MSCB advantages and PSI specific implementations

Stefan Ritt

Paul Scherrer Institute, Switzerland

Overview

Slow Control = DAQ at 10ms … 10s

- Temperatures, Pressures, High Voltages, …

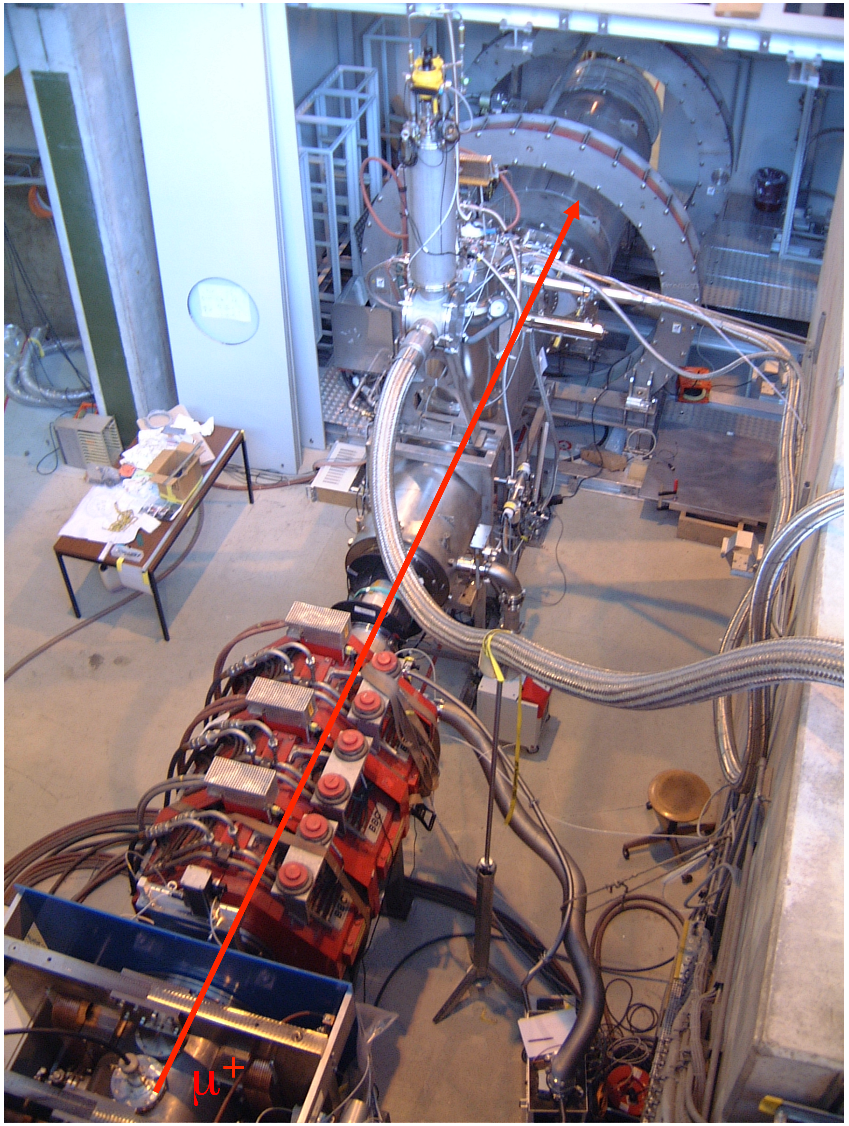
MSCB

- Midas Slow Control Bus
- Developed at PSI since 2001 mainly for MEG

This Talk

- Short introduction
- Specific hardware solutions
- Software overview (LabView)
- "Informal" talk: Ask questions, start discussion

MEG Overview



- Search for $\mu \rightarrow e\,\gamma$ down to 10^{-13}
- 80 People, 11 MCAD
- R & D started in 2000, data taking in 2007-2010
- Complex detector system (liquid Xenon calorimeter, superconducting magnets)
- Long term stability

→ Demanding slow control system

Univ. of Tokyo
Y. Hisamatsu, T. Iwamoto, T. Mashimo, S. Mihara, **T. Mori**, Y. Morita, H. Natori, H. Nishiguchi, Y. Nishimura, W. Ootani, R. Sawada, Y. Uchiyama, S. Yamashita

KEK
T. Haruyama, K. Kasami, A. Maki, Y. Makida, A. Yamamoto, K. Yoshimura

Waseda Univ.
K. Deguchi, T. Doke, J. Kikuchi, S. Suzuki, K. Terasawa

INFN Pisa
A. Baldini, C. Bemporad, F. Cei, L.del Frate, L. Galli, G. Gallucci, M. Grassi, F. Morsani, D. Nicolò, A. Papa, R. Pazzi, F. Raffaelli, F. Sergiampietri, G. Signorelli

INFN and Univ. of Genova
S. Cuneo, D. Bondi, S. Dussoni, F. Gatti, S. Minutoli, P. Musico, P. Ottonello, R. Valle

INFN and Univ. of Pavia
O.Barnaba, G. Boca, P. W. Cattaneo, G. Cecchet, A. De Bari, P. Liguori, G. Musitelli, R. Nardò, M. Rossella, A.Vicini

INFN and Univ. of Roma I
A. Barchiesi, D. Zanello

INFN and Univ. of Lecce
M. Panareo

Paul Scherrer Institute
J. Egger, M. Hildebrandt, P.-R. Kettle, **S. Ritt**, M. Schneebeli

BINP Novosibirsk
L. M. Barkov, A. A. Grebenuk, D. N. Grigoriev, B. I. Khazin, N. M. Ryskulov

JINR Dubna
A. Korenchenko, N. Kravchuk, A. Moiseenko, D. Mzavia

Univ. of California, Irvine
W. Molzon, M. Hebert, P. Huwe, J. Perry, V. Tumakov, F. Xiao, S. Yamada

Subsystems

Beam line magnet

- 2 valves, 10 temperature sensors
- communication with LHe plant and quench control (24V signals)

COBRA Magnet

- 40 temperature sensors, communication with quench control (GPIB)

Beamline

- 14 magnets (EPICS-like)

NaI mover

- Two ultrasonic stepping motors

LXe system

- ~100 valves, flow meters, pressure sensors
- Capacitive level meters

DC gas system

- Similar to TWIST (~1Pa diff. pressure regulation)

High Voltage

- 1000 channels PMT
- 32 channels drift chamber

Air conditioning

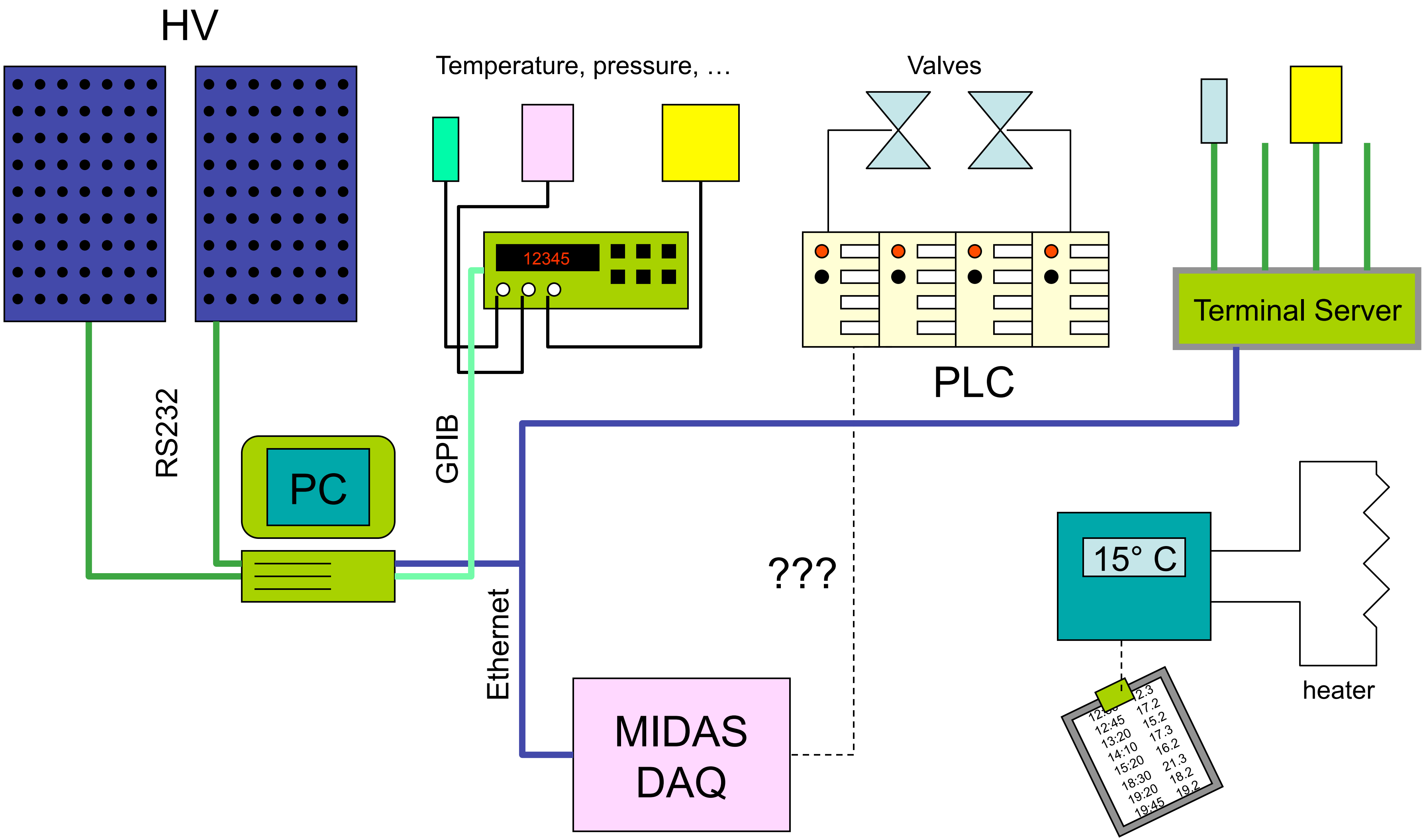
VME crates

- Fans, voltages, temperatures

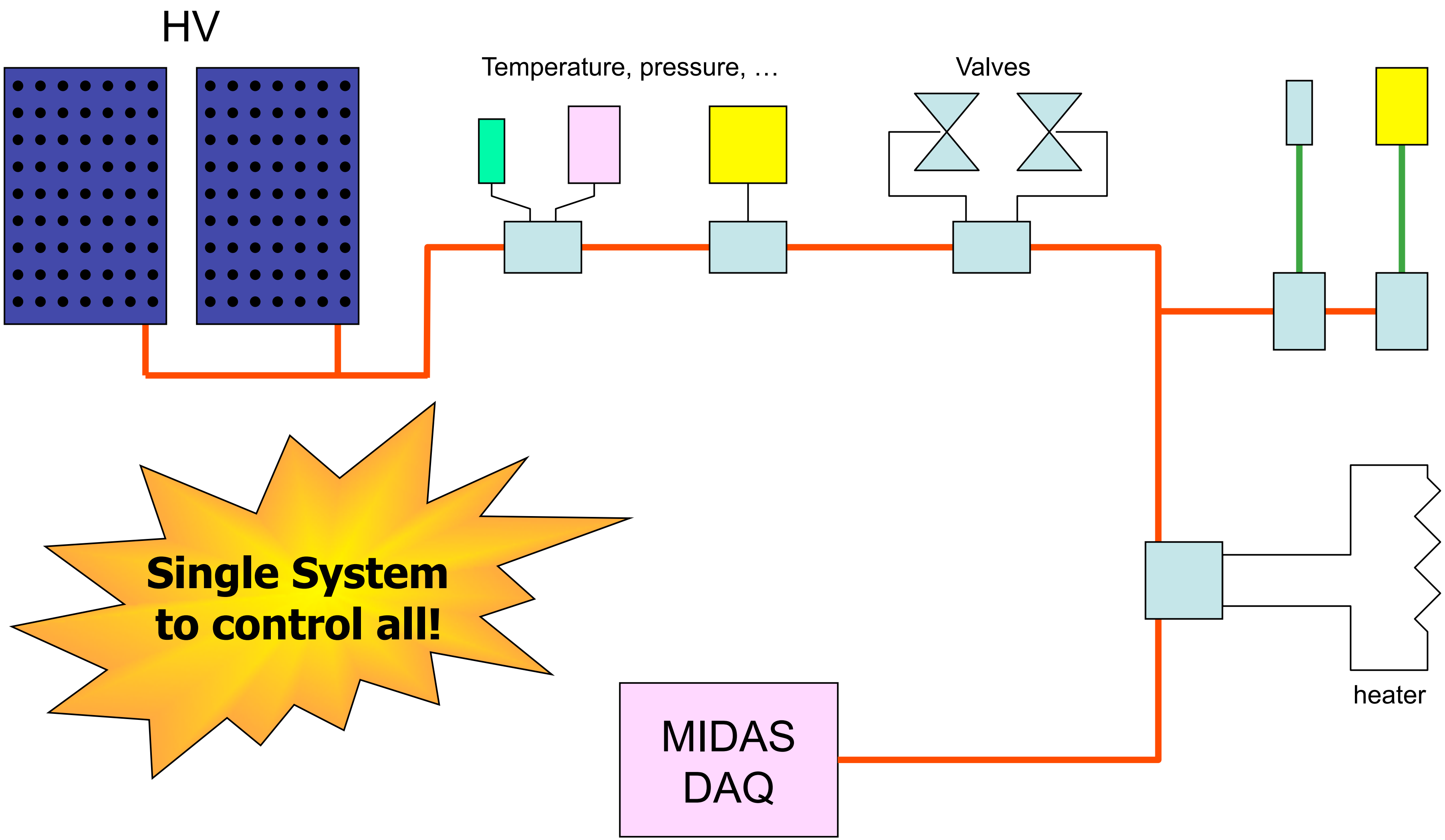
Cooling water

- 10 secondary circuits

Traditional Slow Control



Single Slow Control System



A long and winding road

Various demands in an experiment are pretty demanding (inhomogeneity, stability, ease of use)

It took finally three iterations to make a good system

- Many lessons learned
- Some unusable hardware produced
- Project started in 2001, now (kind of) finished

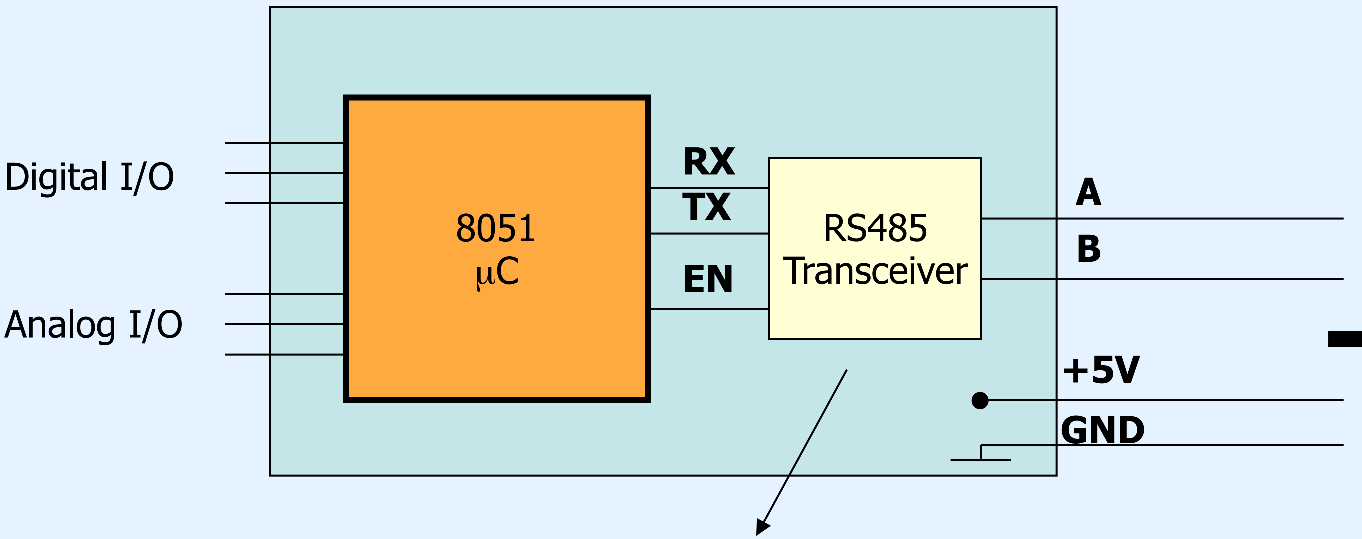
We have now a very good and flexible system

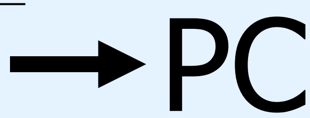
- Used in MEG, μSR, SLS, PEN at PSI
- Can be extended very easily

