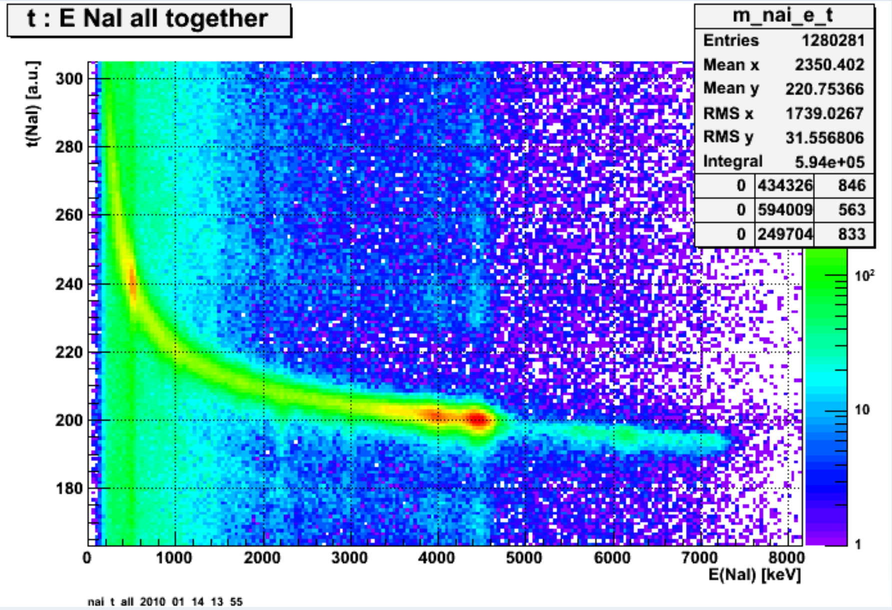
SiRi Calibration

- use ^{12}C data with sufficient statistics
- calculate energies for ground state and first excited state with rkinz
- find all 3*8*8 peak positions in E and DE
  - "only" 384 peaks
  - ROOT macro `peaks2D.C` might help
- extract calibration coefficients by fitting
  - spreadsheet might help



NaI Calibration

- again, use ^{12}C data with good statistics
- find 511keV and 4.4MeV peaks for all 28 NaI detectors in time:energy matrices (m_nai_e_t_00, ...)
  - ROOT macro `peaks2D.C` might help
- extract calibration coefficients by fitting
  - spreadsheet might help

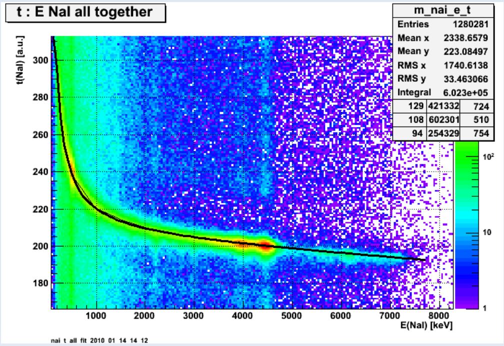


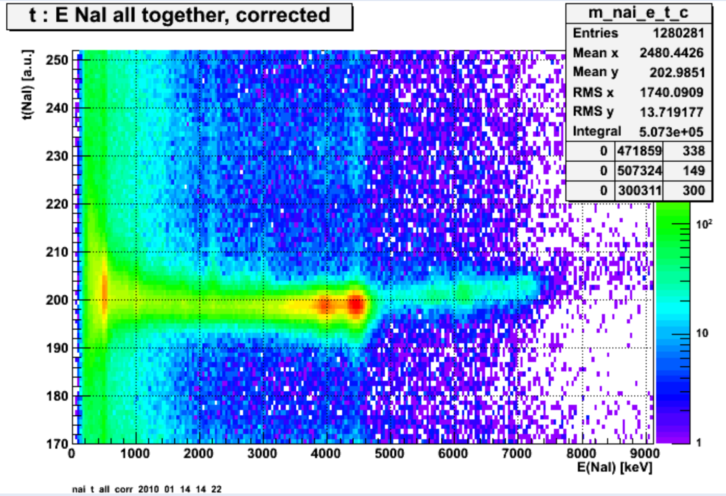
NaI Time Correction

- determine points on curve in m_nai_e_t
  - ROOT: toolbar, new Graph, click, click, click, ...

```
{ for(int i=0; i<Gname->GetN(); ++i) { cout << Gname->GetX()[i] << ' ' << Gname->GetY()[i] << endl; } }
```

- fit curve with function:
  - `TF1* f3=new TF1("f3","200+[0]+[1]/(x+[2])+[3]*x")`
  - this goes into the parameter `tnai_corr_enai`
- same for E_{back} with m_siri_e_t
  - parameter `tnai_corr_esi`





Other Steps

- check that everything is okay
  - e.g., by looking at m...evol where various things are plotted as a function of time
- determine excitation energy
  - usually need re-fit to get correct energies
- make ALFNA matrix
  - use macro to write ROOT matrix in MAMA format
- run `mama`
- ...