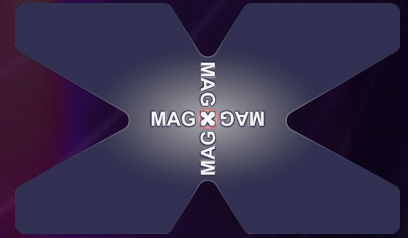


EPICS SlowControl for SRS/APV

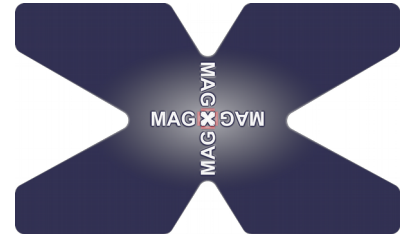


MiniWeek CERN 14.12.2017

Pepe Gülker

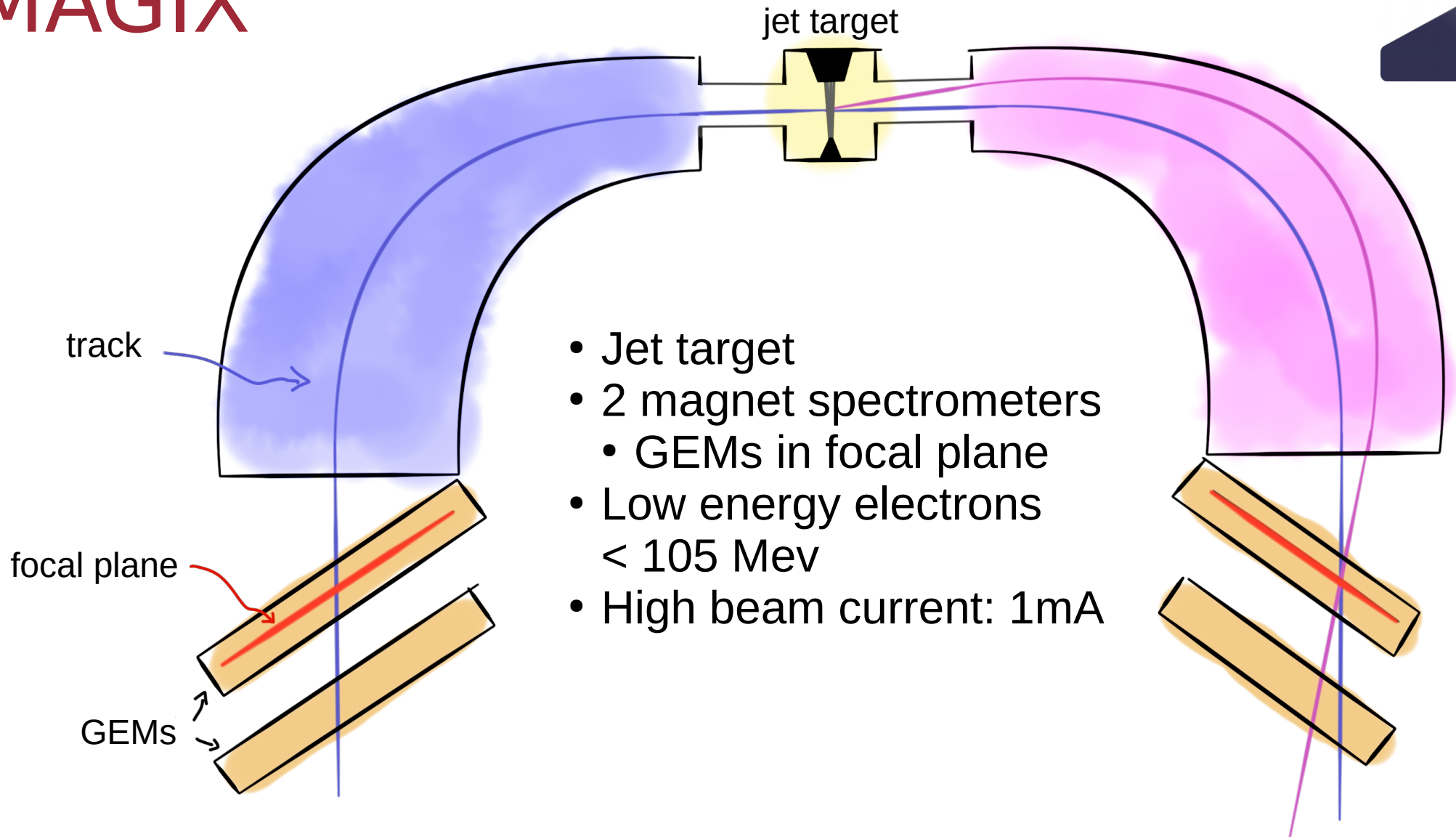
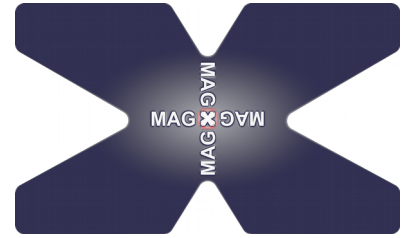
jguelker@uni-mainz.de