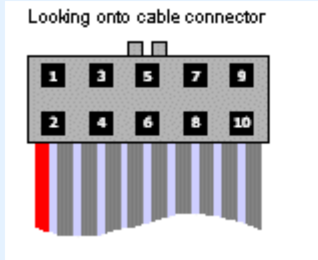
First version MSCB system

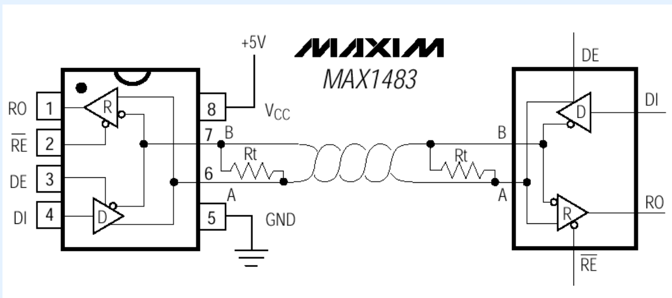
New generation of 8-bit microcontrollers with analog I/O

RS-485 communication of hundreds of meters







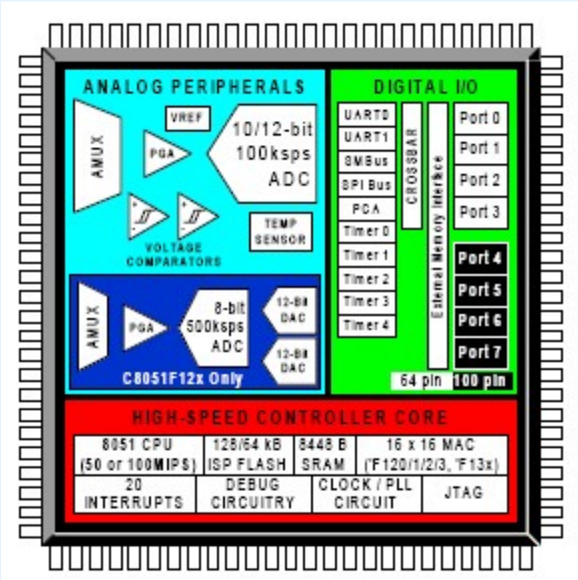


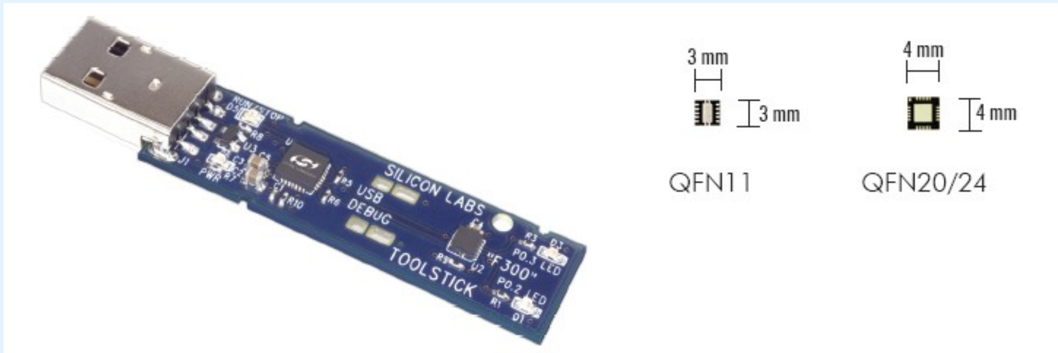
Up to 256 nodes in parallel connected to RS-485 bus

C8051Fxxx uC from Silicon Laboratories

Huge variety of mixed signal microcontrollers

Part Number	MIPS (peak)	Flash Memory (bytes)	RAM (bytes)	Digital Port I/O Pins	Serial Buses	Internal Osc	ADC1	DAC	Temp Sensor	VREF	Other	Package
C8051F000	20	32 KB	256	32	UART, SMBus, SPI	±20%	12-bit, 8ch., 100ksps	12-bit, 2ch.	Y	Y	-	TQFP64
C8051F005	25	32 KB	2304	32	UART, SMBus, SPI	±20%	12-bit, 8ch., 100ksps	12-bit, 2ch.	Y	Y	-	TQFP64
C8051F020	25	64 KB	4352	64	2 UARTs, SMBus, SPI	±20%	12-bit, 8ch., 100ksps	12-bit, 2ch.	Y	Y	-	TQFP100
C8051F040	25	64 KB	4352	64	CAN2.0B, 2 UARTs, SMBus, SPI	±2%	12-bit, 13ch., 100ksps	12-bit, 2ch.	Y	Y	±60V PGA	TQFP100
C8051F064	25	64 KB	4352	59	2 UARTs, SMBus, SPI	±2%	16-bit, 1ch., 1Msps	-	-	Y	DMA	TQFP100
C8051F121	100	128 KB	8448	32	2 UARTs, SMBus, SPI	±2%	12-bit, 8ch., 100ksps	12-bit, 2ch.	Y	Y	16x16 MAC	TQFP64
C8051F300	25	8 KB	256	8	UART, SMBus	±2%	8-bit, 8ch., 500ksps	-	Y	-	-	MLP11
C8051F320	25	16 kB	2304	25	USB 2.0, UART, SMBus, SPI	±1.5%	10-bit, 17ch., 200ksps	-	Y	Y	-	LQFP32
C8051F340	48	64 kB	5376	40	USB 2.0, 2 x UART, SMBus, SPI	±1.5%	10-bit, 17ch., 200ksps	-	Y	Y	-	TQFP48
C8051F410	50	32 kB	2304	24	UART, SMBus, SPI	±2%	12-bit, 24ch., 200ksps	12-bit, 2ch.	Y	Y	Volt Reg, RTC	LQFP32





MSCB Protocol

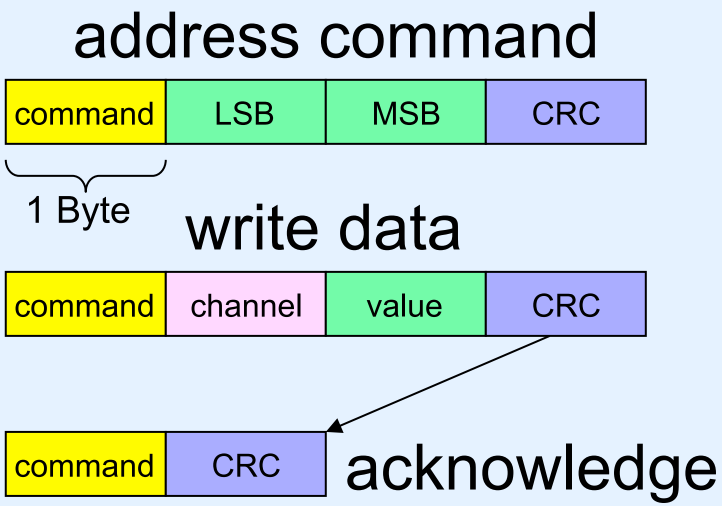
Asynchronous 115 kBaud

16-bit addressing (64k nodes), CRC-code, acknowledge

Concept of typed "network variables"

Optimized protocol: 300 reads/sec.

Firmware upgradeable over MSCB bus

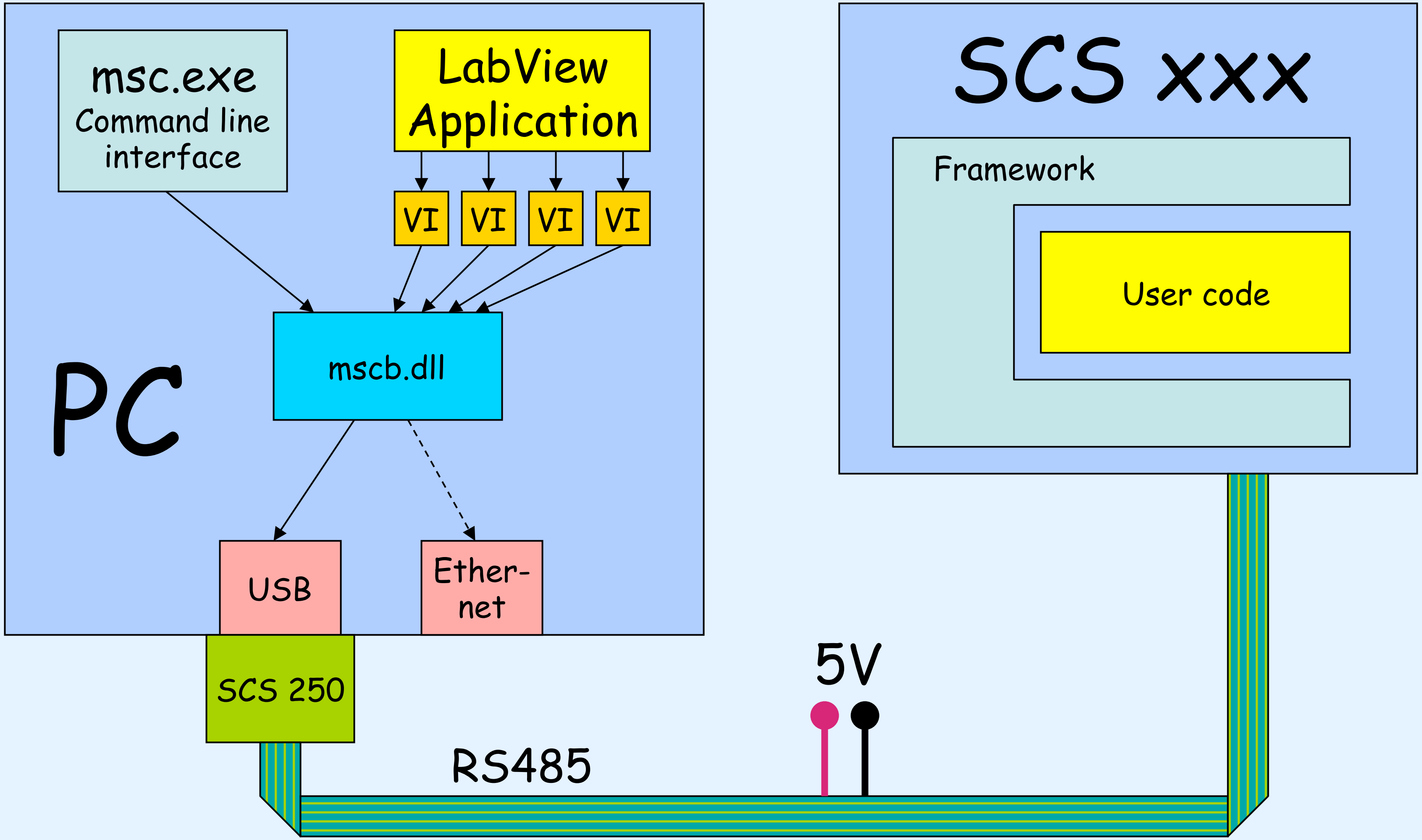


Node programming

```
struct {
  float adc;
  float dac;
} user_data;

main()
{
  …
  user_data.adc = read_adc(0);
  write_dac(user_data.dac);
  …
}
```

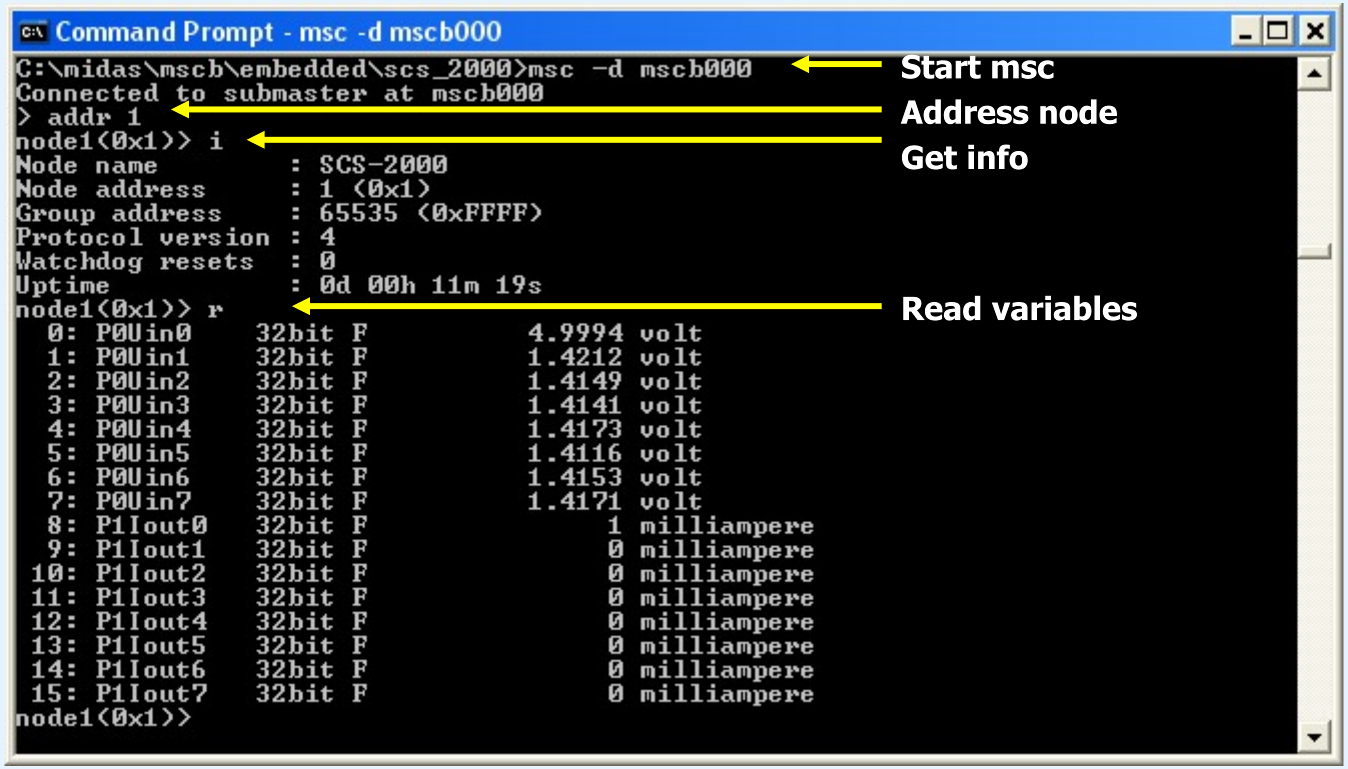
Software overview

Easy application development due to powerful framework™ (MIDAS!)



MSCB Command Line Interface

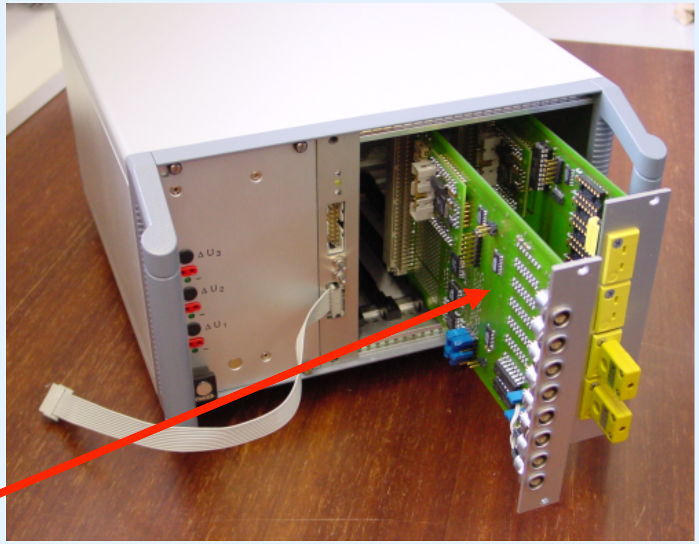
Simple ASCII CLI under Windows and Linux as a human interface to the mscb C library

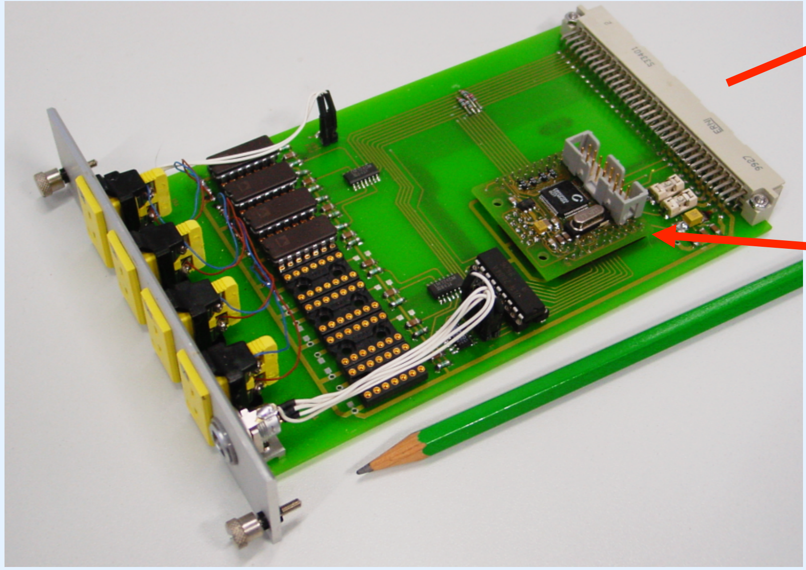


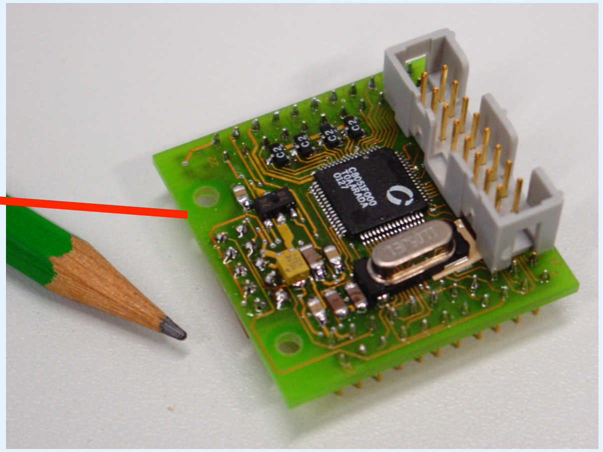
First Version

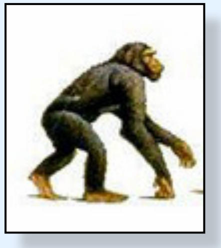
3HE Card with piggy-back CPU

Various cards for digital and analog I/O









Problems with first version

System was stable and reliable, unless there were noisy environments

Low densisty (full slot for ~8 channels)

PC is always required for operation (MSCB only used as DAC/ADC)

- Labview sometimes crashes
- One PC had hard disk failure → LHe reservoir evaporated
- Replacement laptop did Windows update over night → LHe reservoir evaporated again ☹

No local display

Always crate needed

Difficult cabling (no outputs at back!)

