SEU Tolerance in the ELMB

Henk Boterenbrood
software engineer
NIKHEF

Outline

- What is the ELMB ? (plus brief history)
- Some applications using the ELMB
- Radiation tests on the ELMB
- SEUs in the ELMB
- SEUs and the ELMB software
- Conclusion

ELMB: Embedded Local Monitor Board

- Credit-card sized plug-on board
  - microcontroller (8-bit, 4MHz, 128 kByte flash)
  - communication: CAN interface, 125 kbit/s
  - I/O capabilities
    - digital I/O
    - analog inputs: 64-channel 16-bit ADC (optional), max ca. 30 samples/s, calibrated in 6 voltage ranges
  - comes standard with software to operate digital-in/out and analog-in via CAN bus and *CANopen* protocol
  - relatively easy to customize software with low-cost tools and existing source code
  - in-system-programmable, remotely via CAN bus

- General-purpose standard building block with CAN-bus interface for various control and monitoring tasks in the LHC experiments (initially just for ATLAS)
  - qualified for the radiation levels expected in the LHC experimental caverns

- Designed and produced by ATLAS Detector Control System group (H. Burckhart)
  - hardware design by Björn Hallgren

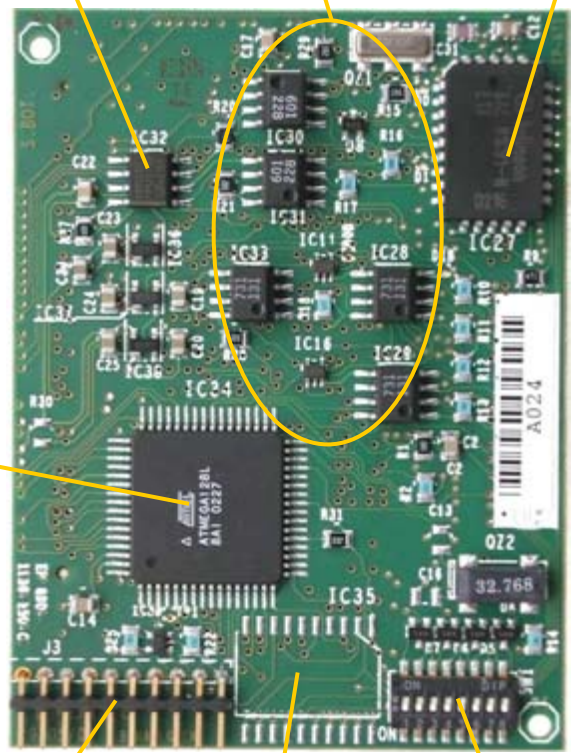
Why develop the ELMB ?

- Common solution for relatively simple control/monitoring tasks
  - Reduce design effort (hardware, software) by individual institutes/subdetectors
  - Simplify spares and maintenance issues (15 years)
  - Interfacing custom designs in a 'standard' way to the (ATLAS) Detector Control System (DCS)
    - hardware *and* software (*CANopen* protocol on the CAN-bus)
- No commercial solution to meet all requirements:
  - low power
  - low cost
  - high I/O density (possibility to connect many channels to one module, in particular analog-in)
  - *In-System-Programmable* (i.e. remotely *in-situ*, via CAN-bus)
  - for use in the LHC experiments
    - not sensitive to magnetic field
    - radiation tolerance, qualified to a certain level
      - be able to change component if not sufficiently rad-tolerant (rad-hard components out of the question because of cost)

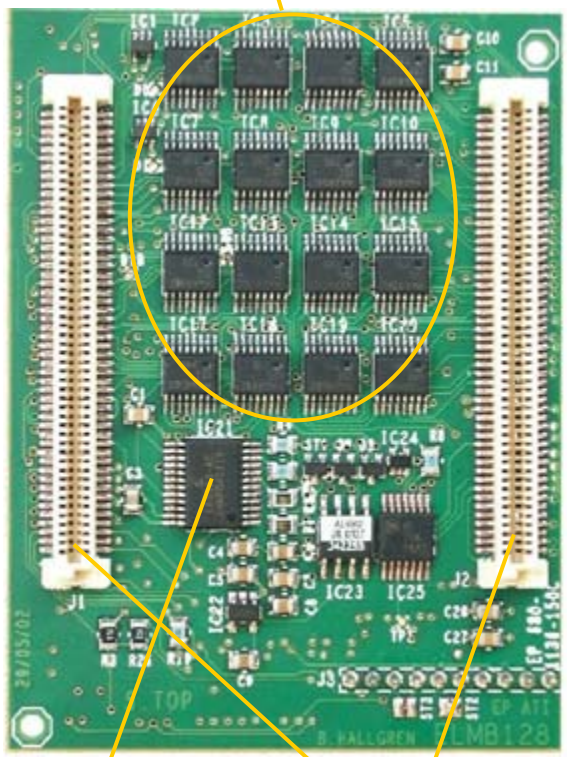
ELMB: brief history

- After a few prototypes…
  - CERN — LMB (with 2 micros with small memory) ca. 40 produced
  - CERN + NIKHEF — ELMB103 (with ATmega103 microcontroller + other) ca. 300 produced, in 2001
- Final design: ***ELMB128*** (with ATmega128 microcontroller: Bootloader section)
  - *ELMB128A*: with analog part
  - *ELMB128D*: without analog part
- Pre-series of 650 ELMB produced, end of 2002
  - to satify initial (ATLAS) subdetector needs
- Production of >10000 units, in 2004
  - ATLAS, >5000
  - LHC Rack & Gas Systems, ca. 2000
  - Other LHC experiments, ca. 1400?