Outline

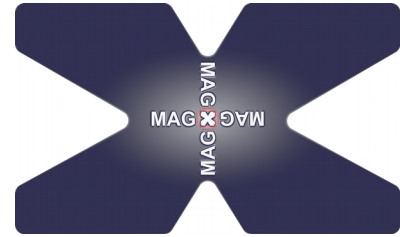


- new experiment -> new slow control
- EPICS in a nutshell
- EPICS 4 SRS
- GUI / CSS

MAGIX

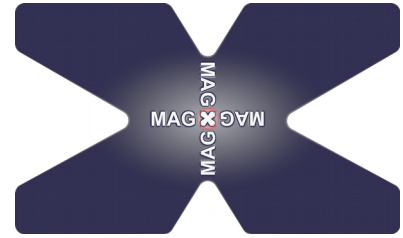
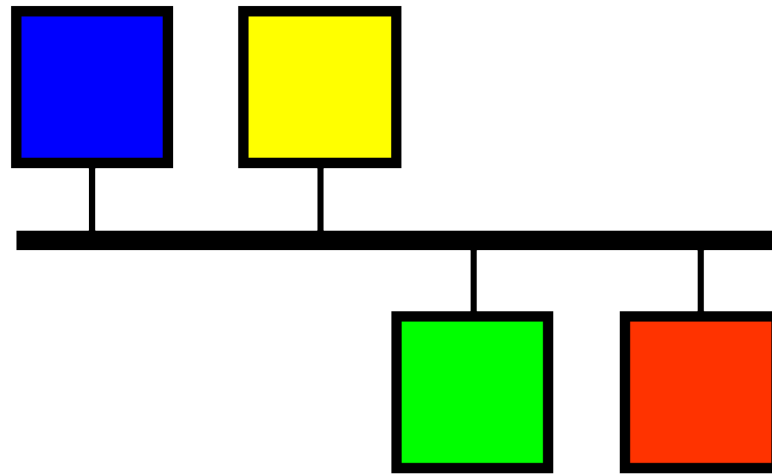


Decide which system to use



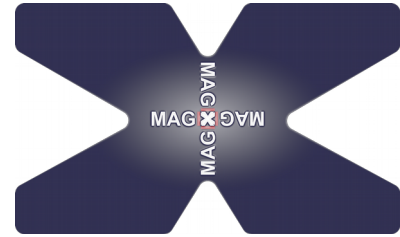
Centralized	<Structure>	Decentralized
Implement it	<Reinvent the wheel?>	Use existing
For lab	<Scope>	For experiment
Flexibility	<Stiffness>	Stability

EPICS



Experimental Physics and
Industrial Control System

10 Things...

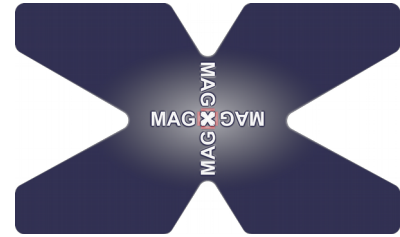


Ten Really Neat Things About EPICS

- 1) It is free. No license fees, no new payment for every upgrade. You can download EPICS free of charge from the web.
- 2) It is Open Source (i.e. the source code is accessible). Adaptions and changes due to a special environment are therefore possible.
- 3) There are lots of users. It is tested and most bugs are already found.
- 4) All a client needs to know to access data is a PV name. No single point of failure due to a nameserver and no messing around with fixed addresses.
- 5) You can pick the best tools out there ...
- 6) ... or build your own.
- 7) The boring stuff is already done. For example the communication with Channel Access is stable and well tested.
- 8) There is a lot of expertise available close by.
- 9) A good contribution becomes internationally known.
- 10) It doesn't matter whether you need 10 PVs or 10 Million PVs. You can scale EPICS almost freely.

<http://www.aps.anl.gov/epics/neat.php>

2 (4) Names...



PVs (Process Variables)

- Holding the values
- Different datatypes
- Unique name
- Several control fields
 - Operational range
 - Alarm levels
 - Accuracy
 - Update rate
 - ...
- Defined in .db files

IOCs (Input Output Controller)