Second MSCB Version

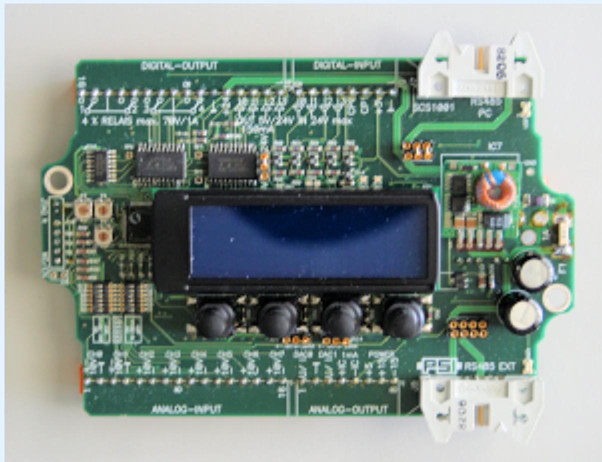
SCS-1000

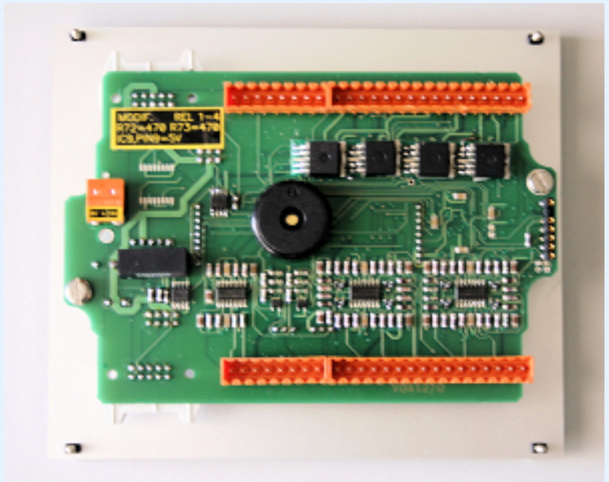
LCD, buttons, screw terminals

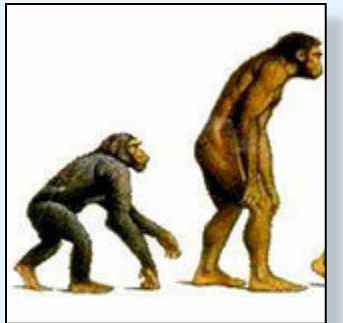
Rack mounted and standalone (24V)

8 analog in, 2 analog out, 4 Relais
4 digital in

Local application software

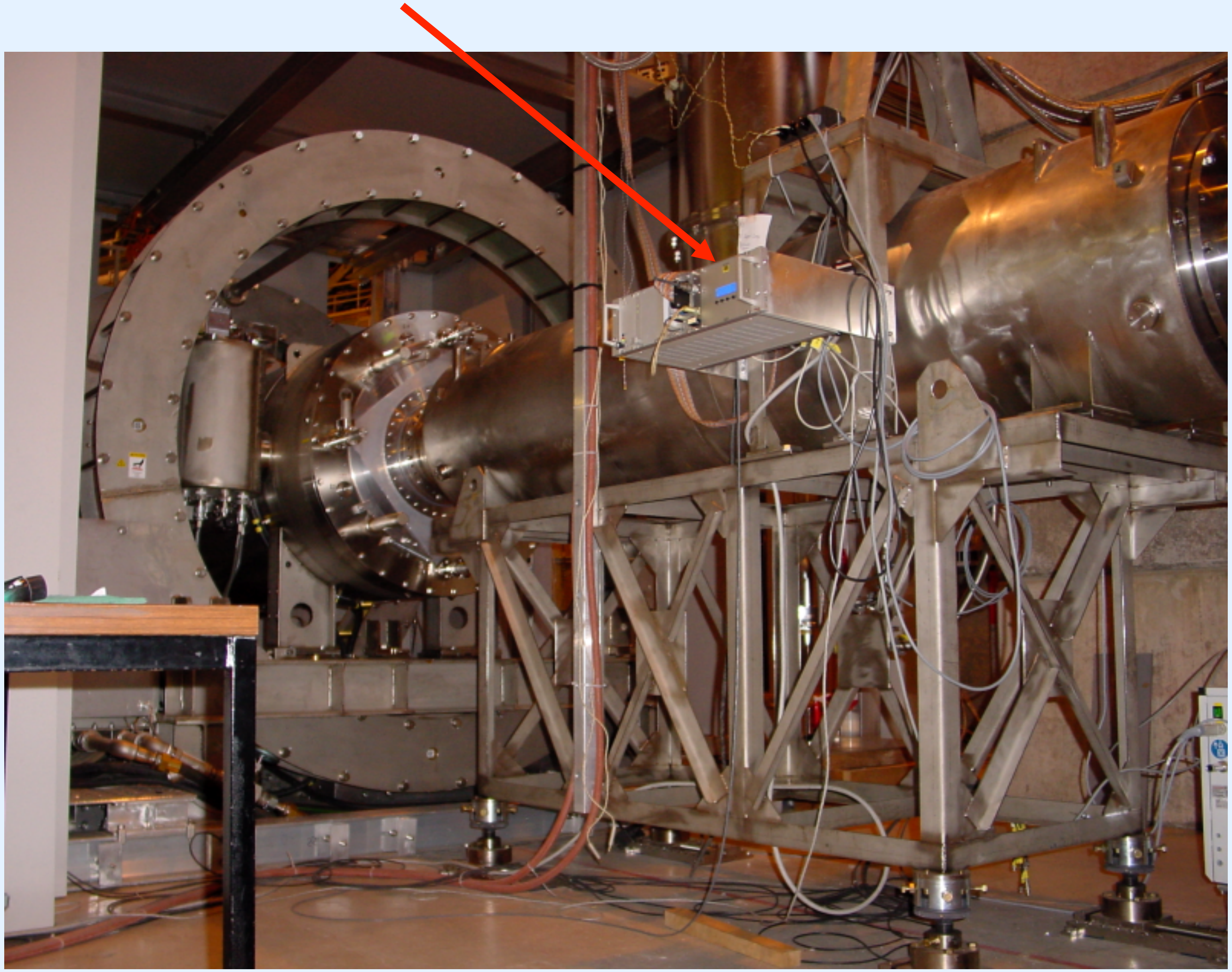






Magnet Control

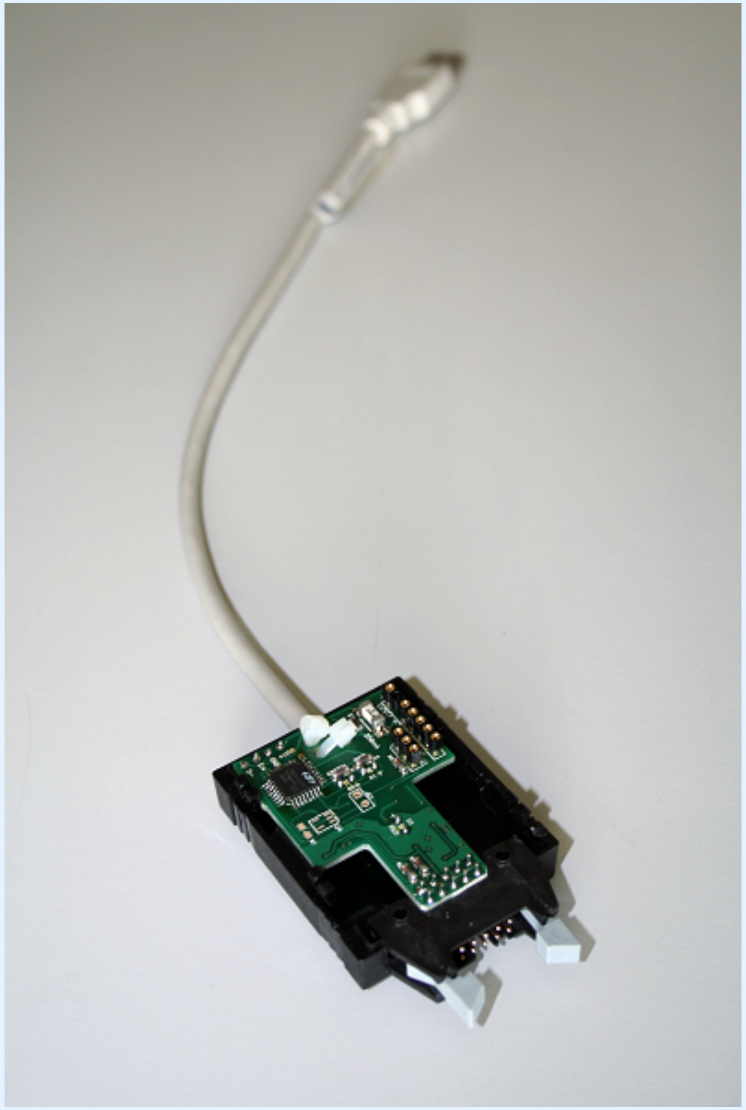
Standalone (non-PC) control of superconducting magnet



USB Submaster

USB submaster (SCS-250) replaced parallel port adapter. Drivers were written for Windows (difficult!) and Linux (easy but…)

5V/0.5A from USB can be used to power MSCB nodes over bus



Problems second version

Limited number of IOs

Microcontroller directly coupled to IOs

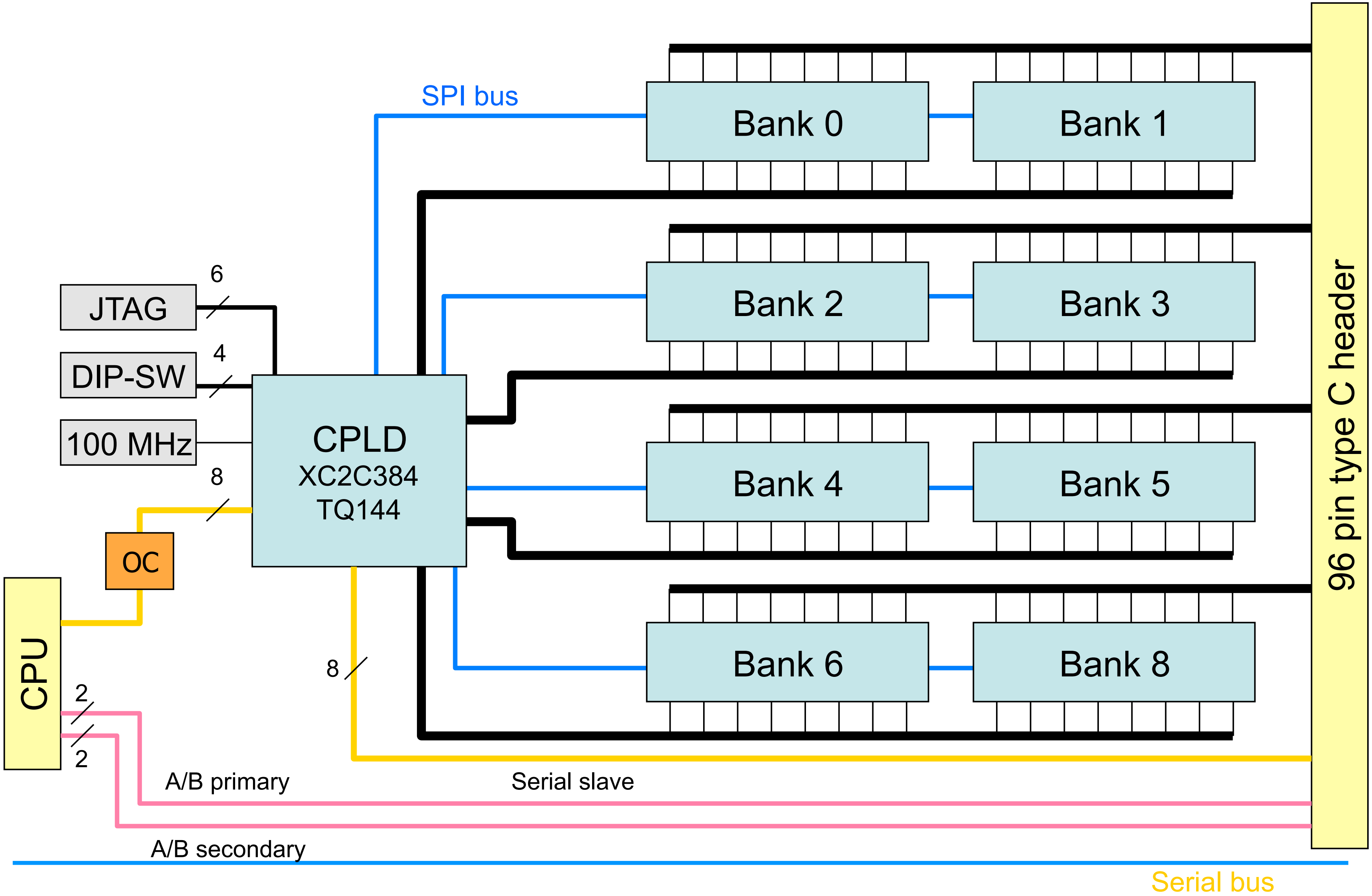
Outputs go high during reboot/firmware upgrade