ELMB: the board

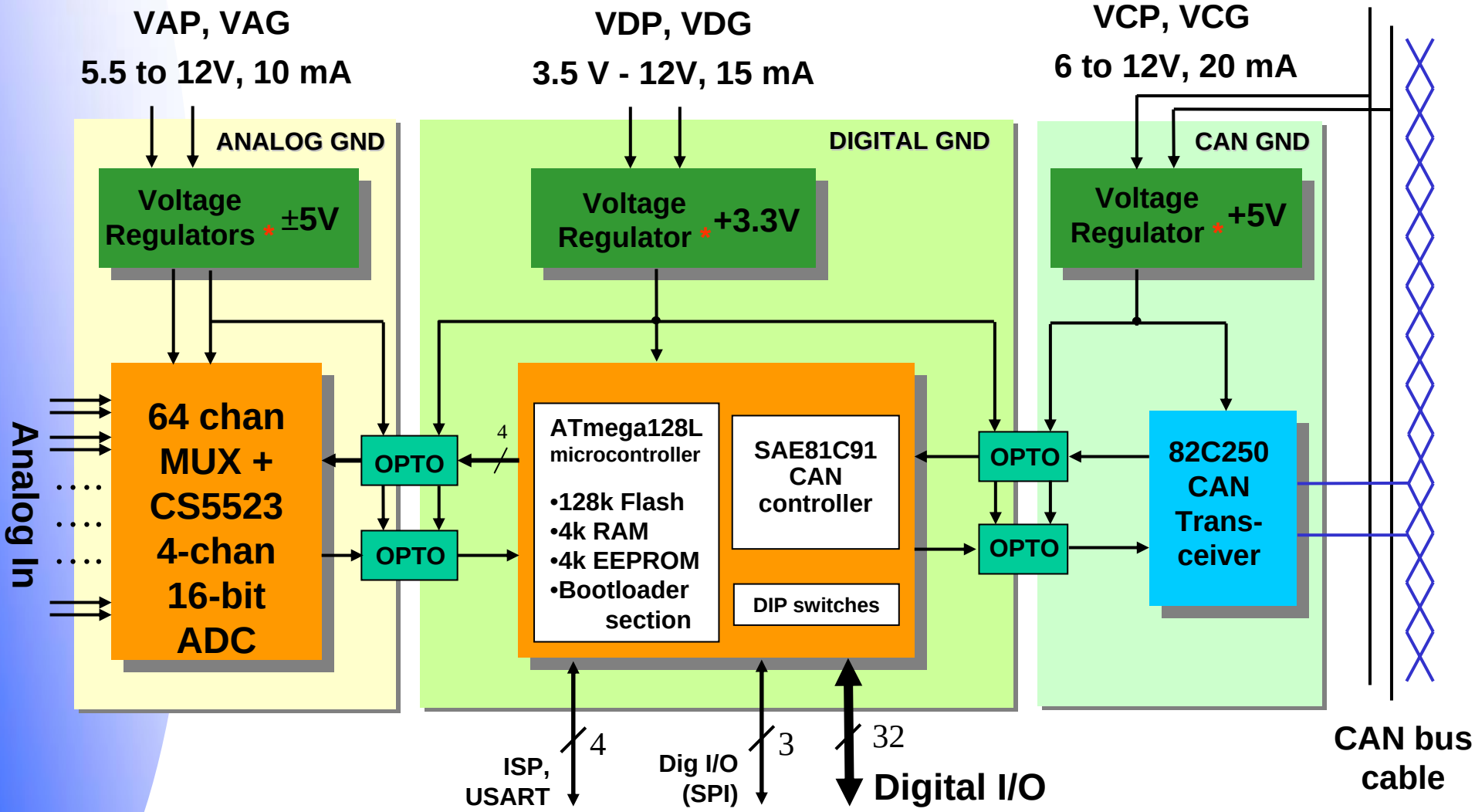


Size:
50x67 mm^2



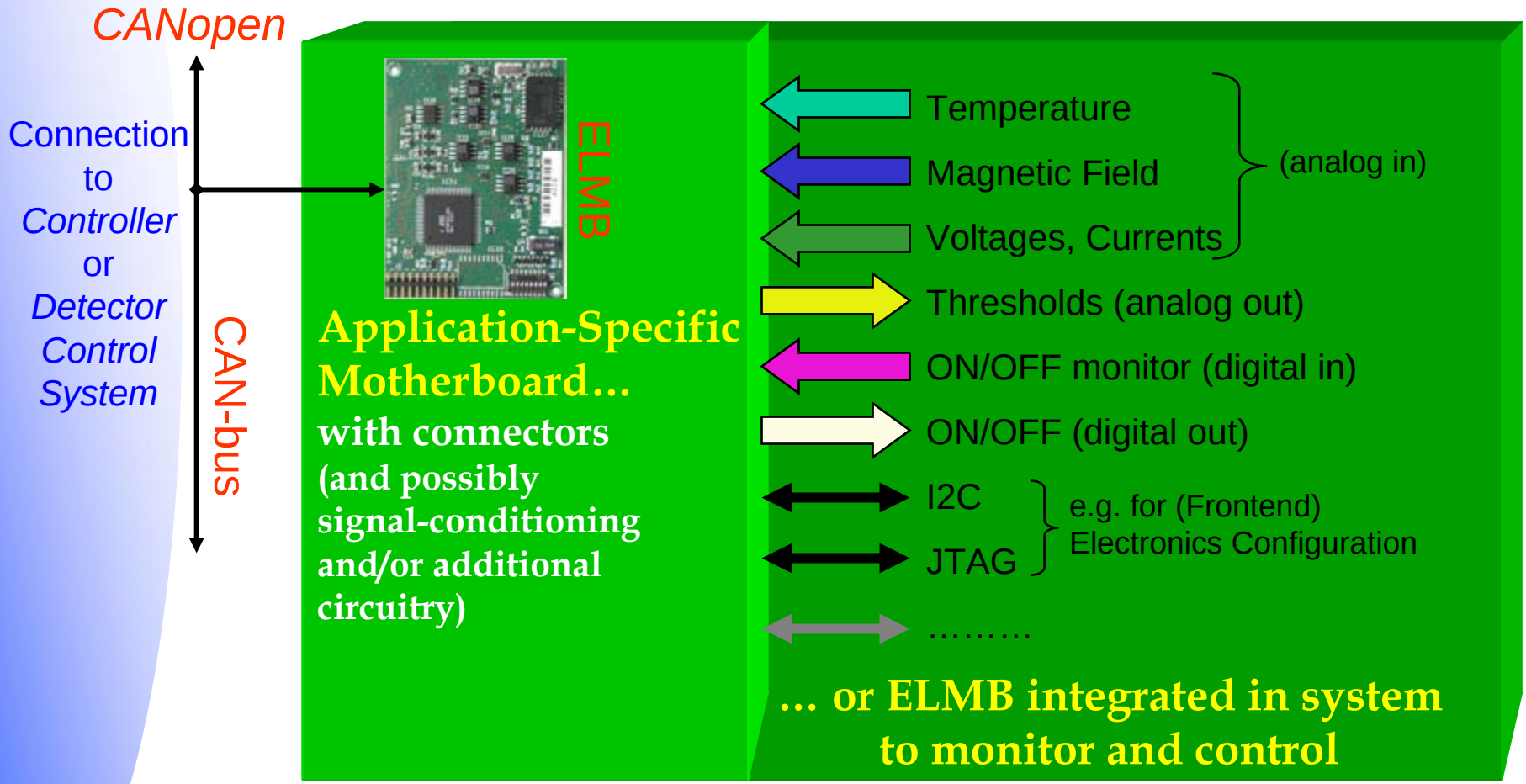
Version
without ADC:
bottom side empty
and on frontside
