Progres in SVD slow control

Szymon Bacher

Institute of Nuclear Physics, PAN Kraków
University of Science and Technology, Kraków

Currently targeted topics

- Startup procedure
- SVD-QM (quality monitor)
- NTC software

Startup sequence (1)

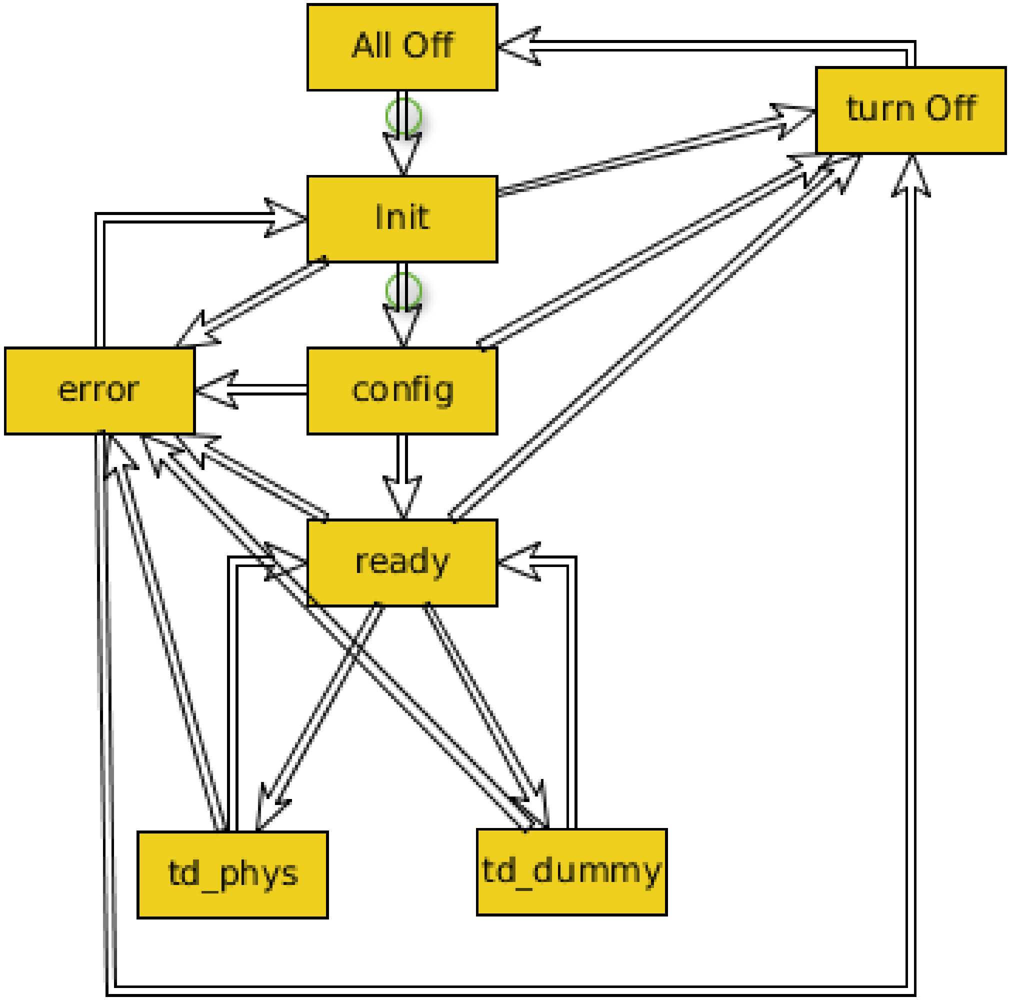
Action	Abbreviation	Prerequisite	master state transition	Extra Conditions	flag raised for MASTER RC
Turn on all the PCs running EPICS	SC_ON				
Turn on ENV monitors	ENV_ON	SC_ON			
Turn on Cooling	IB_ON	SC_ON?			
Turn on DQM	DQM_ON	SC_ON			
Initialize FADC	FADC_INIT	SC_ON			
Turn on Low Voltage	FADC_INIT	ENV_ON&IB_ON			
Configure APV	APV_INIT	FADC_INIT			READY
Ramp UP HV	HV_ON	APV_INIT	PS(any->PEAK)	Beam is down or stable	
Turn on separation voltage	SEP_ON	HV_ON			
Take real data	Running	SEP_ON&DQM_ON	RC(READY->Running)		RUNNING
Take dummy data	RunNOP	APV_INIT&DQM_ON	RC(READY->RUNNINGwithNOPS)		RUNNINGwithNOPS