DACs go to zero during reboot/firmware upgrade

No operation when CPU crashes

(optional) PC need physical connection to MSCB bus via USB

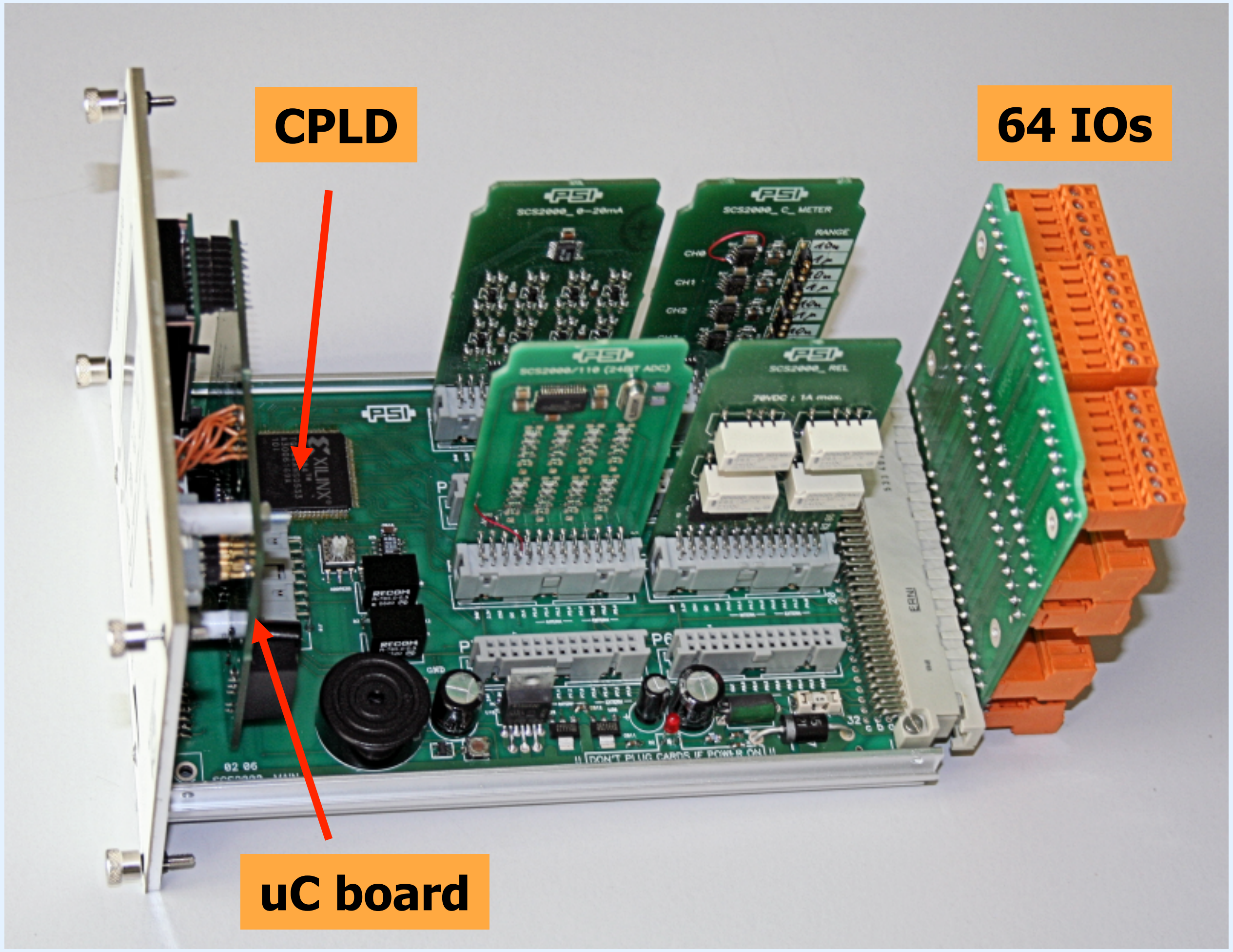
New concept

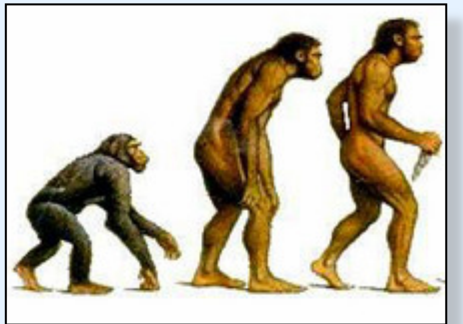


SPI bus
Bank 0
Bank 1
Bank 2
Bank 3
Bank 4
Bank 5
Bank 6
Bank 8
96 pin type C header
JTAG
6
DIP-SW
4
100 MHz
8
CPLD
XC2C384
TQ144
OC
CPU
2
2
8
A/B primary
Serial slave
A/B secondary
Serial bus

Third version

SCS-2000





PAUL SCHERRER INSTITUT

SCS-2000 Advantages

Flexible IO, 8 banks with 8 IOs each

Simple IO boards

CPU optically decoupled

Serial slave bus for daisy-chaining 16 SCS-2000

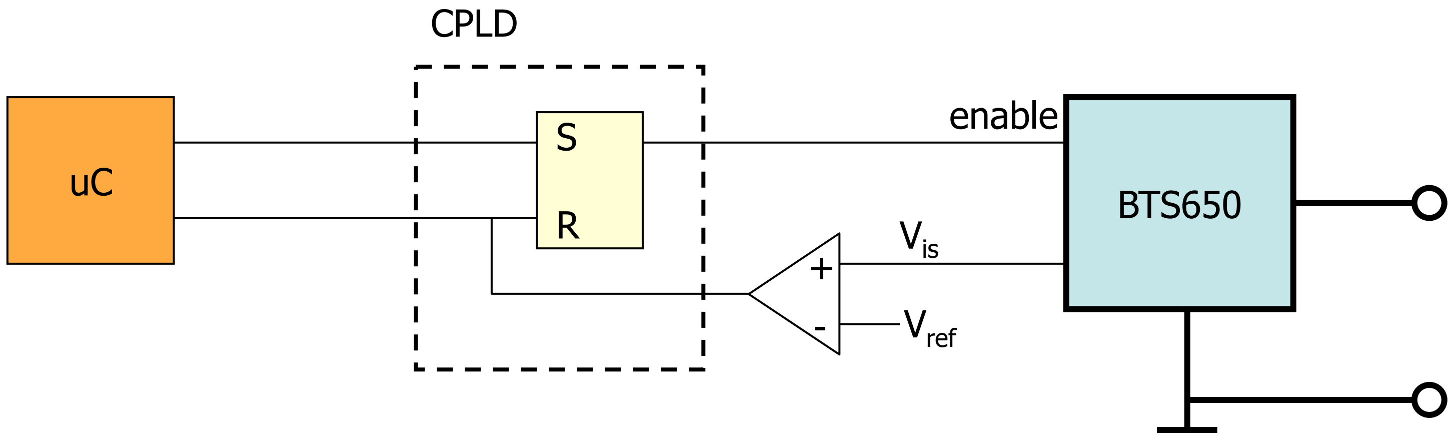
CPLD keeps state during reboot

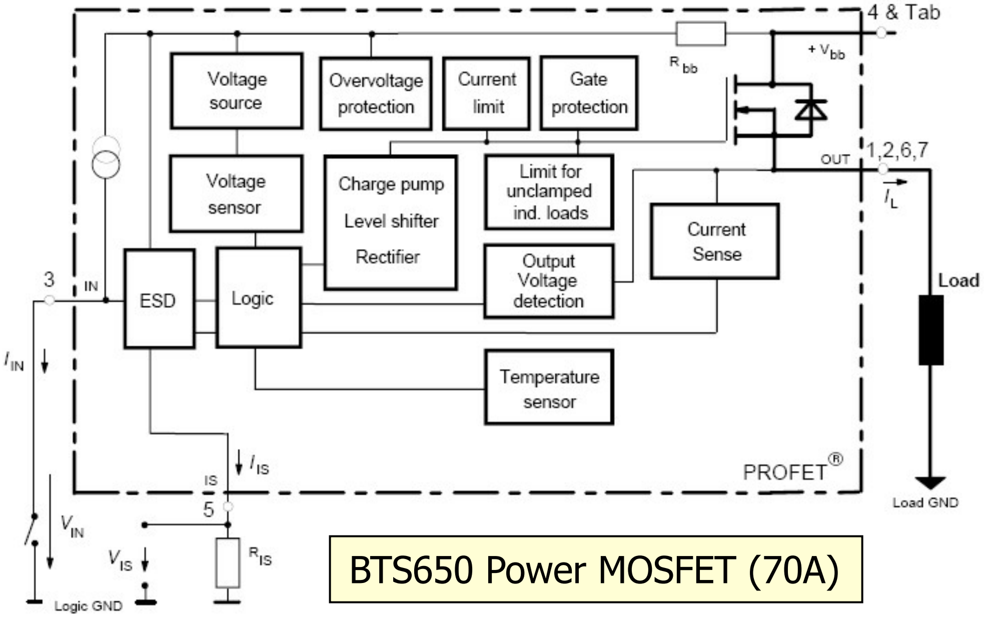
CPLD can do low-level tasks independent of CPU

CPLD can do fast tasks (100 MHz clock)

Soft-fuse

Soft-fuse

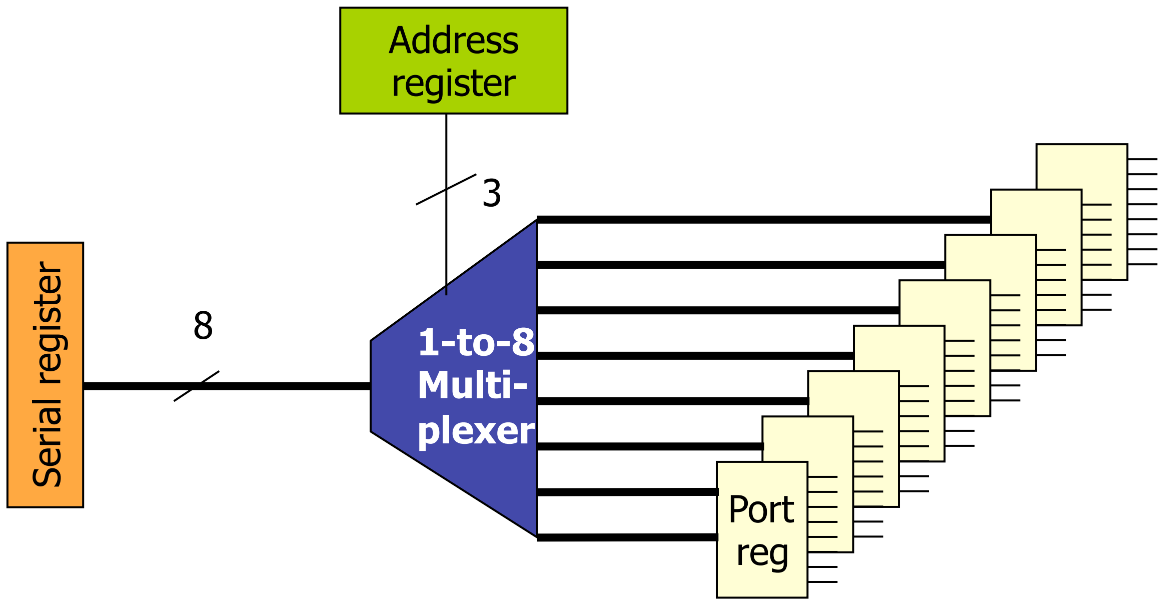




BTS650 Power MOSFET (70A)

No blown fuses in the middle of the night any more!

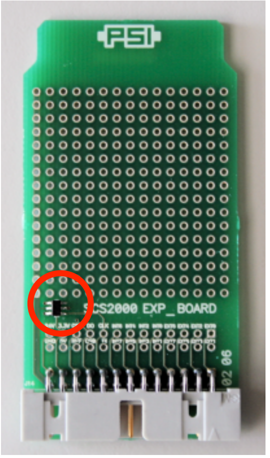
CPLD programming with VHDL



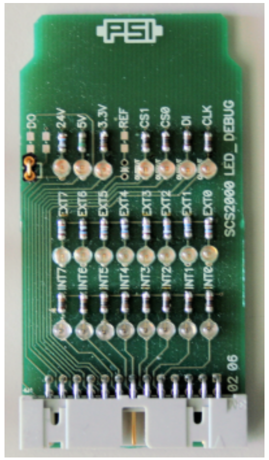

```
type type_port_reg is array (7 downto 0) of std_logic_vector(7 downto 0);
signal port_reg     : type_port_reg;
...
port_reg(CONV_INTEGER(addr_reg)) <= ser_reg;
```

Use VHDL even for CPLD programming !

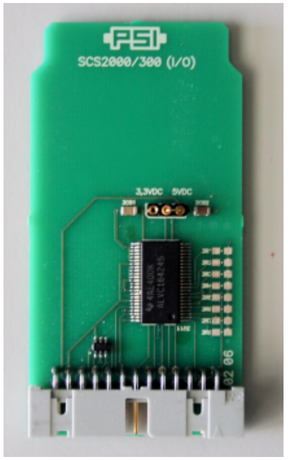
SCS-2000 IO Cards



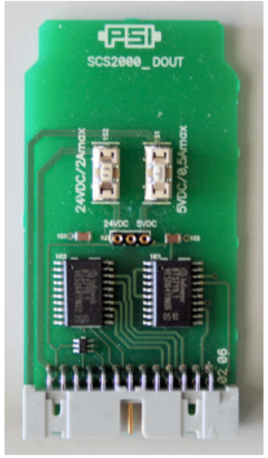
Experimental Board



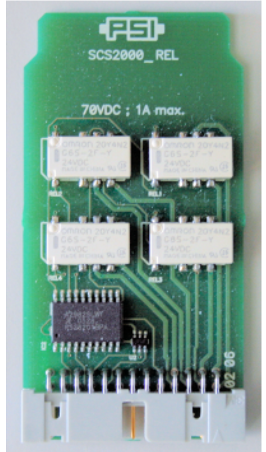
Debug Board



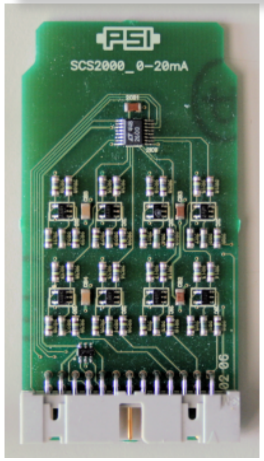
3.3V/5V
8 In/Out



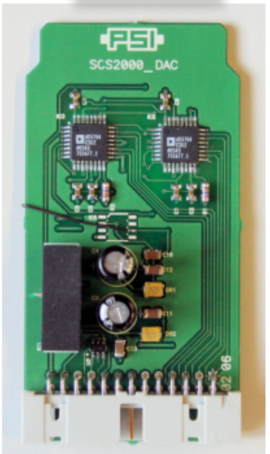
24V 2A
8 Out



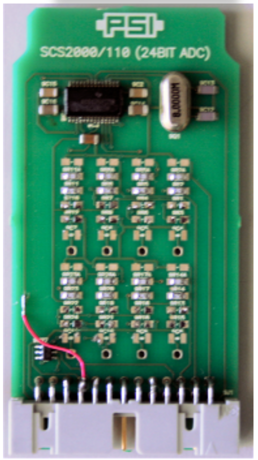
4 Relais



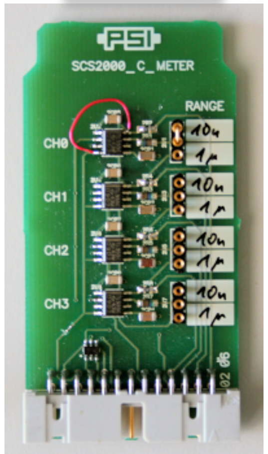
0-20mA
8 Out



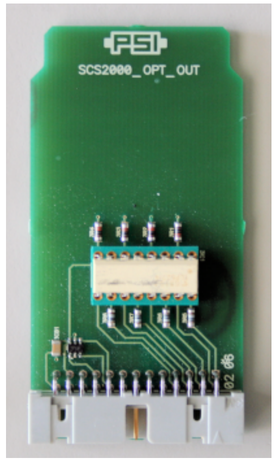
-10…+10V
8 Out



-10…+10V
0..2.5V
0..20mA
8x24bit In



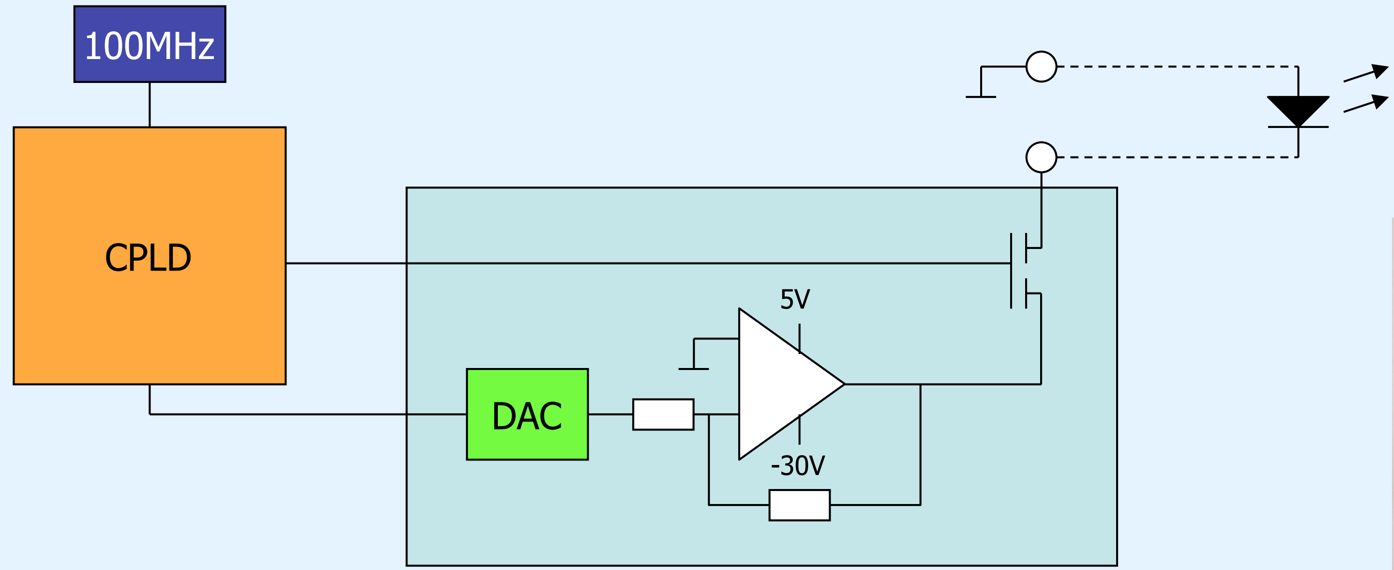
0…10nF
0…1uF
4 In

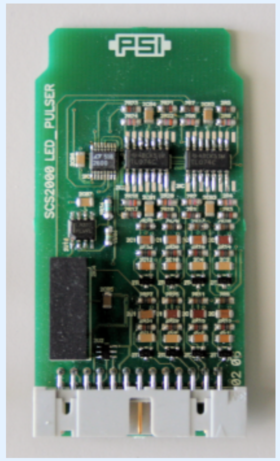


Opt.-Coupler
4 Out

LED pulser

Needed: LED pulser 0…-30V @ 50 Ohm, 30ns width, 100Hz – 1MHz repetition rate, triggerable