2 instead
of 5 opto-ICs

ELMB: block diagram

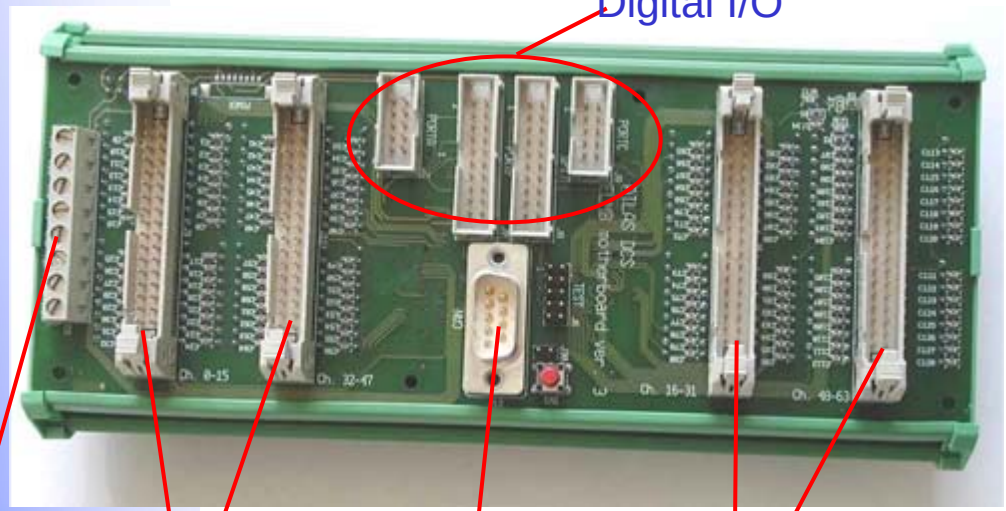


*regulators with thermal & current limits, protection against Single-Event-Latch-up (SEL)