- Server
- Loads the PVs
- Loads the drivers
- Defined in st.cmd files

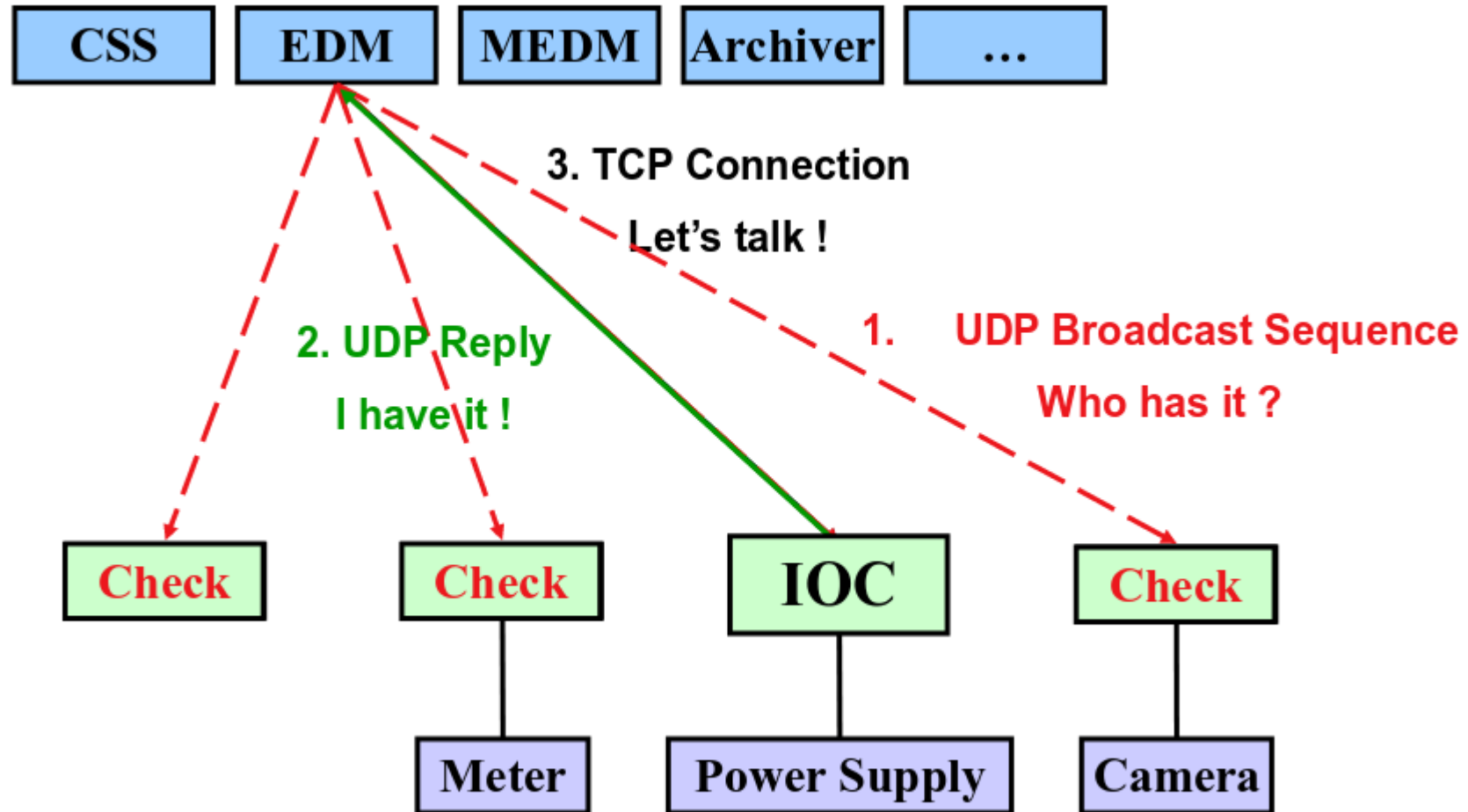
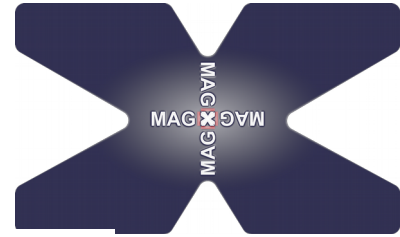
OPI (Operational Interface)

- Hardware running EPICS

CA (Channel Access)