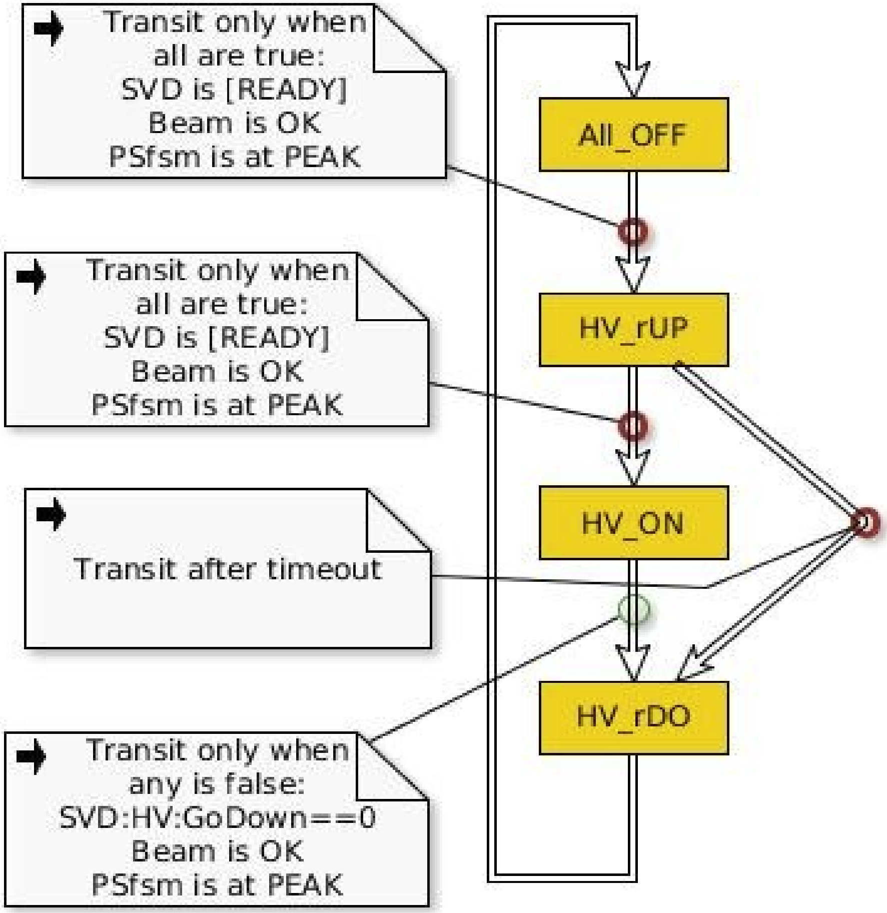
Do I have to care about cooling?
Do we need time out between IB_ON and FADC_INIT?

Startup sequence (2)

Preparing for injection	Turn off separation voltage	SEP_OFF			
	Ramp DOWN HV	HV_OFF			
	//injection is allowed				
	Ramp UP HV	HV_ON	APV_INIT		Beam is down or stable
	Turn on separation voltage	SEP_ON	HV_ON		
Turning OFF	Turn off separation voltage	SEP_OFF		PS(PEAK->any)	
	Ramp DOWN HV	HV_OFF	SEP_OFF		
	Shut down APV	APV_OFF	HV_OFF?		
	Shut down FADC	FADC_OFF	APV_OFF		
	ramp down LV	LV_OFF	FADC_OFF		
	shut down DQM	DQM_OFF			
	Turn off Cooling	CO2_OFF	LV_OFF		
	Shut down all the PCs running EPICS	SC_OFF			

Startup sequence (3)





Transit only when all are true:
SVD is [READY]
Beam is OK
PSfsm is at PEAK

Transit only when all are true:
SVD is [READY]
Beam is OK
PSfsm is at PEAK

Transit after timeout

Transit only when any is false:
SVD:HV:GoDown==0
Beam is OK
PSfsm is at PEAK

For more data, refer to Twiki:
https://belle2.cc.kek.jp/~twiki/bin/viewauth/Detector/SVD/SVD_control