ELMB: application

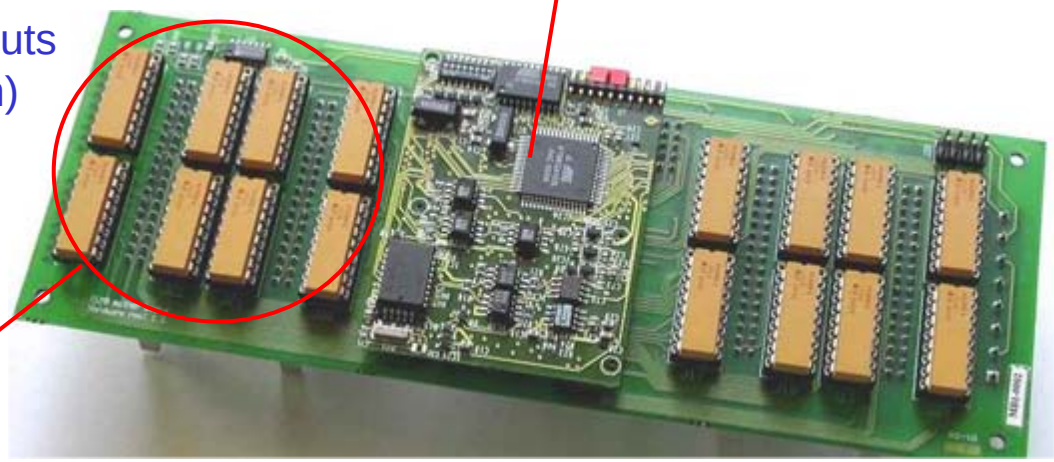


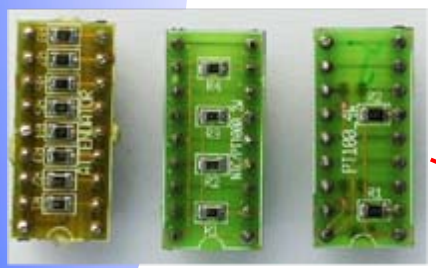
ELMB: general-purpose Motherboard



(ca. 300 produced)

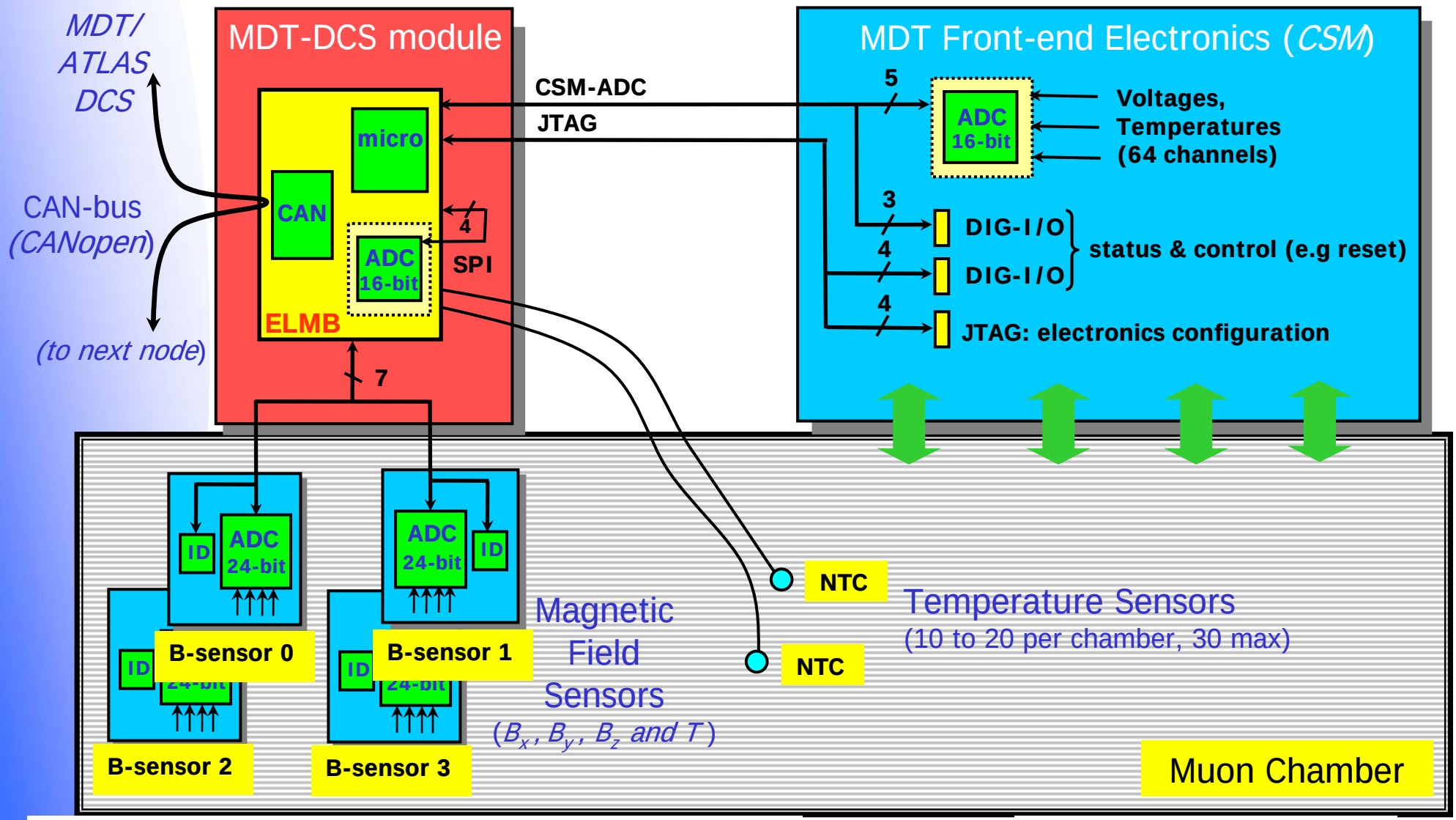
ELMB with 'standard'
CANopen application firmware
and Bootloader (*off production*)