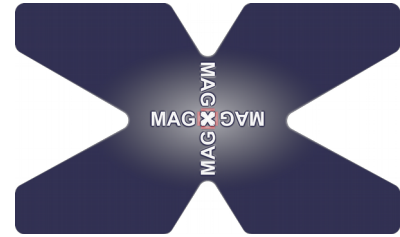
- Network independent access to IOCs

1 Broadcast...



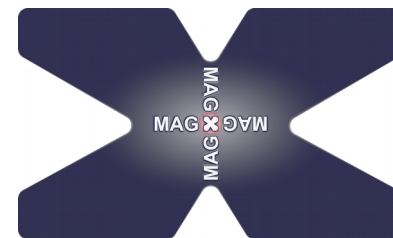
<http://www.aps.anl.gov/>

3 Commands...



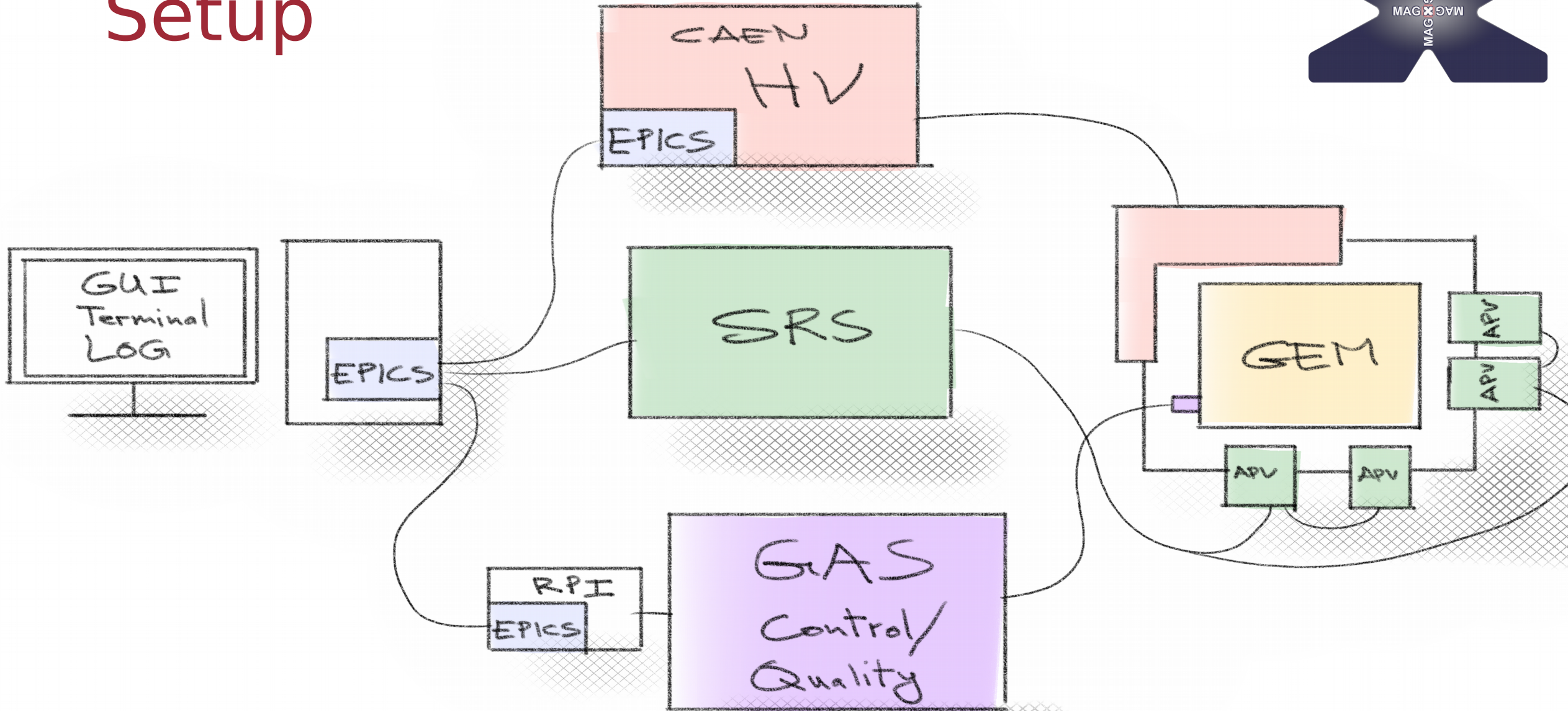
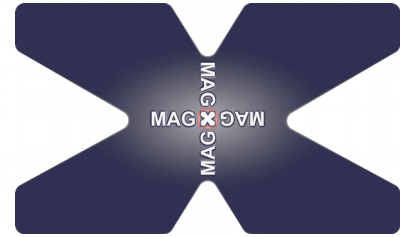
caget
camonitor
caput

```
magixadm@magix-23:~$ caget BEAM:SCAMON:FRQ:Ch0
BEAM:SCAMON:FRQ:Ch0          23745
magixadm@magix-23:~$ camonitor BEAM:SCAMON:FRQ:Ch0
BEAM:SCAMON:FRQ:Ch0          2017-12-02 21:00:16.681497 23820.2
BEAM:SCAMON:FRQ:Ch0          2017-12-02 21:00:16.681497 23733.1
BEAM:SCAMON:FRQ:Ch0          2017-12-02 21:00:17.715404 23733.1
BEAM:SCAMON:FRQ:Ch0          2017-12-02 21:00:17.715773 23875.1
BEAM:SCAMON:FRQ:Ch0          2017-12-02 21:00:18.749571 23875.1
BEAM:SCAMON:FRQ:Ch0          2017-12-02 21:00:18.749878 23838.3
BEAM:SCAMON:FRQ:Ch0          2017-12-02 21:00:19.783808 23838.3
BEAM:SCAMON:FRQ:Ch0          2017-12-02 21:00:19.784110 23712.2
BEAM:SCAMON:FRQ:Ch0          2017-12-02 21:00:20.818996 23712.2
^C
magixadm@magix-23:~$ caput BEAM:SCAMON:FRQ:Ch0 5
Old : BEAM:SCAMON:FRQ:Ch0          23852.3
New : BEAM:SCAMON:FRQ:Ch0          5
magixadm@magix-23:~$
```