NTC

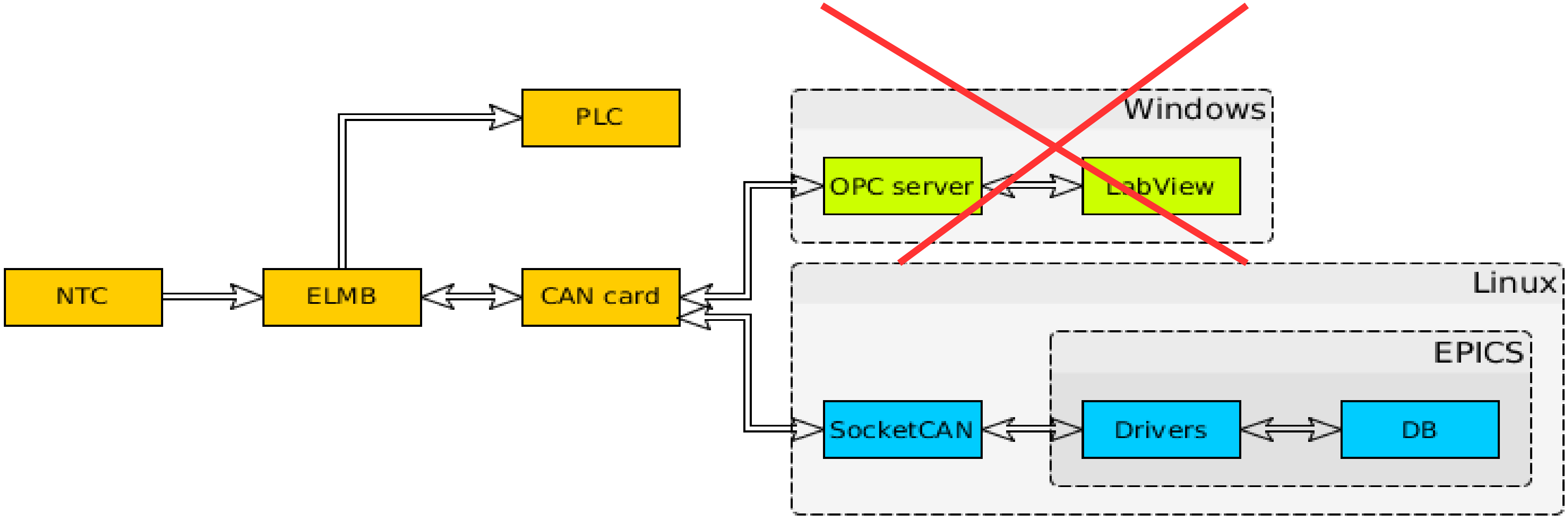
- LabView software is being replaced with EPICS
- Preliminary software works, but needs to be reimplemented in more elegant way.
- New version should be tested by next B2GM,