analog input
signal adapters
(available for
PT100, NTC and voltage
measurements)

ELMB custom app + custom motherboard: ATLAS MDT Muon chambers (in rad env)



(ca. 600 chambers with one to four B-sensor modules each)

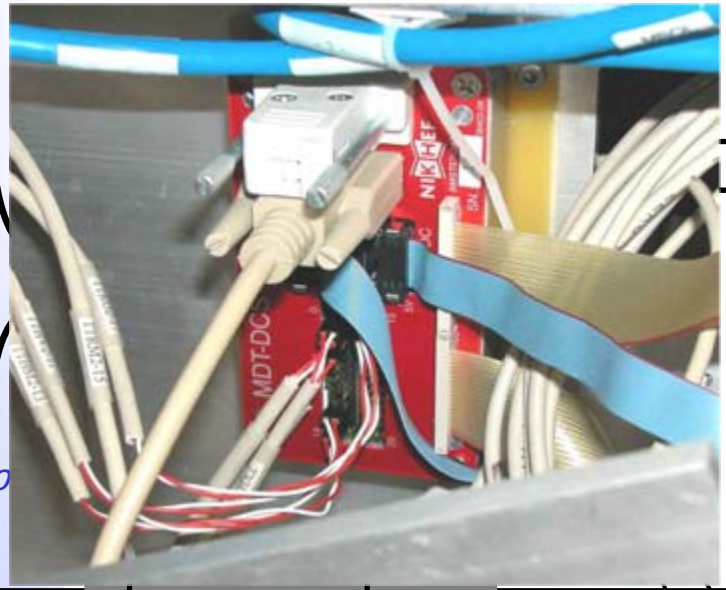
(1150 chambers in total)

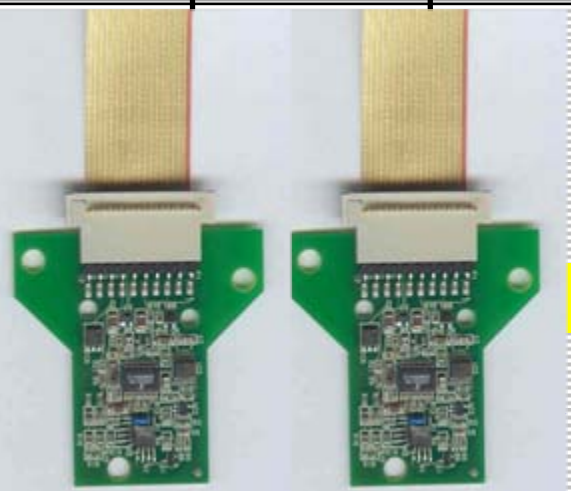
ELMB custom app + custom motherboard: ATLAS MDT Muon chambers (in rad env)

MDT/ ATLAS DCS

CAN-bus *(CANopen)*

(to next no





Magnetic Field Sensors
, B_y, B_z and T)