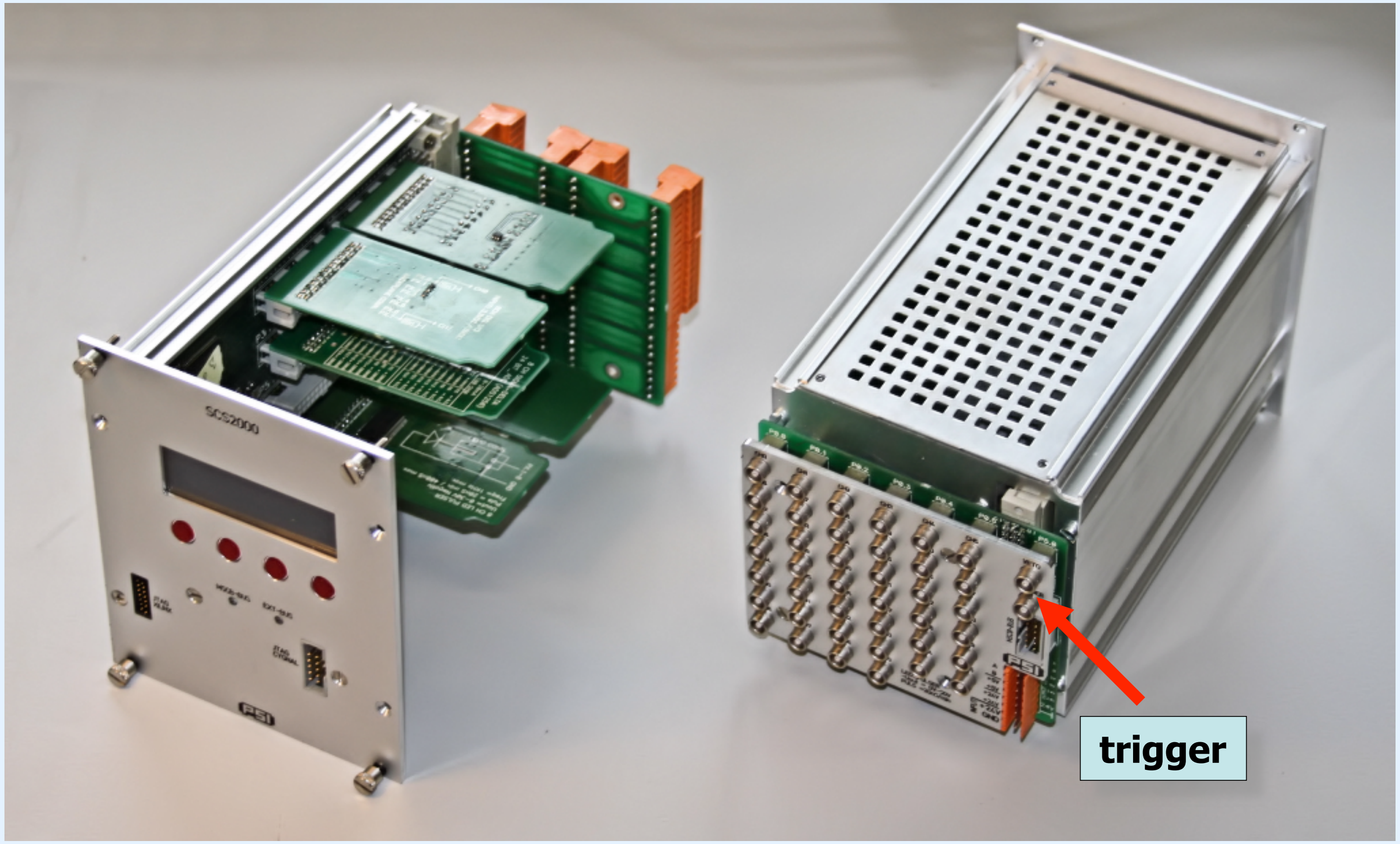


LED Pulser
8 Out

LED pulser box

Single SCS-2000 fits 40 channels LED pulser

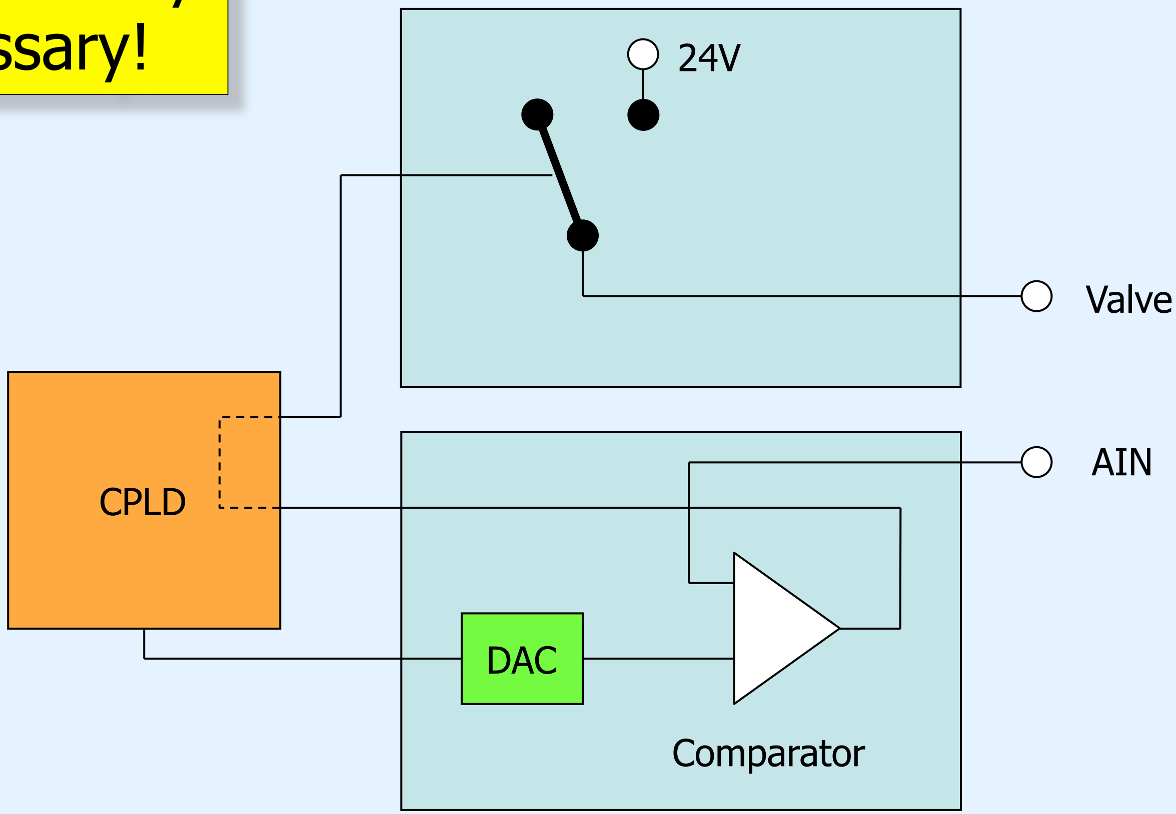
Either stand-alone operation or MSCB controlled



Example for low level control

CPLD can for example switch valve if pressure gets too high

No µC activity necessary!



PAUL SCHERRER INSTITUT

What else?

Other MSCB solutions used at PSI

SCS-260 Ethernet Submaster

Uses C8051F121 microcontroller @ 50 MHz

Cirrus CS8900A 10Base-T MAC chip

MICRONET TCP/IP stack from CMX

- ~6k CAD
- Full source code
- DHCP, TCP, UDP, HTTP

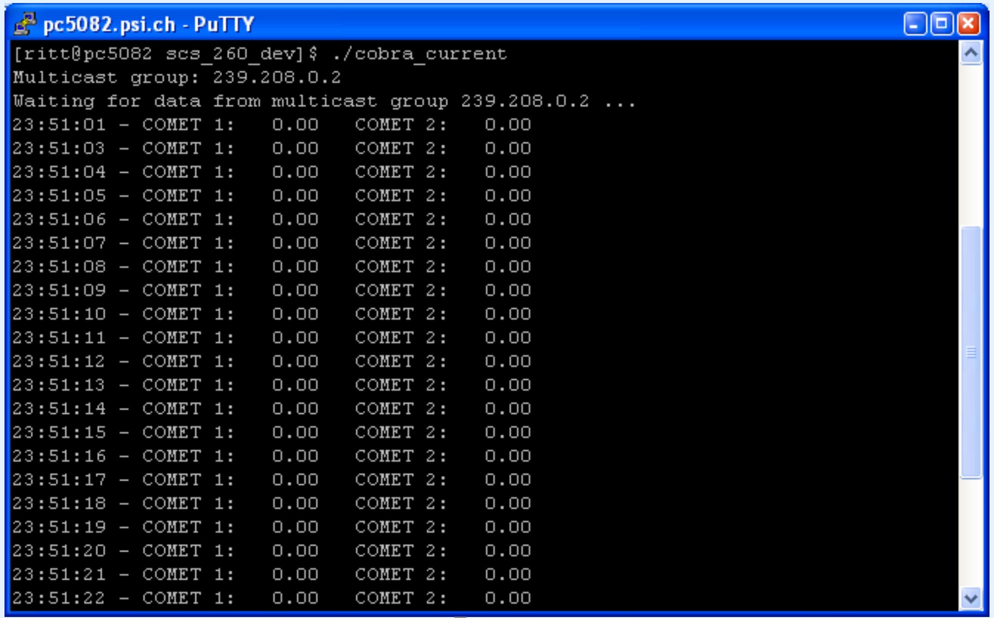
Had to request MAC addresses

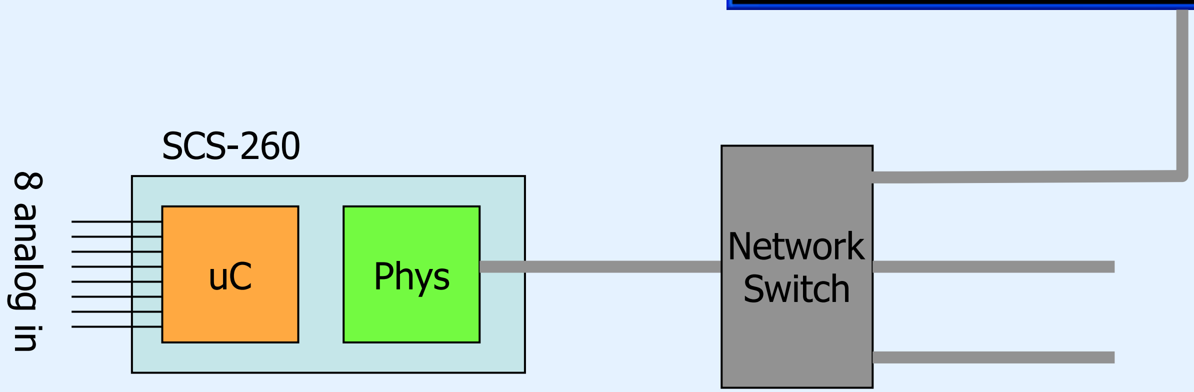
Boots in 100ms

Replaces more and more USB interface

SCS-260 as a beacon

- ADC on µC measures magnet current and distributes it through IP multicasts
- Multicasts are better than broadcasts (distributed only on request)
- Used now also for accelerator current status



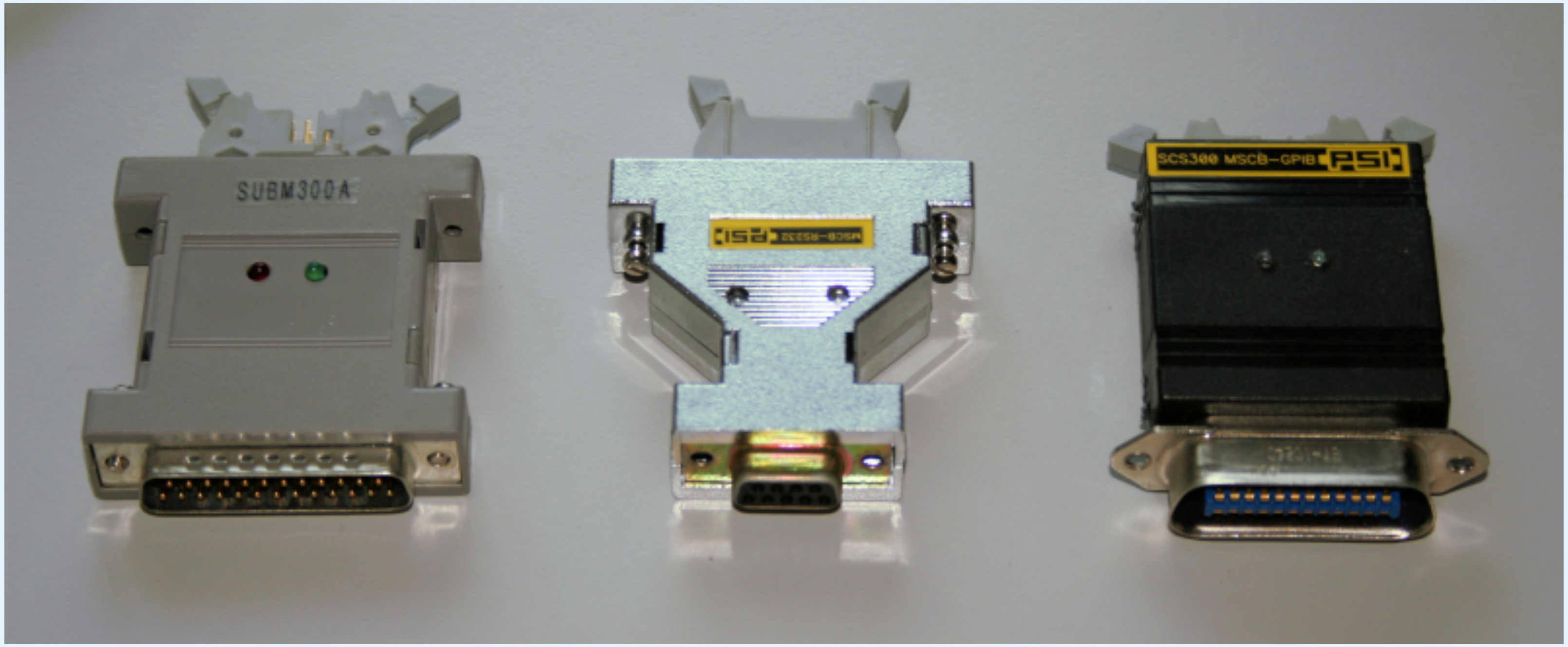


Interface nodes

Some experiment hardware has own controllers → need interfaces

Parallel (Centronics), RS-232 and GPIB adapter

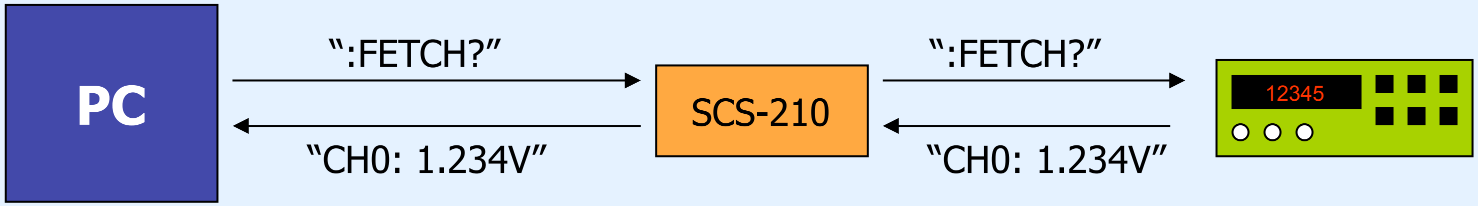
Work either string oriented or with local protocol handlers



Interface firmware

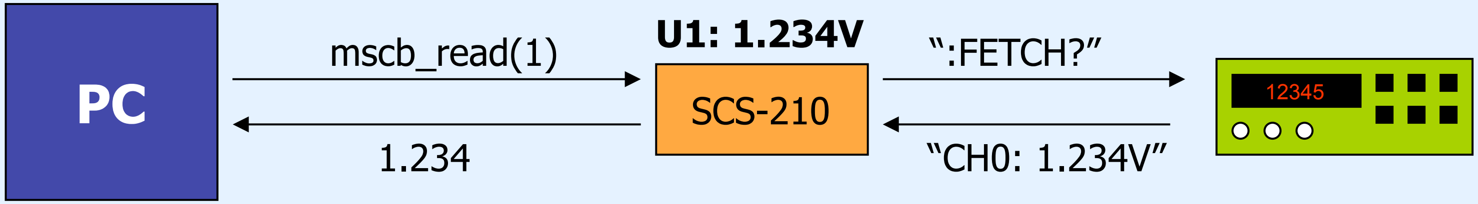
String oriented:

PC has to run protocol, MSCB node is device independent



Local protocol handler:

MSCB node runs protocol, PC is device independent

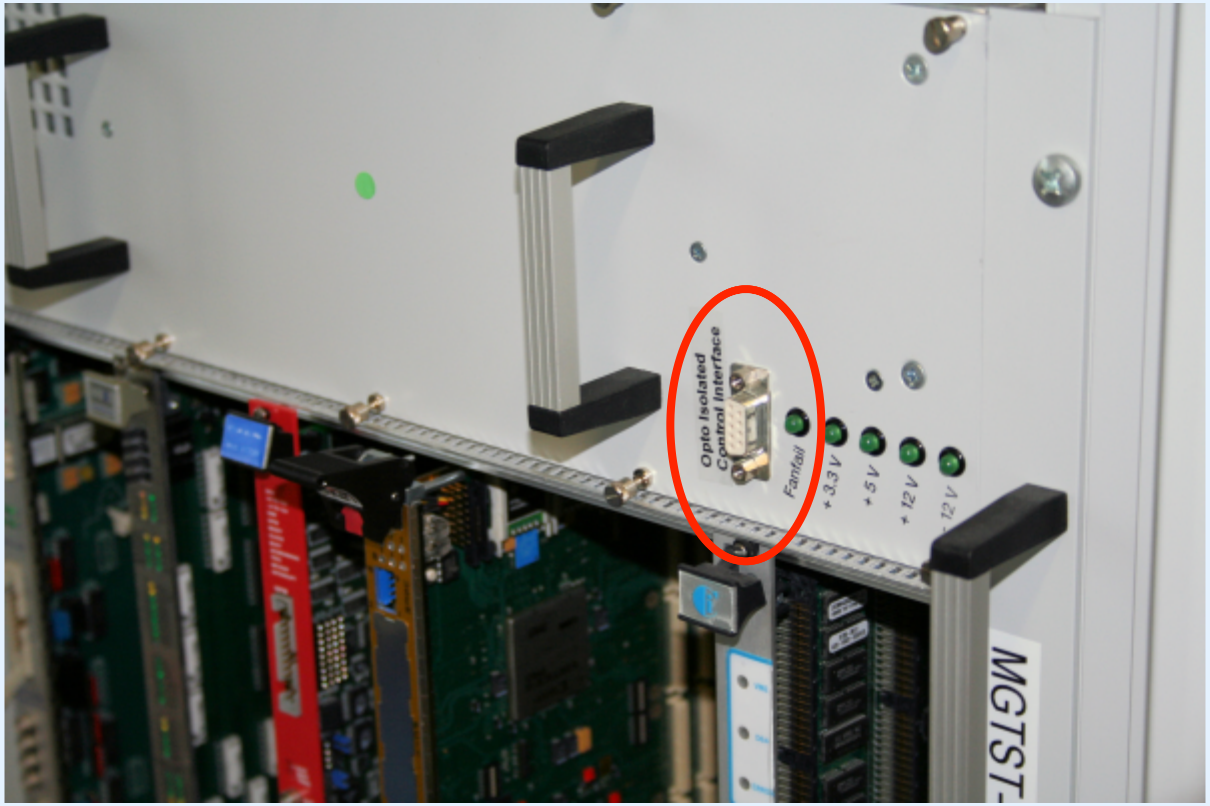


Crate controller requirements

VME crates have status bits and control bits (VME reset)

Current solution: CAN node (1000 CAD/crate + PC)

Also need
- temperature (currently not implemented)



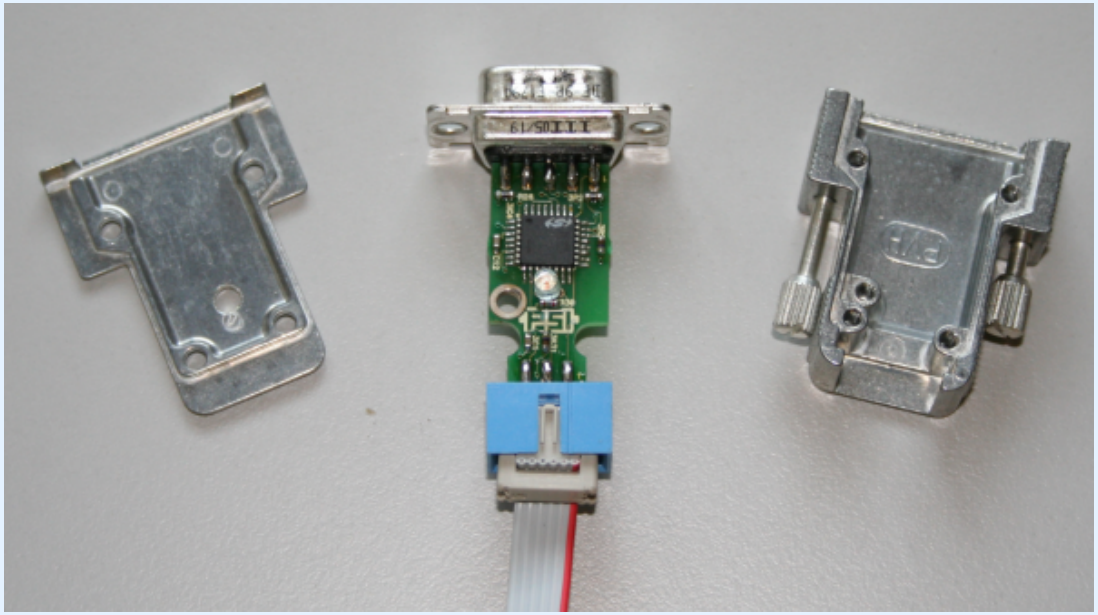
Solution with MSCB

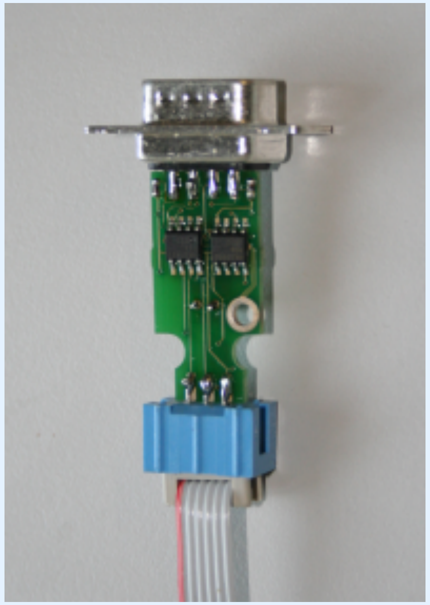
Put μC+RS-485+Temp. Sensor in 9-pin Sub-D connector

Power from MSCB bus, interfaced with Ethernet submaster

Took 1 day for engineer to design and couple of hours for me to program

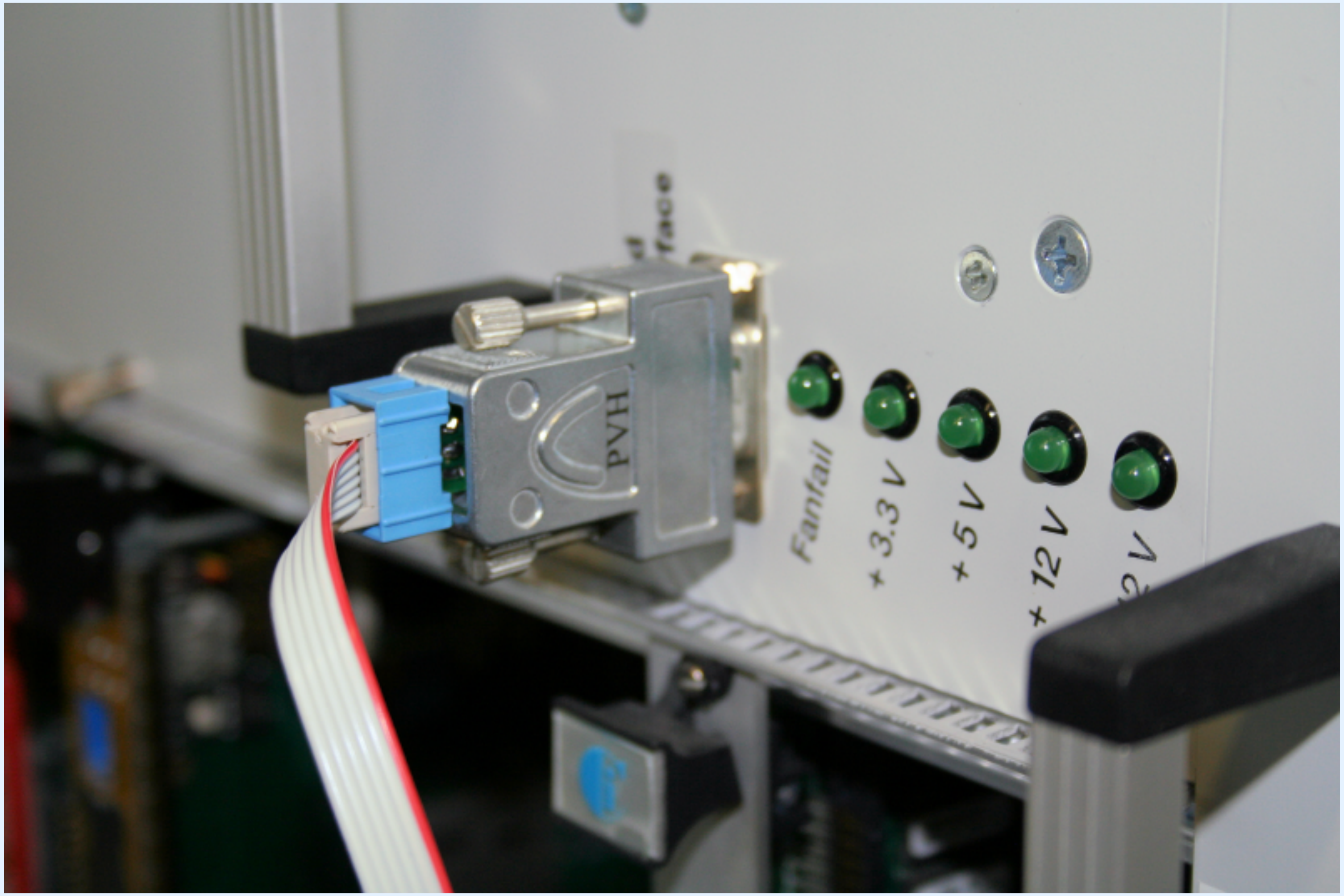
Costs 30 CAD





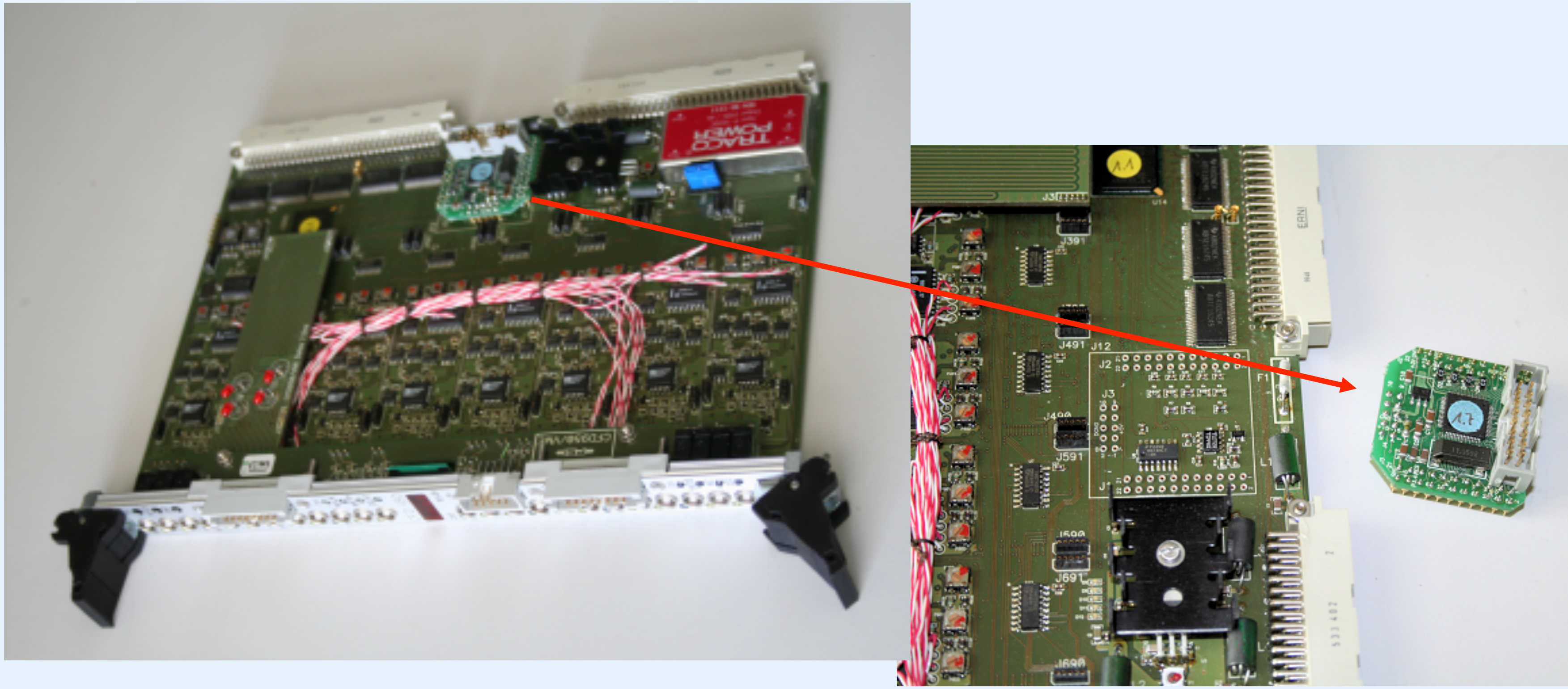
It fits!

SLS department installs this for ~100 crates

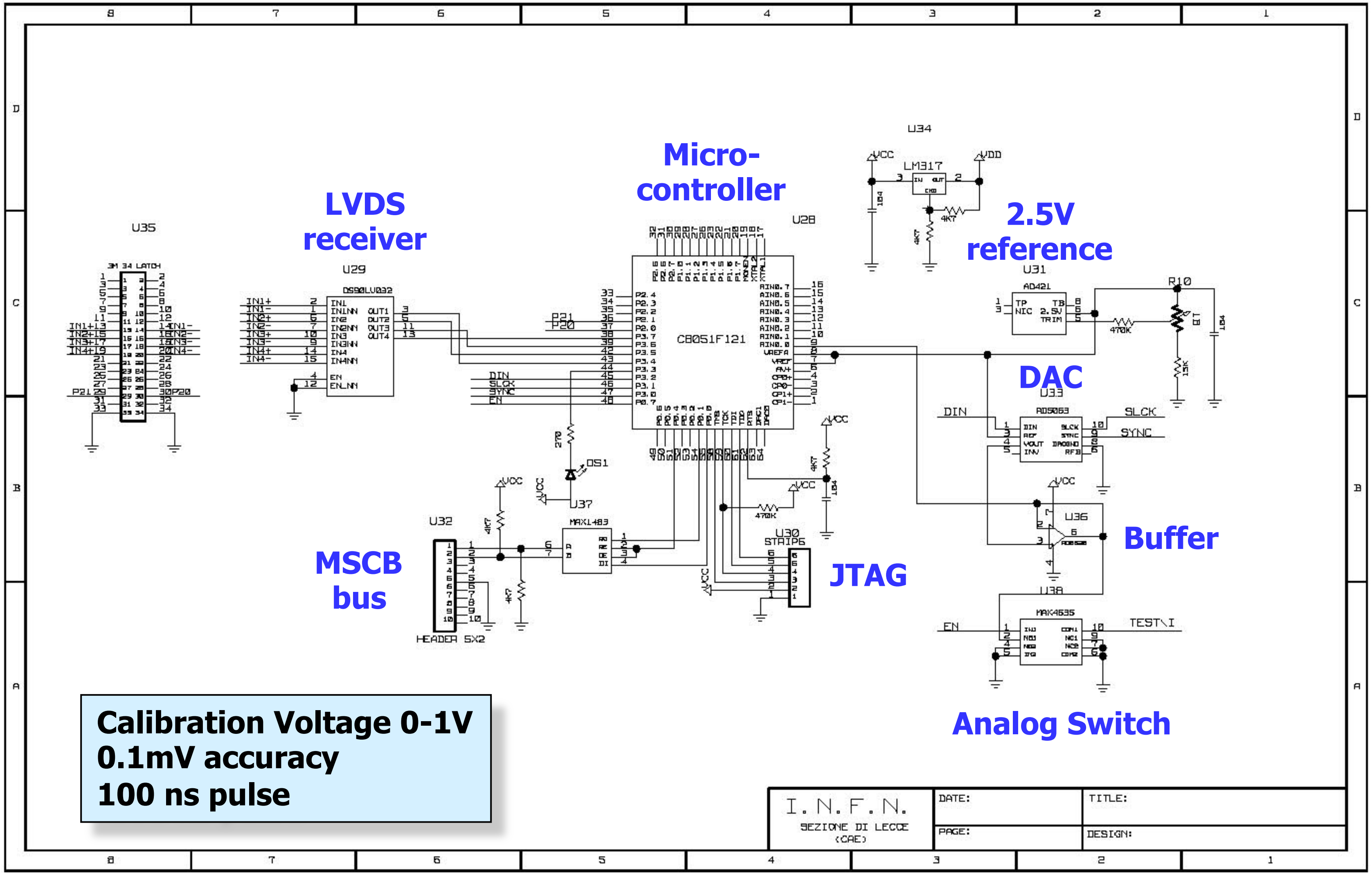


Piggy-back Controller

PSI-developed Constant-Fraction-Discriminator needed user interface to set delay lines and fractions as well as remote control