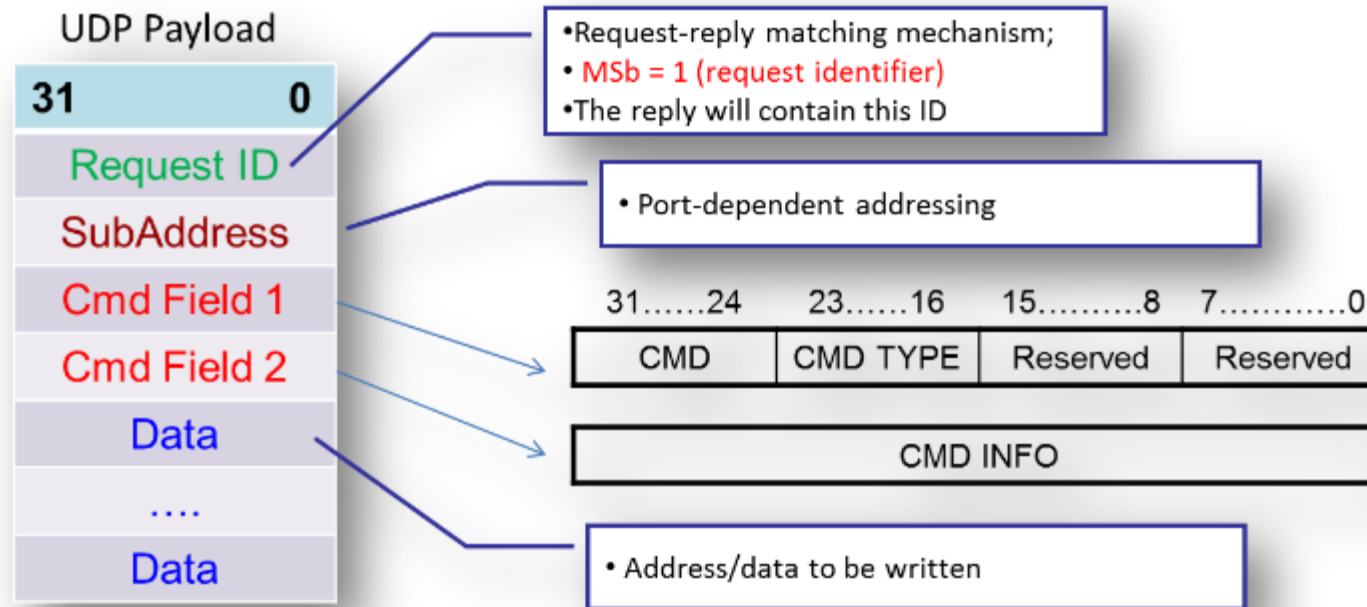
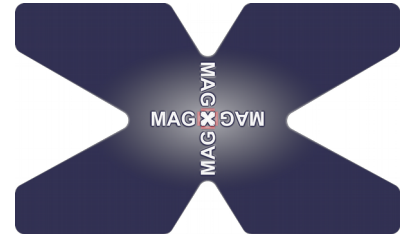


How to get the system running?

Setup



SRS Request



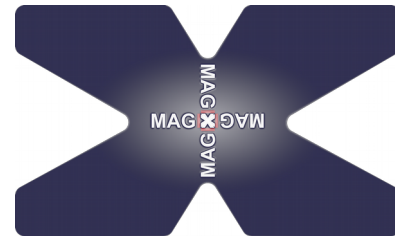
https://espace.cern.ch/rd51-wg5/srs/Documentation/SRS_Slow_Control_Manual.pdf

```

UDPWritePairs{
  LockTimeout = 120;
  out "\x80%(\$1:UDP_IDCounter:tmp)0.3r",
      "\x00\x00\xff%(\$1:APVSelector)0.1r", # Subadress
      "\xAA\xAA\xff\xff", # CMD1
      "\x00\x00\x00\x00", # CMD2
      "\x00\x00\x00", $2, # Data Adress
      "%0.4r"; # Data
  in "\x00%(\$1:UDP_IDCounter:tmp)=0.3r",
      "\x00\x00\xff%(\$1:APVSelector)=0.1r" # Subadress
      "\xAA\xAA\xff\xff" # CMD1
      "\x00\x00\x00\x00" # CMD2
      "\x00\x00\x00\x00" # Error Code
      "%04r"; # Data
  EXTRA_INPUT=ignore;
}

```

← .proto file



→ .db file

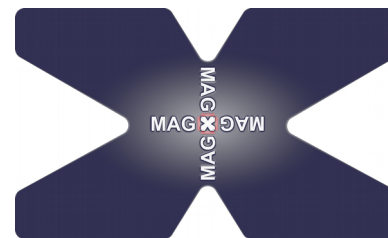
```

record(ao, "$(P):BCLK_FREQ")
{
  field(DESC, "Period of the trigger sequencer")
  field(DTYP, "stream")
  field(OUT, "@devSRS_UDP.proto UDPWritePairs($(P),0x02) $(PORT)")
  field(FLNK, "$(P):UDP_IDCounter:tmp")
}

record(ao, "$(P):BCLK_TRGDELAY")
{
  field(DESC, "Delay between the ext/int and APV trigg")
  field(DTYP, "stream")
  field(OUT, "@devSRS_UDP.proto UDPWritePairs($(P),0x03) $(PORT)")
  field(FLNK, "$(P):UDP_IDCounter:tmp")
}

record(ao, "$(P):BCLK_TPDELAY")
{
  field(DESC, "Delay between the trig and test-pulse")
}