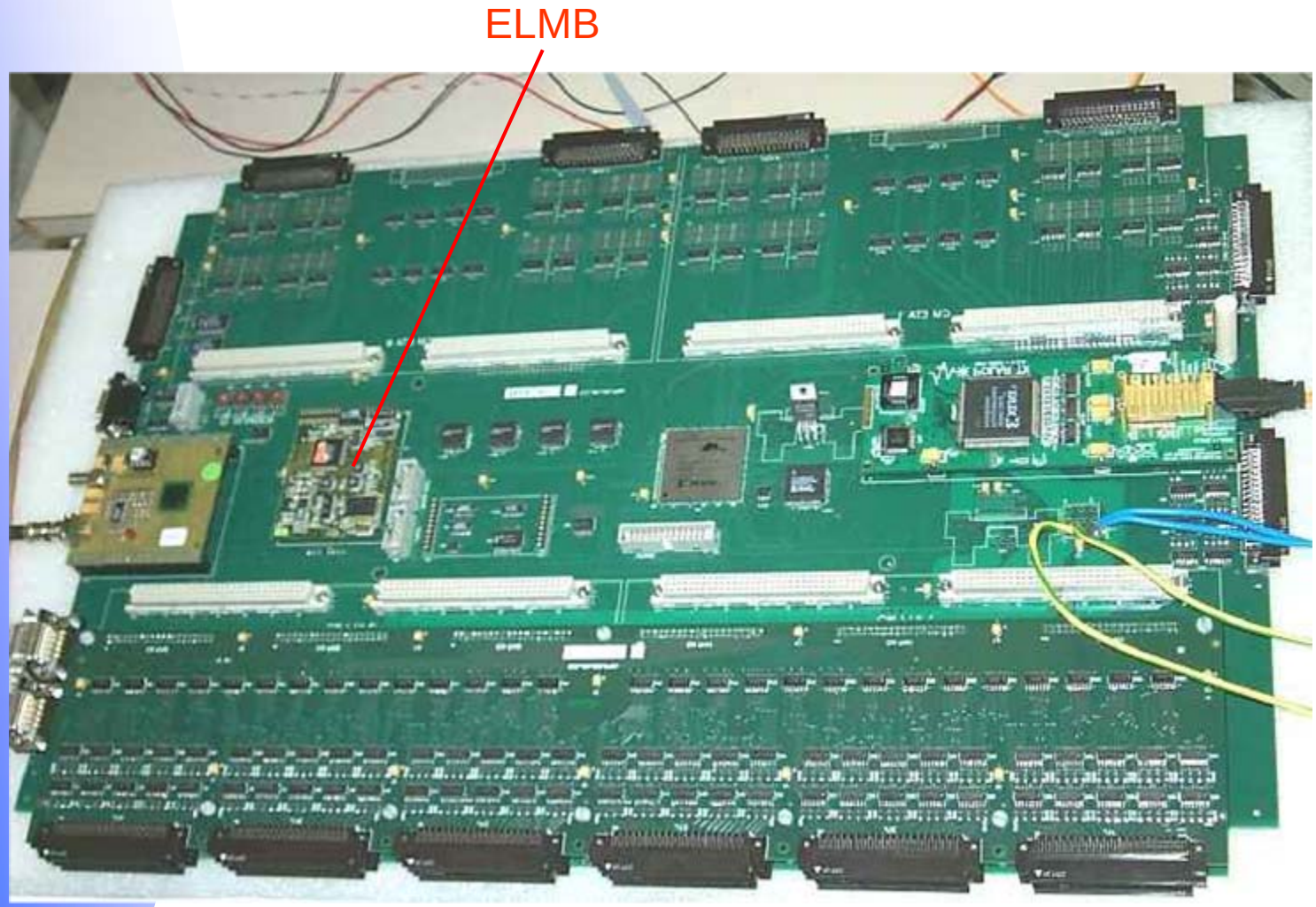
e Sensors
hamber, 30 max)

Muon Chamber

(ca. 600 chambers with one to four B-sensor modules each)

(1150 chambers in total)

ELMB custom app, integrated in system: ATLAS RPC Muon chambers (in rad env)



PAD board with TTCrx, ELMB, XCV200 and Optical Link

ELMB controls/ configs all of this:

- Temperature sensors
- TTC
- Delay chips
- FPGA
- Flash prom FPGA
- Flash prom SPI
- I^2C I/O registers
- Coincidence matrix ASIC (about 200 I^2C registers)
- Optical link controls

using JTAG and I^2C protocols and Dig I/O

(courtesy of S.Veneziano)

More applications using ELMBs...

- ATLAS
  - Muon TGC front-end electronics configuration & monitoring
  - Silicon Tracker (SCT) Low- & High-Voltage system controller
  - Liquid Argon Calorimeter (LAr) temperature monitor
  - Tile Calorimeter Low-Voltage system controller
  - and more…
- Electronics rack control (all LHC experiments)
- Gas flow meter read-out (all LHC experiments)
- …

ELMB Radiation Tests (1)

- Test on ELMB components, such as optocouplers (from 1998-)
- Series of tests on ELMB (2001 - 2004):
  TID (protons, gamma), NIEL (neutrons), SEE (protons)
  - prototype
  - final version
  - production series
- Documented/reported in ATLAS *Internal Working Notes* (IWN)
  http://atlas.web.cern.ch/Atlas/GROUPS/DAQTRIG/DCS/iwn.html
  - "Irradiation Measurements of the ELMB", 9 Mar 2001, *IWN9*
  - "Radiation test at GIF and accelerated aging of the ELMB", 2 May 2001, *IWN10*
  - "Radiation test of the 3.3V version ELMB at GIF", 31 Aug 2001, *IWN11*
  - "Single Event Effect Test of the ELMB", 20 Sep 2001, *IWN12*
  - "Non Ionising Energy Loss Test of the ELMB", 22 Jan 2002, *IWN14*
  - "TID radiation test at GIF of the ELMB with the ATmega128L processor", 8 Apr 2002, *IWN15*
  - "Results of radiation tests of the ELMB (ATmega128L) at the CERN TCC2 area", 26 Sep 2002, *IWN16*
  - "Results from Neutron Irradiations of the ELMB128", 29 Sep 2003, *IWN19*
  - "SEE and TID Tests of the ELMB with the ATmega128 Processor", 29 Sep 2003, *IWN20*
  - "NIEL Qualification of the ELMB128 Series Production", 18 Feb 2004, *IWN21*
  - "SEE and TID Qualification of the ELMB128 Series Production", 15 Nov 2004, *IWN23*