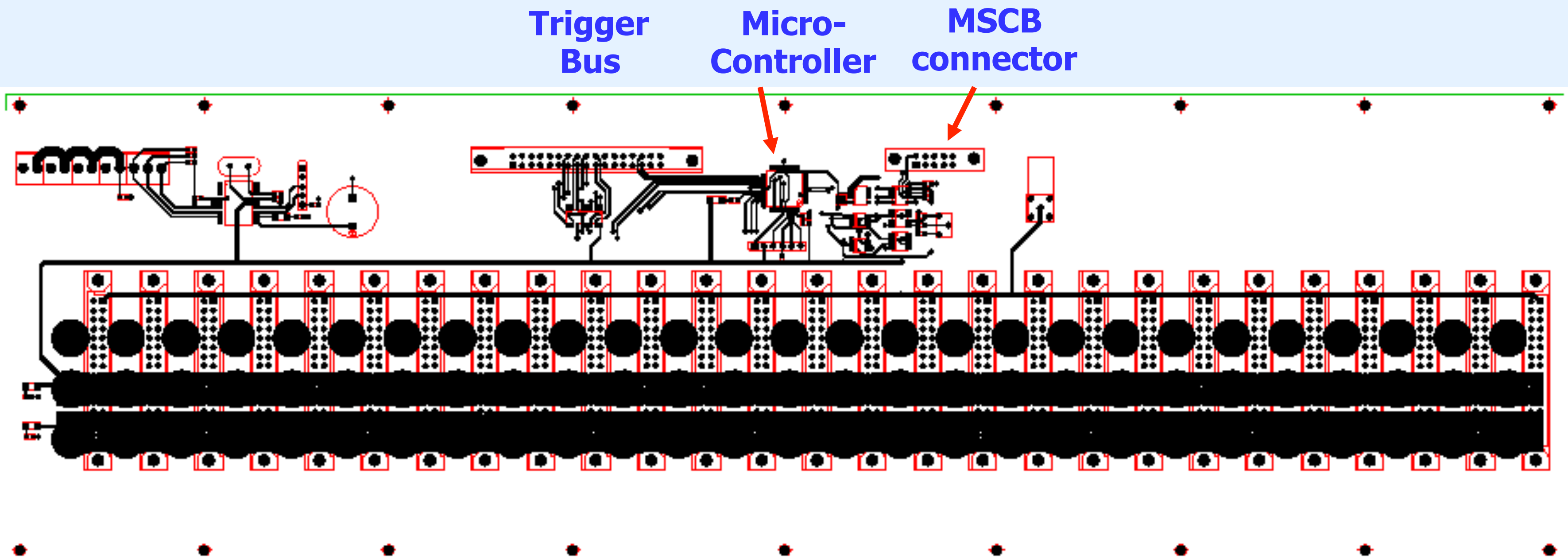


Precision Voltage Source



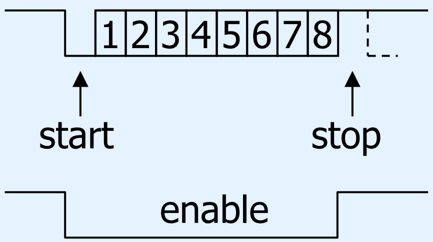
Calibration Voltage 0-1V
0.1mV accuracy
100 ns pulse

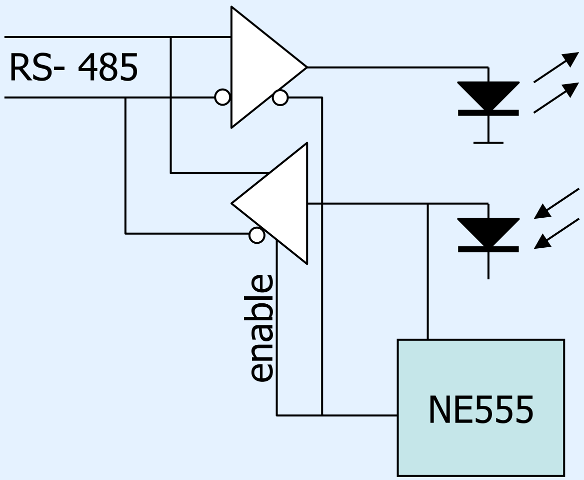
PCB Signal Splitter



Optical Link

Optical transceiver for >5kV insulation, necessary for electrostatic separator (200kV)

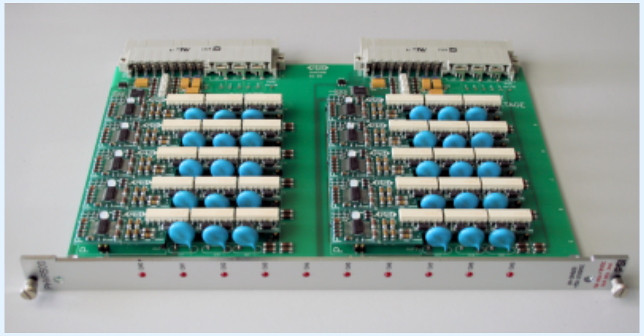


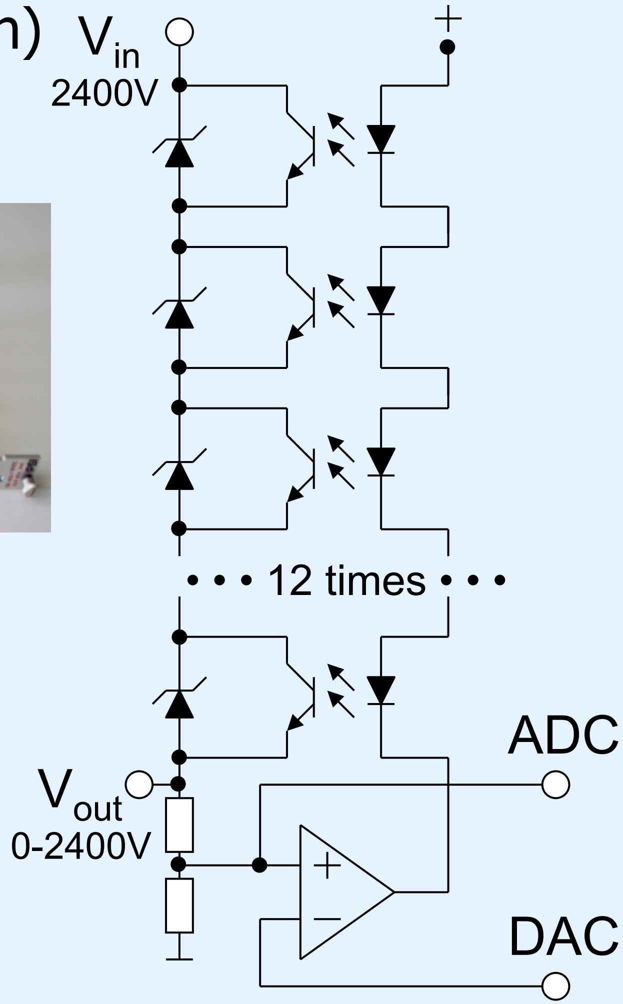


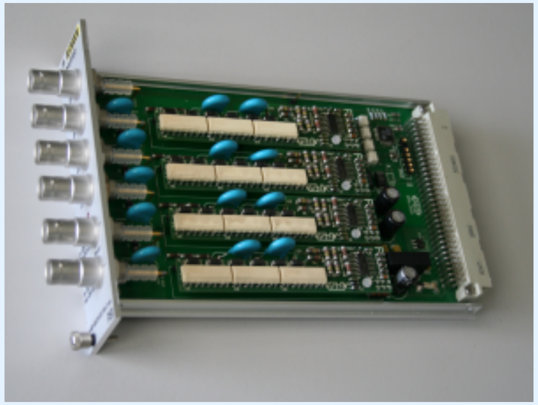
High Voltage Regulators

Very high accuracy (~10mV @ 1000V)

Low cost (~50 CAD/chn)

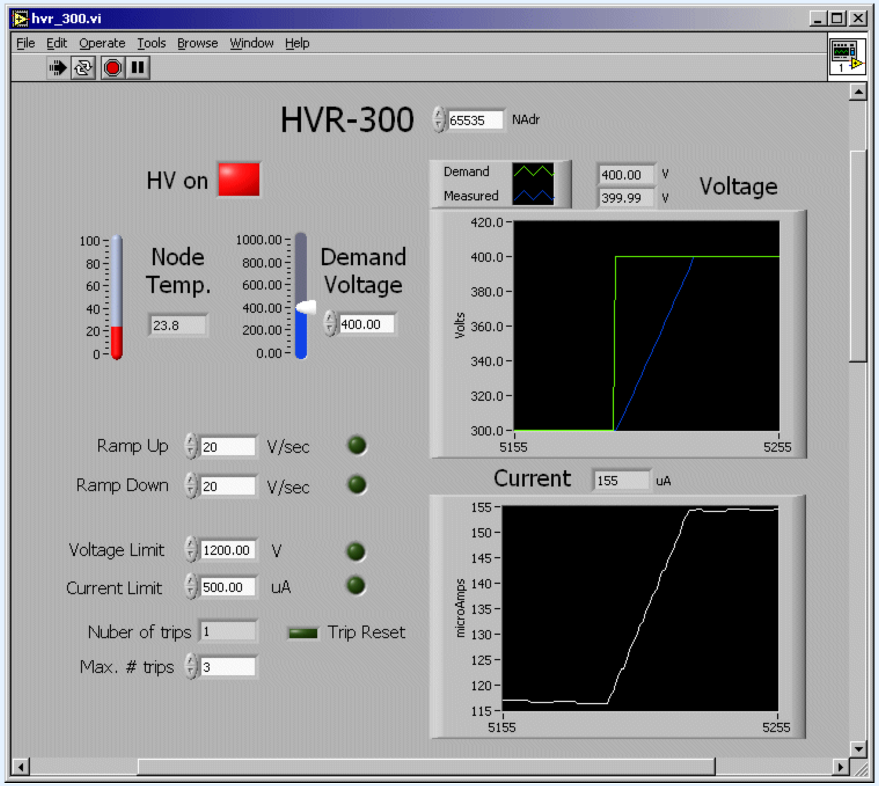


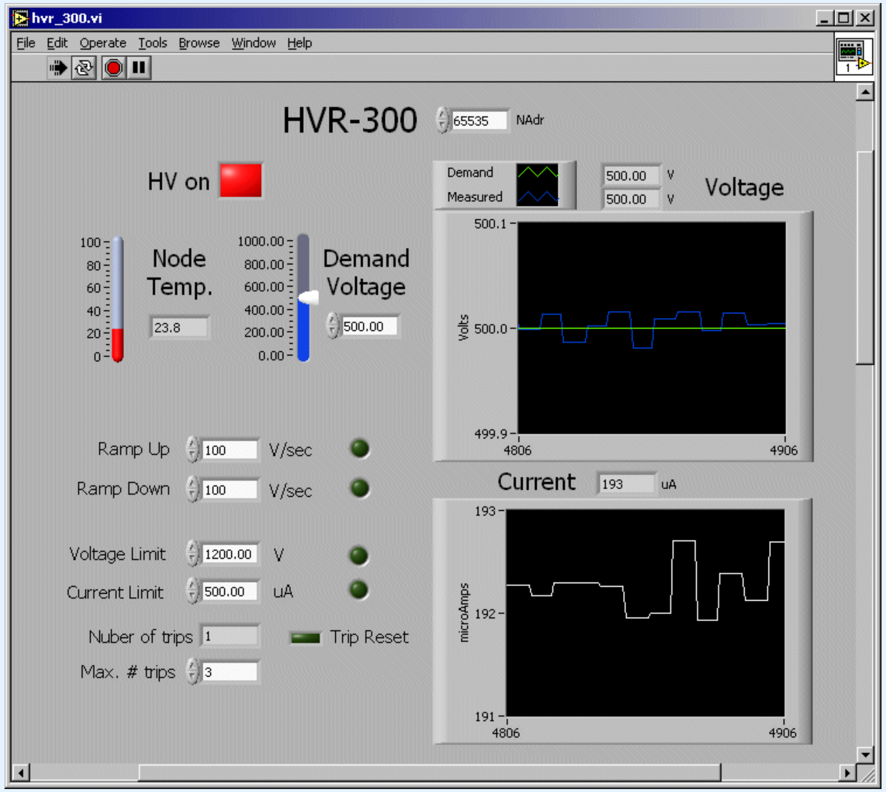




HV Control

Accuracy – ramping – current trip – trip reset



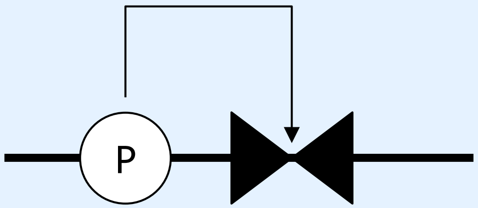


Software aspects

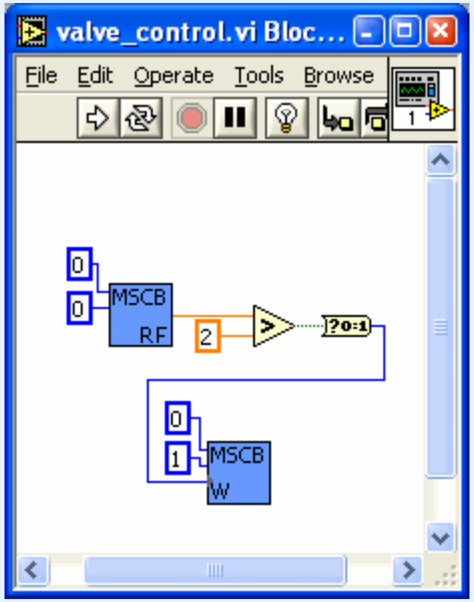
PAUL SCHERRER INSTITUT

Software topology

Three possibilities for control loops




```
adc0 = mscb_read(0, 0);
if (adc0 > 2.0)
   mscb_write(0, 1, 1);
else
   mscb_write(0, 1, 0);
```

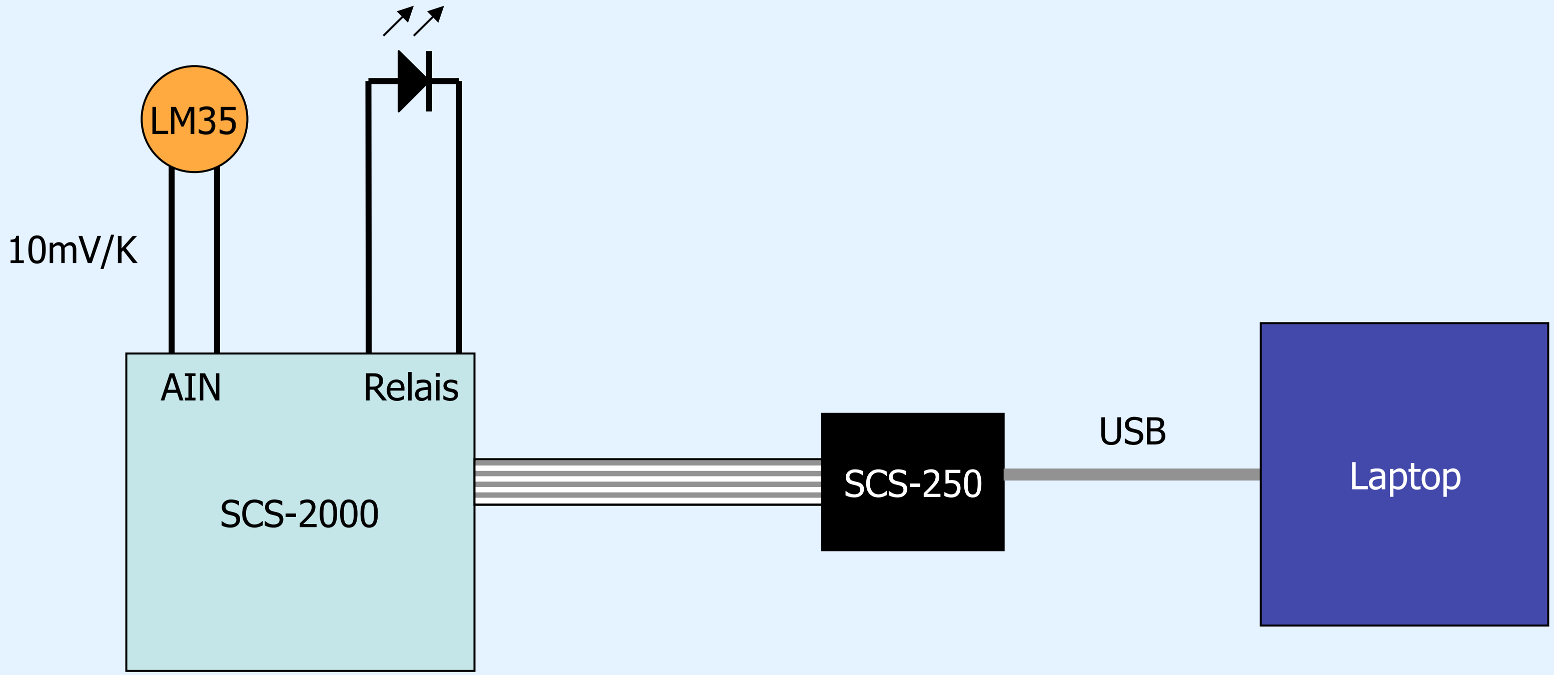



```
user_data.adc0 = adc_read(0);
if (user_data.adc0 > 2.0)
   user_data.valve0 = 1;
else
   user_data.valve0 = 0;

user_write(0);
```

PC (e.g. Midas Front-End) **PC (LabView)** **MSCB node**

Labview Demo

Simple demo to read a temperature and to control a relais



PAUL SCHERRER INSTITUT

Comparison FE+LV+uC

Which topology should I use?