```



```
## Load record instances
#####
# Set up ASYN ports
# drvAsynSerialPortConfigure port ipInfo priority noAutoconnect noProcessEos

drvAsynIPPortConfigure("SRS_App","10.0.0.2:6039:6007 UDP",0,1,0)
asynSetTraceIOMask("SRS_App",-1,0x2)
asynSetTraceMask("SRS_App",-1,0x9) #0x1 print only error 0x9 print all information

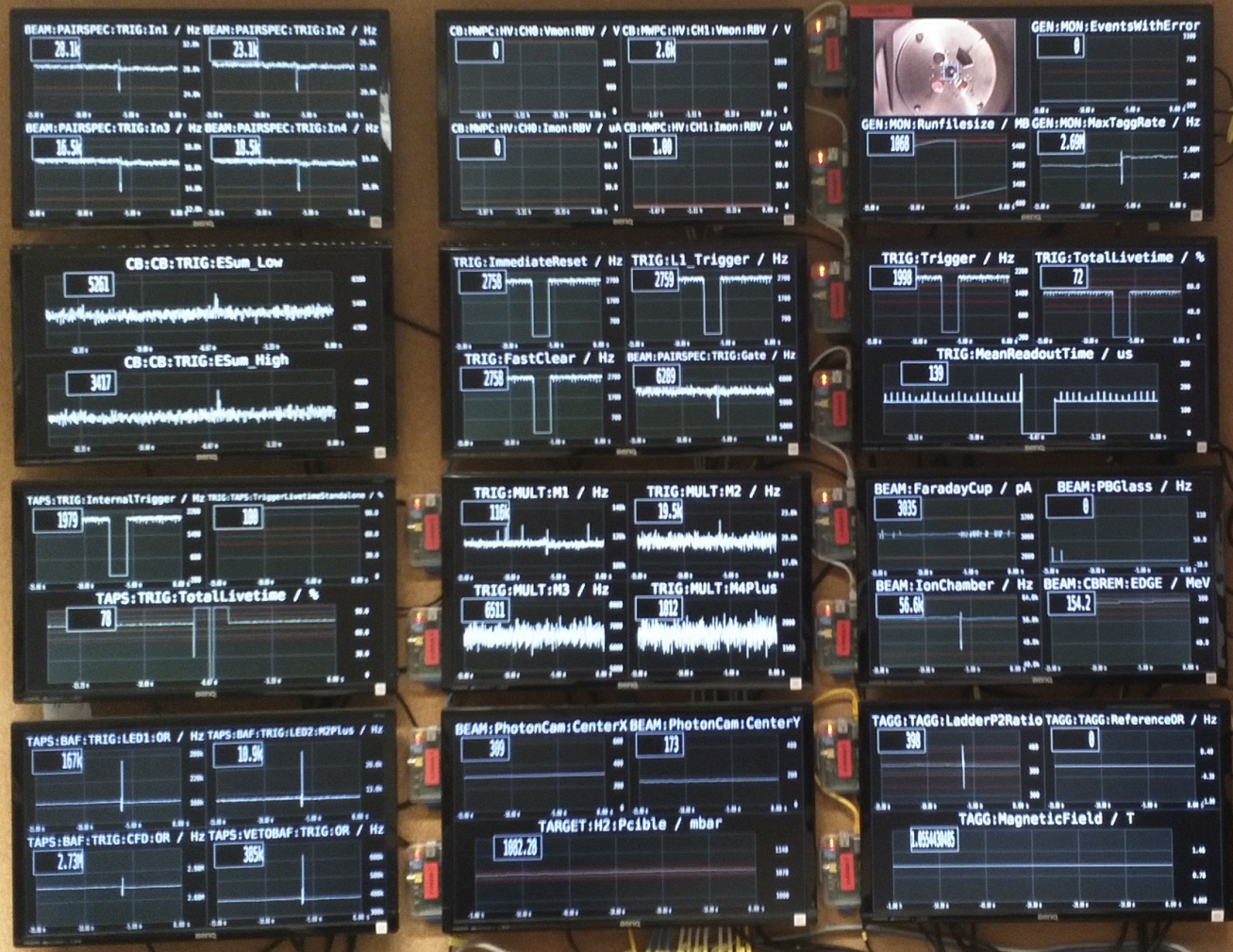
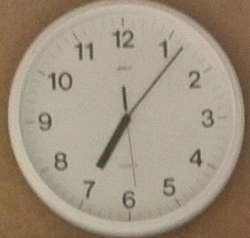
drvAsynIPPortConfigure("SRS_Hybrid","10.0.0.2:6263:6007 UDP",0,1,0)
asynSetTraceIOMask("SRS_Hybrid",-1,0x2)
asynSetTraceMask("SRS_Hybrid",-1,0x9) #0x1 print only error 0x9 print all information

drvAsynIPPortConfigure("SRS_ADCCARD","10.0.0.2:6519:6007 UDP",0,1,0)
asynSetTraceIOMask("SRS_ADCCARD",-1,0x2)
asynSetTraceMask("SRS_ADCCARD",-1,0x9) #0x1 print only error 0x9 print all information

drvAsynIPPortConfigure("SRS_System","10.0.0.2:6007:6007 UDP",0,1,0)
asynSetTraceIOMask("SRS_System",-1,0x2)
asynSetTraceMask("SRS_System",-1,0x9) #0x1 print only error 0x9 print all information
#####
## Load record instances
cd "$(TOP)"

dbLoadRecords("db/devSRS_AppReg.db","P=Specs:GEM:SRS, PORT=SRS_App")
dbLoadRecords("db/devSRS_HybridReg.db","P=Specs:GEM:SRS, PORT=SRS_Hybrid")
dbLoadRecords("db/devSRS_ADCCARDReg.db","P=Specs:GEM:SRS, PORT=SRS_ADCCARD")
dbLoadRecords("db/devSRS_SystemReg.db","P=Specs:GEM:SRS, PORT=SRS_System")
dbLoadRecords("db/devSRS_TmpReg.db","P=Specs:GEM:SRS")
dbLoadRecords("db/devCAEN_Calc.db","P=Specs:GEM:HV, N=hvpower")
```