Choice of driver

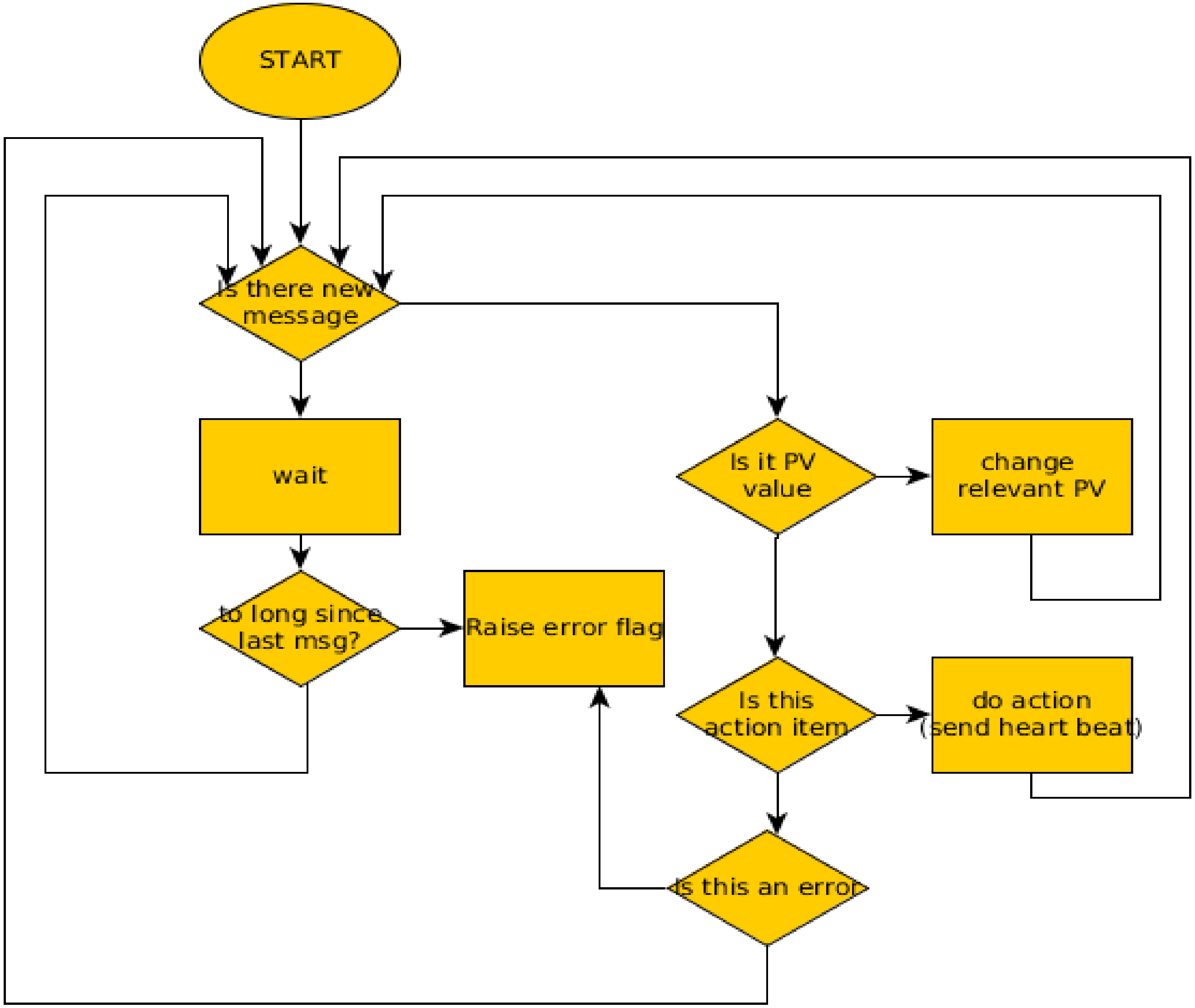
Name	Open	Kvaser USB	Kvaser PCI	Other Interfaces	Documentation
CAN4Linux	Yes	no	yes	yes	Very Good, some examples
LinuxCAN	Yes	yes	yes	no	Good, many examples
SocketCan	Yes	yes	yes	yes	Moderate. complicated **universal driver**