	PC (Front-End)	Labview	Microcontroller
pros	Can be easily integrated into existing Midas Front-End Well known debugging environment Access via MIDAS slow control system	Easy to learn "Automatic" documentation **Quick getting started**	**No PC required** Very stable Access via MIDAS slow control system
cons	Needs PC-MSCB connection	Instability Not suited from complex tasks Costs Needs PC-MSCB connection **No (easy) remote access**	Limited resources on uC Requires uC development environment ($$)

LabView experience

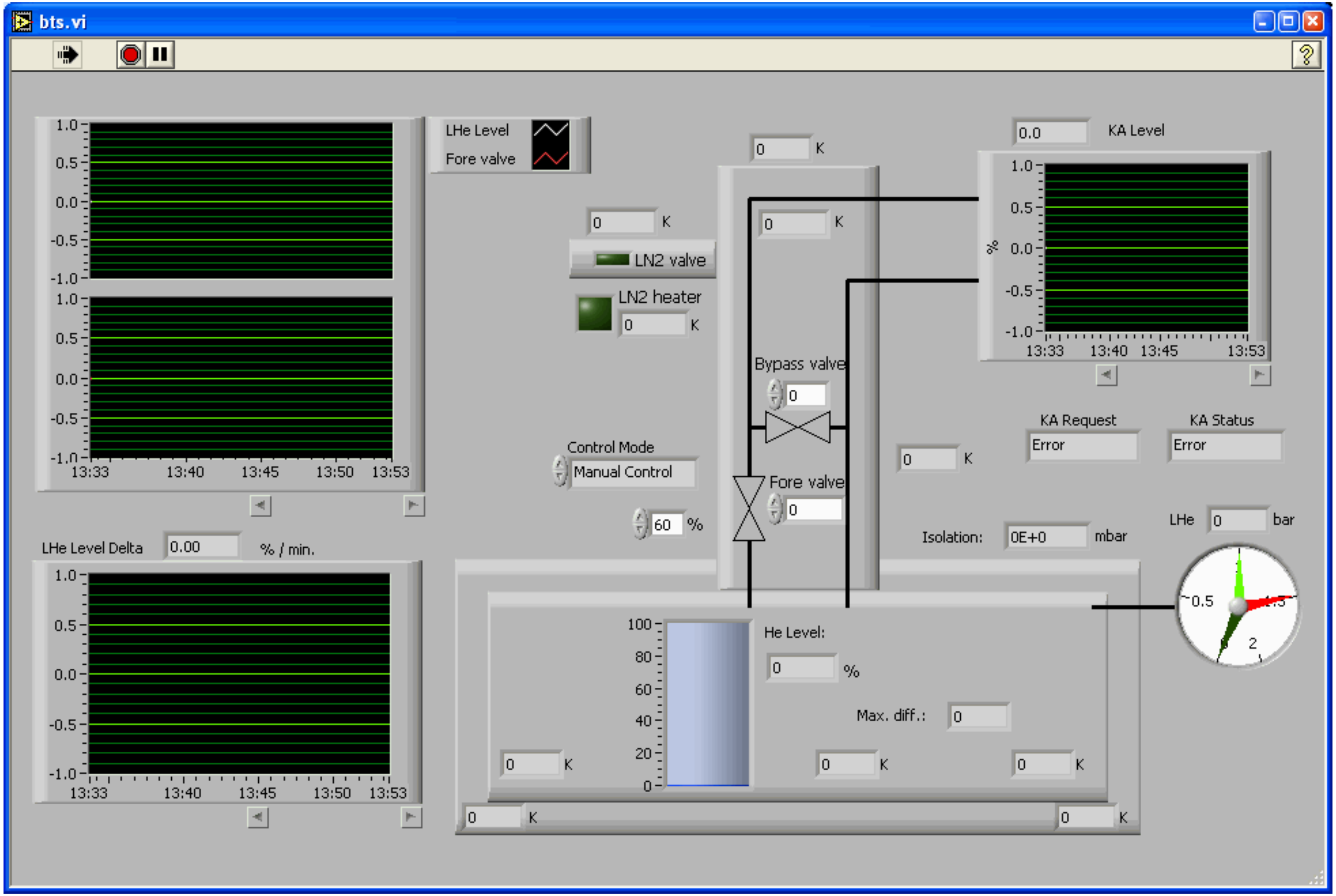
LabView very well suited to get quickly started and to develop control algorithms and visualization

For complex applications, LabView becomes cumbersome

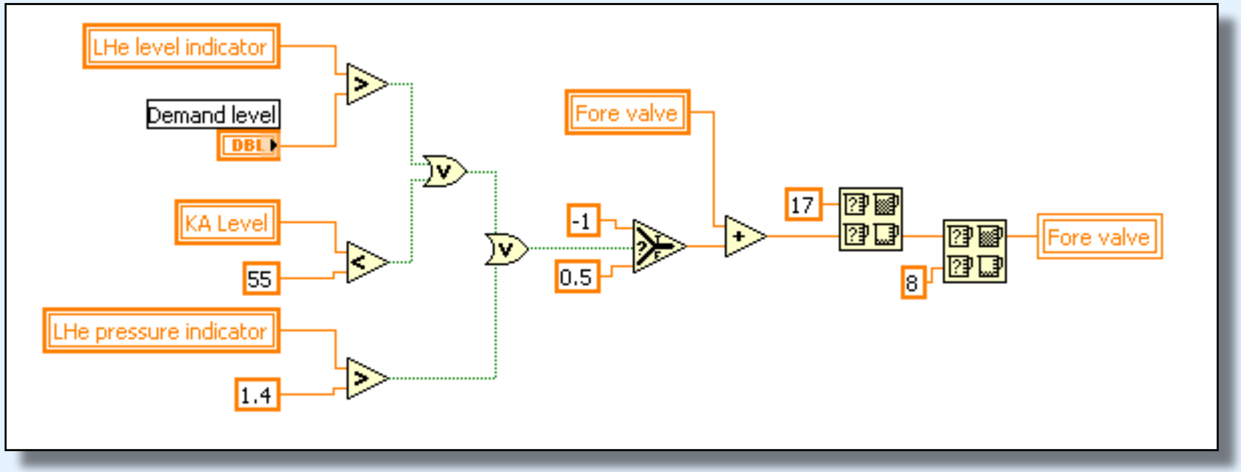
The “golden” road

- Development under LabView
- Implementation in μC
- Visualization in MIDAS slow control system

Transition LabView → uC/Midas

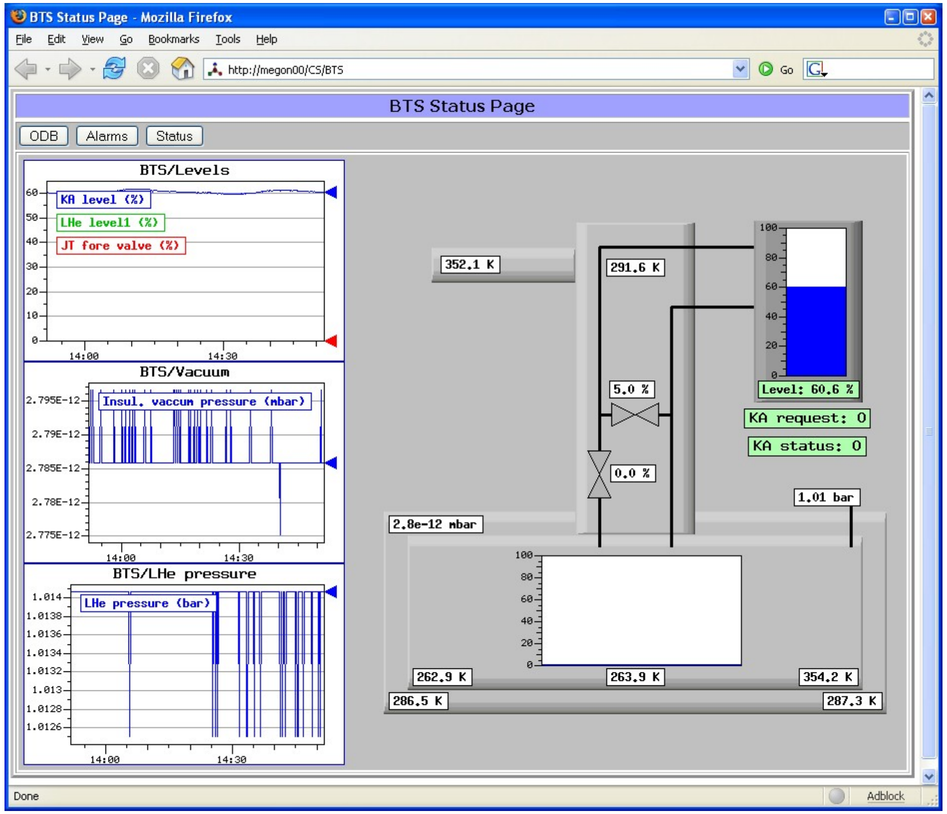


BTS Control

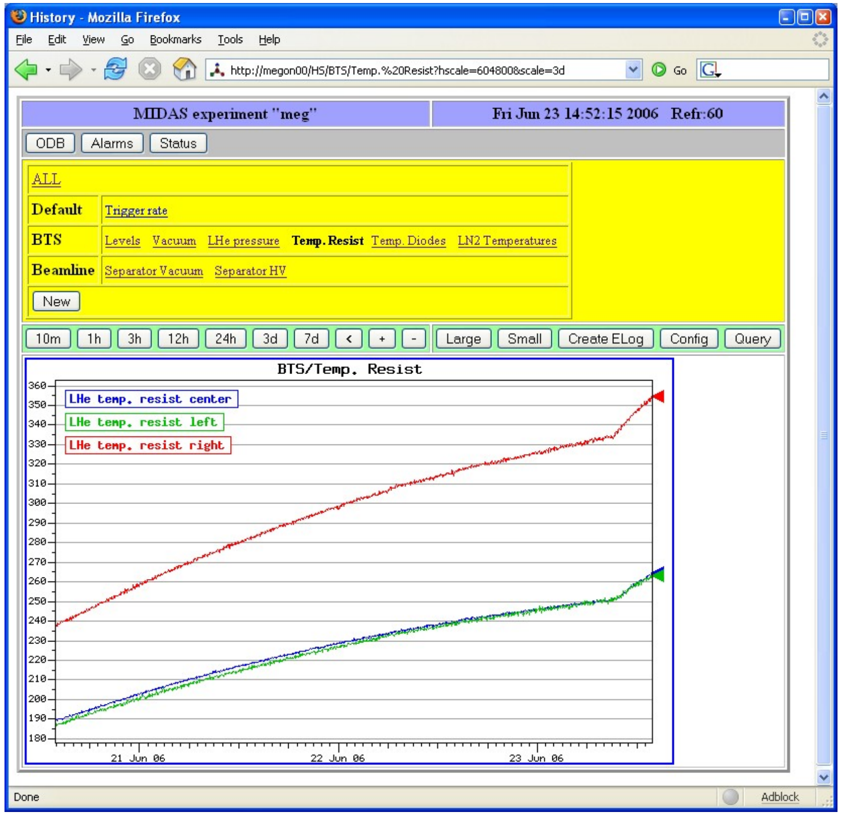


```
v = user_data.jt_forerun_valve;
if (user_data.lhe_level1 > user_data.lhe_demand ||
    user_data.ka_level < 55 || user_data.lhe_bar > 1.4)
  v = v - 1;
else
  v = v + 0.5;

if (v > 17)
  v = 17;
if (v < 8)
  v = 8;

user_data.jt_forerun_valve = v;
user_write(14);
```

MIDAS Custom Pages & History



MIDAS “custom” page



MIDAS histoy

Conclusions

Would I do it again?

- Money-wise: 5 years development (2MY), saved 200k CAD in HV and 100k CAD for experiment → **NO**
- Flexibility: Now takes a couple of days to develop new I/O card → **YES**

Would I choose 8051 μC again or 32-bit processor (ARM7)?

- 8 bit power @ 100 MHz enough for regulations, fast control (20 ns port access time)
- 256 Bytes RAM not enough, need at least 1kB + 32kB flash
- On-chip ADC/DAC not enough for high precision applications
- Development environment very good (In-circuit debugging, flash download via JTAG) → **YES**

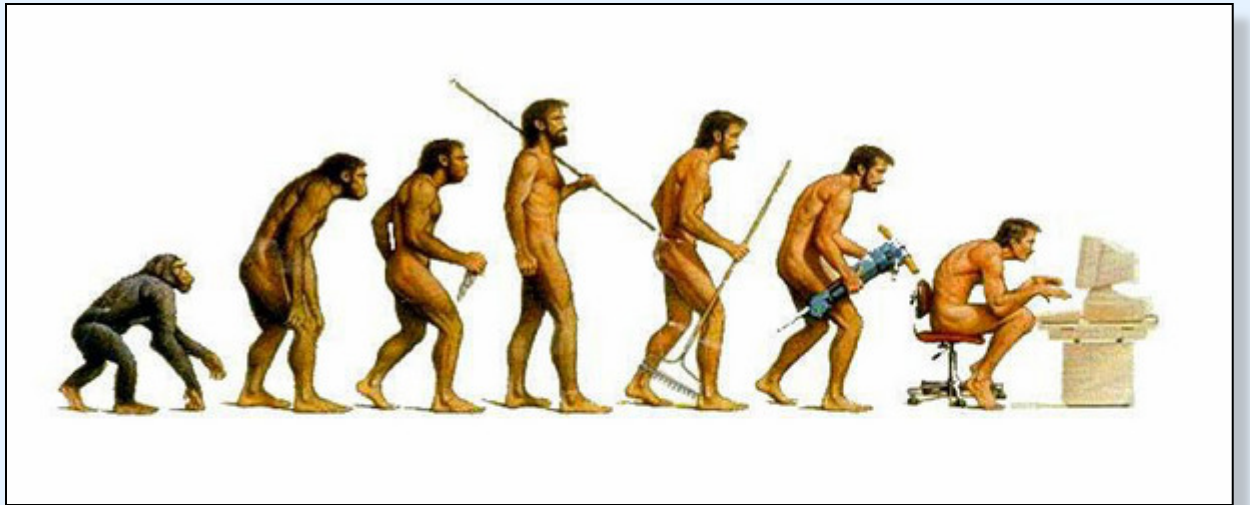
Conclusions II

Choose again RS-485 over CAN?

- MSCB protocol is very simple and optimized (like firmware upgradeable over network) → can debug with oscilloscope
- MSCB protocol can be extended
- Run currently at 115kBaud (good for 500m w/o termination)
- Very nice opto-decoupled RS-485 transceiver (ADM2486)
- C8051F121 @ 100 MHz should go to 2 MBit
- Drawback: RS-485 is single master, while CAN has MAC layer

Where do we stand now?

After all hardware runs nicely, we have to monitor it!



[redacted]

ODB hot-link update