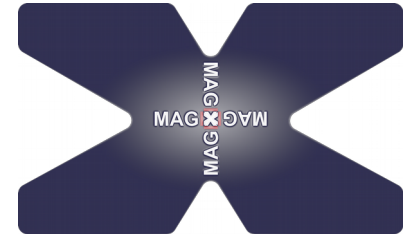
← st.cmd file



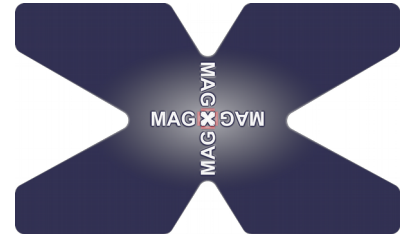
COORDINATORS
 ARB 0177 5130586
 @ 27366
 H. 0170 3075072

EVERY run
 4 run parameters to ELOG EVERY run
 SLECT wall for long term stability
 current scales (RBEETS) to "normal" values
 anything important to ELOG
 day (day shift) shift duties
 EFFICIENCY
 BACKGROUND (~10k/s) Put Bglases in during it
 (~30k/s)
 PARA (~30k/s)
 PERP (~30k/s)
 ROUNDS (~10k/s) REMOVE display from beamline
 OFF!
 AGCEFF, OPTIMIZE LADDER / P2