ELMB Radiation Tests (2)

- Requirements: radiation values guideline
  (calculated for ATLAS Muon Barrel)
  - TID: **4.7 Gy** * 3.5 * 1 * 2 = **33 Gy** = 3.3 kRad in 10 years
  - NIEL: **3.0E10 n/cm^2** * 5 * 1 * 2 = **3.0*10^{11} n/cm^2** (1 MeV eq.) in 10 years
  - SEE: **5.4E09 h/cm^2** * 5 * 1 * 2 = **5.4*10^{10} h/cm^2** (>20 MeV) in 10 years

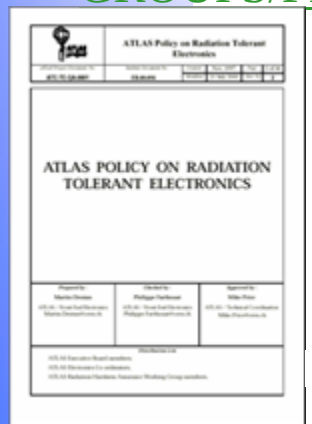
Safety factors:

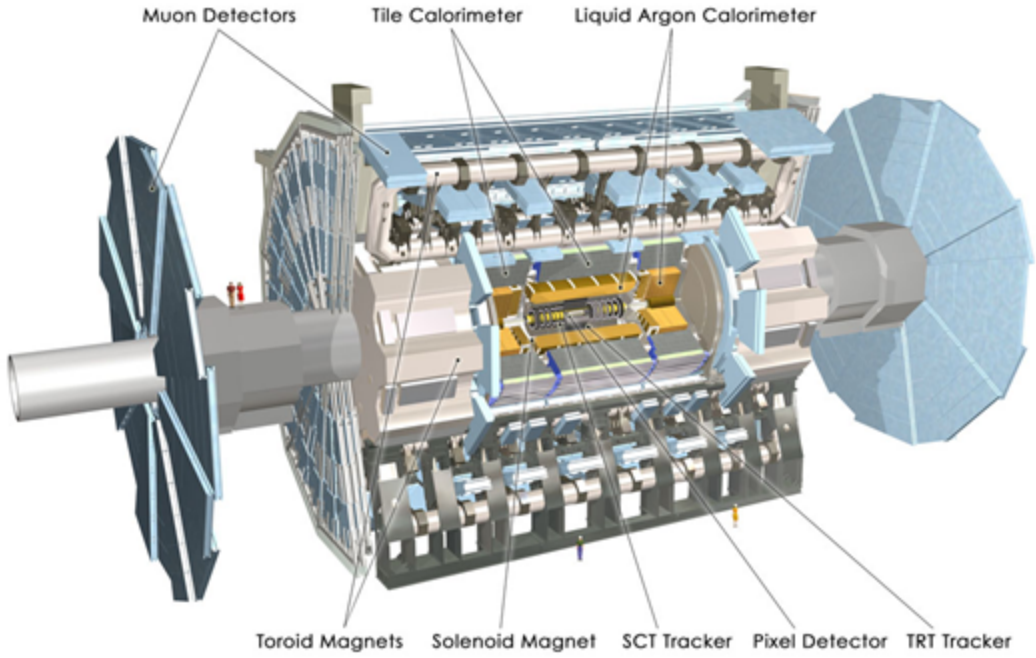
simulated radiation levels

Low Dose Rate Effect

COTS components mixed: factor 4,
COTS components homogeneous preselected: factor 2
COTS components homogeneous qualified: factor 1

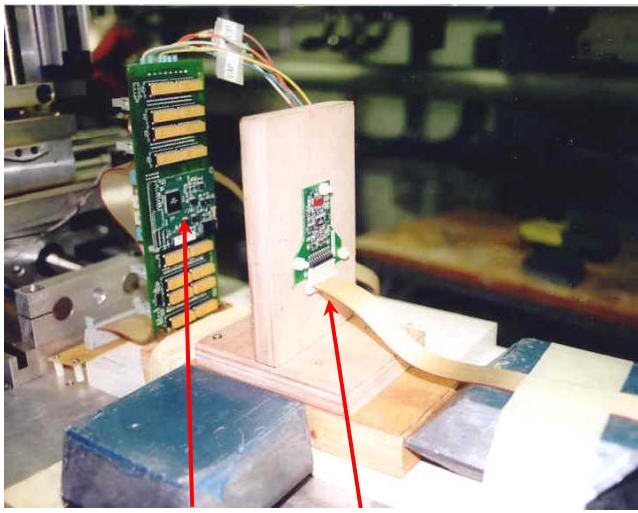
- based on document
  "ATLAS Policy on Radiation Tolerant Electronics"
  http://atlas.web.cern.ch/Atlas/GROUPS/FRONTEND/radhard.htm