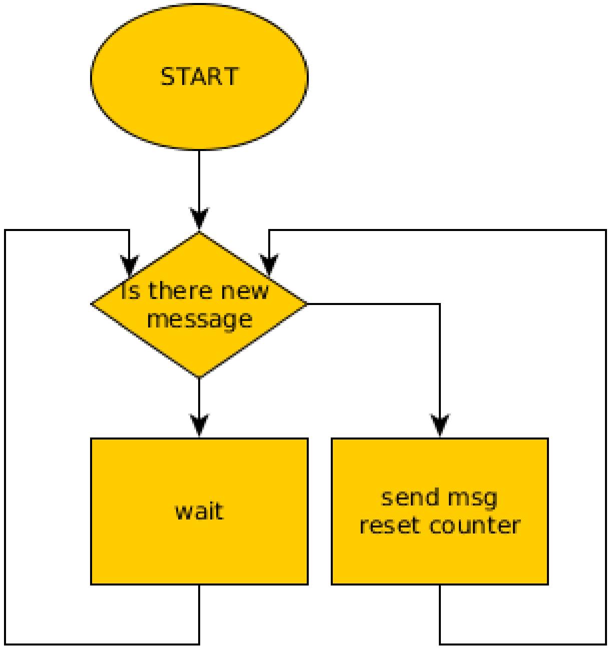
Choice: ~~LinuxCAN~~ SocketCan

System diagram



How the software looks

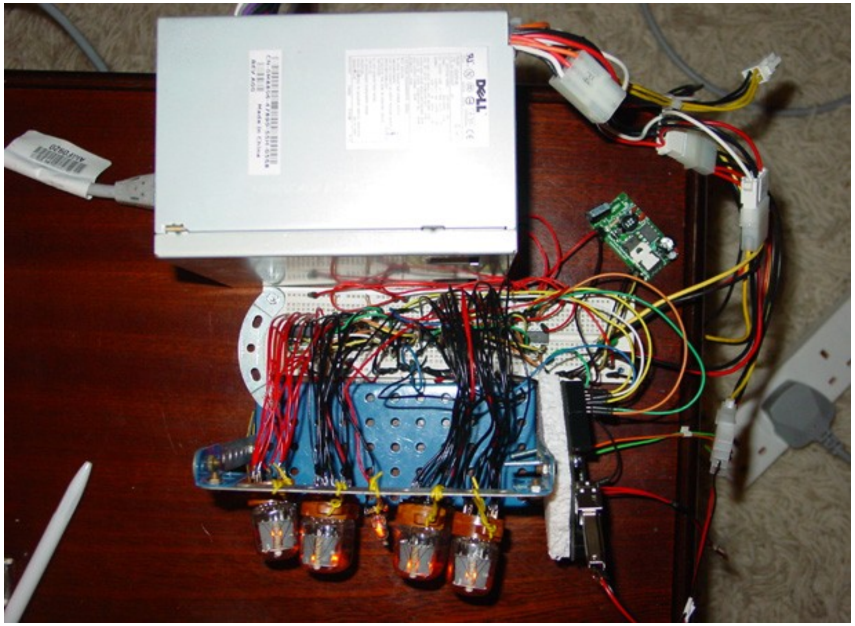




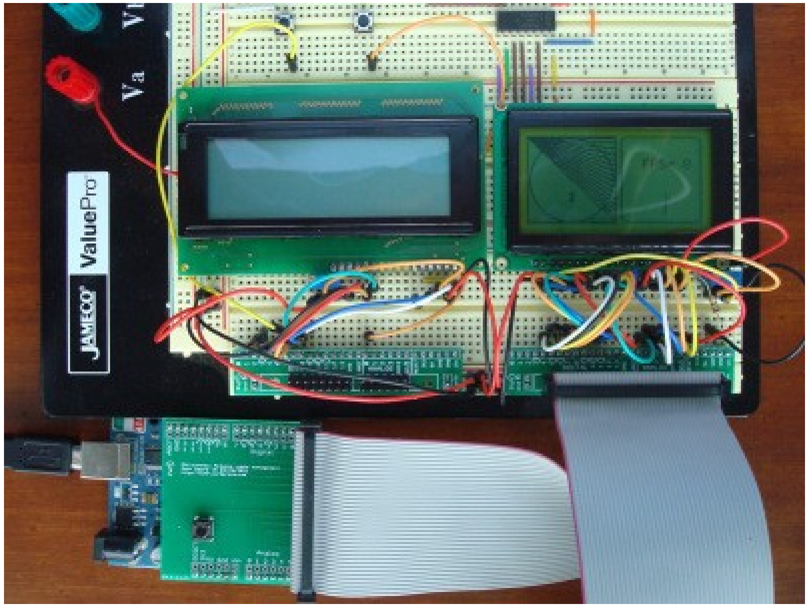
Implementation status

- LinuxCan implementation works, but is ugly, and ~~will get~~ got difficult to maintain.
- SocketCAN+SNL implementation is ongoing.

How it is now:



How it should be



http://forum.arduino.cc/index.php?topic=57951.0

Benefits of SocketCAN+SNL

- Easier to read/maintain: only two files to investigate
- Universal SocketCAN works on every CAN interface I could find
- SocketCAN is big and mature project
- SNL is easily understood by any EPICS user

NTC Time line

- Summer: familiarization with new coding techniques.
- Early fall (before B2GM): Preparing minimalistic state machines.
- Late fall, winter: determining errors, required “OPC-like” functionalities; building draft GUI.

Task to tackle by test beam

Task	T_Szymon [months]	T_Hao [months]
NTC software	1	0
SVD_QM	1	1
FADC_CTRL	1	2
Power Supplies	0,5	0,1
Startup_sequencer	1	0,5
GUIs	1	1
TOTAL:	5,5	4,6

Months till test beam: 6!

Time line

- New NTC software to be ready by B2GM
- Any new input about startup has to come before B2GM
- Work on SVD_QM will be continued after Hao`s thesis defense

Thank you