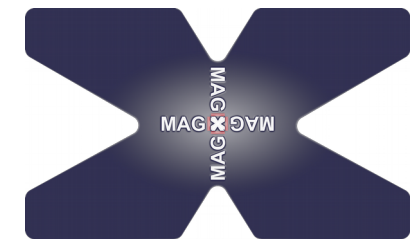
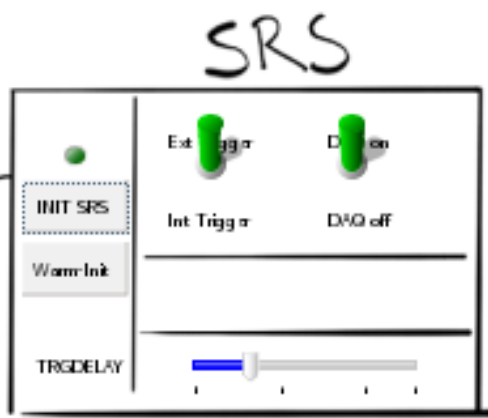
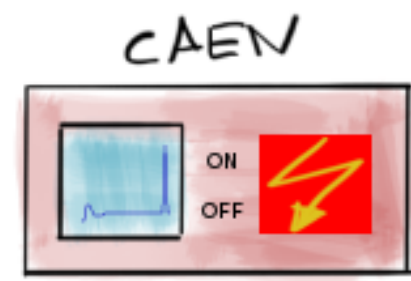
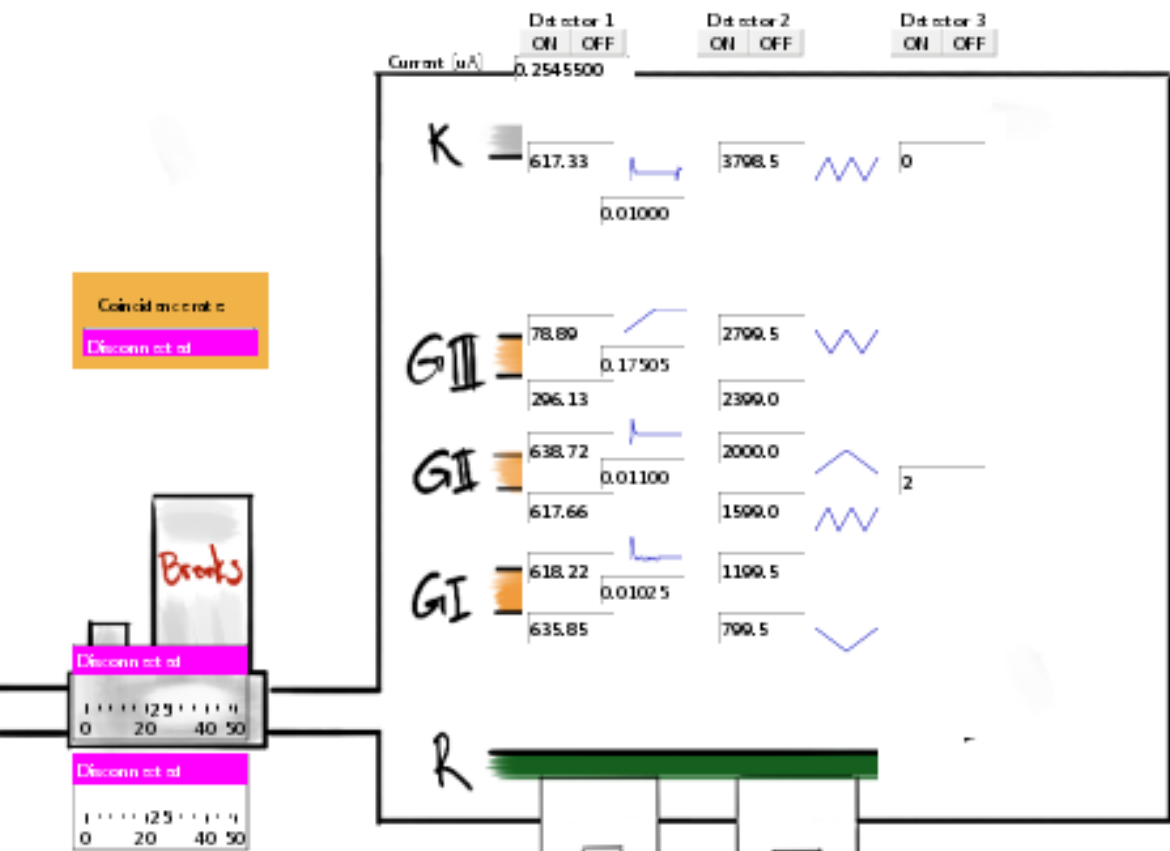
Achtung!
 Notfall im
 Bedienungsraum MAMI
 Bitte helfen sie!



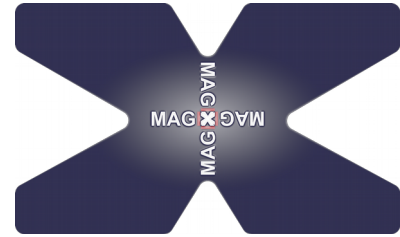
Control System Studio



- drag and drop GUIs
- based on Eclipse
- BOY specialized for usage with EPICS
- Lots of knobs, bars, graphs
- Webkit available
- controlsystemstudio.org



Summary

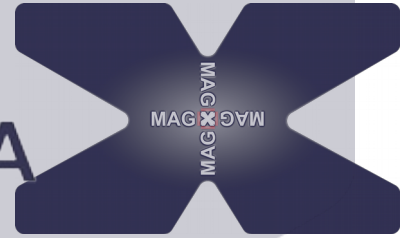


Build slow-control-system on EPICS

- flexible
- stable
- scalable
- all components included

Implemented UDP communication for SRS/APV

- full functionality
- more stable than SRSDCS and Labview-program
 - most likely due to initialization routine
- GUI has to be improved
- try it for VMM3



**THANK YOU FOR YOUR
ATTENTION!**

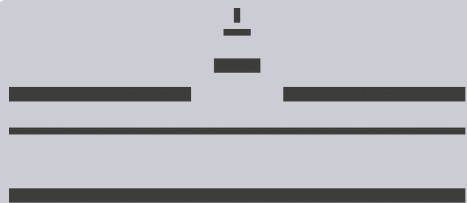
<http://magix.kph.uni-mainz.de>



**Massachusetts
Institute of
Technology**



University of Ljubljana

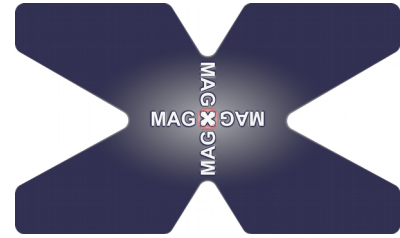


**WESTFÄLISCHE
WILHELMS-UNIVERSITÄT
MÜNSTER**

**JOHANNES GUTENBERG
UNIVERSITÄT MAINZ**



MXSlowControl



Features of MXSlowControl:

- autocompletion (bash and zsh)
- real drivers (no clients)
- GUI separated from control algorithms
- pyEPICS