ATLAS POLICY ON RADIATION TOLERANT ELECTRONICS



ELMB Rad Test: SEU (and TID)

- *CYCLONE* cyclotron, Louvain-la-Neuve (B), 60 MeV protons
  - June 2001: ELMB prototype
  - Apr 2003 : ELMB
  - Nov 2003: ELMB, production series (with components from homogeneous batches)
  - ca. 12 ELMBs, each irradiated with **$1.0*10^{11}$ p/cm^2** (Apr/Nov 2003 tests) corresponding to a TID of **140 Gy** (ELMB starts to degrade…)



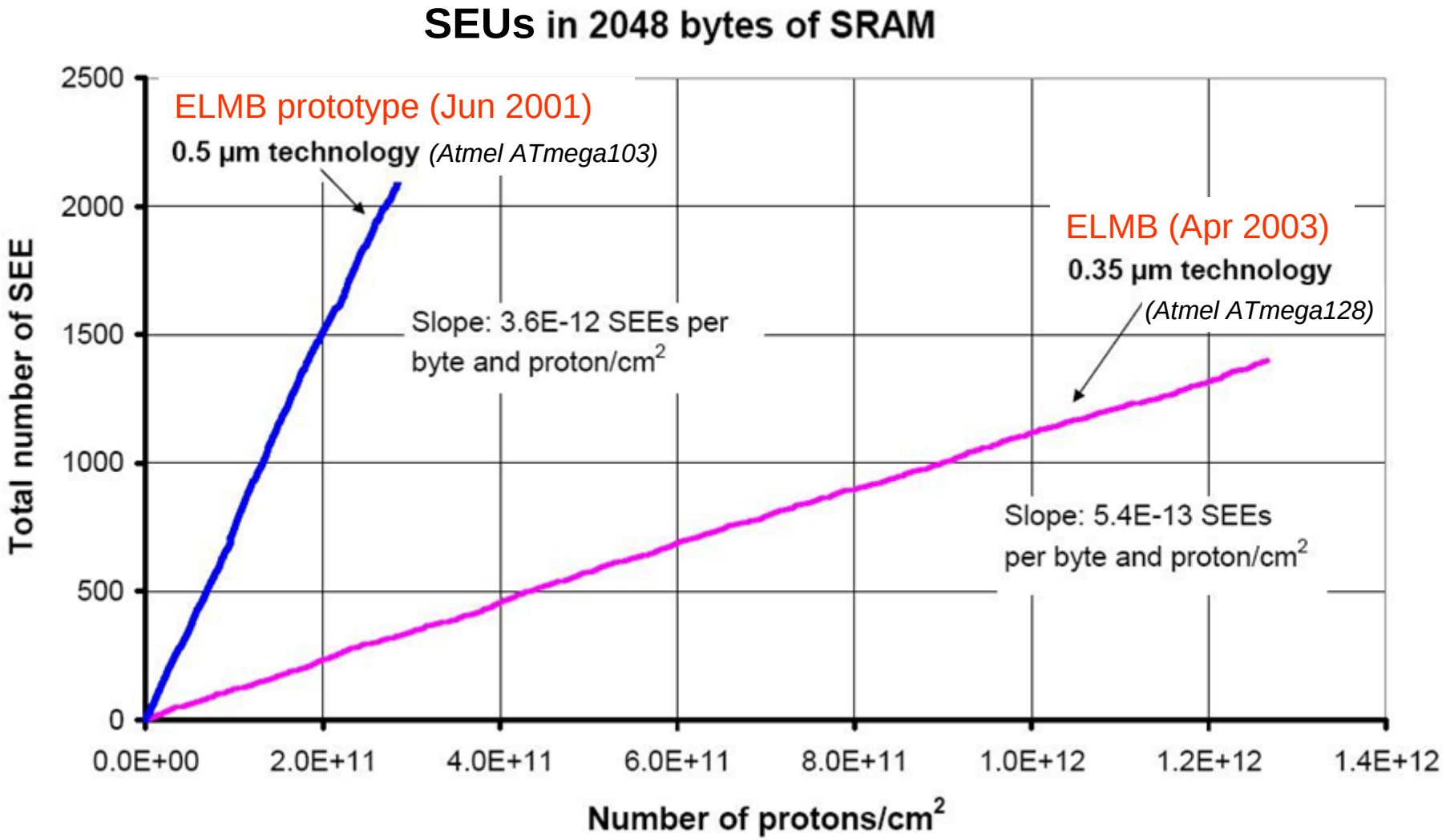
combined test of ELMB and B-field sensor

- ELMBs powered while irradiated, running
  - 'standard' ELMB software: read-out of ADC (constant input voltages) and digital I/O, CAN-bus message handling, sending message to and receiving from host PC
  - additional periodic (every 5 s) check of unused parts of memories and unused device registers, written with predefined bit pattern or device registers with known content

ELMB and SEU test

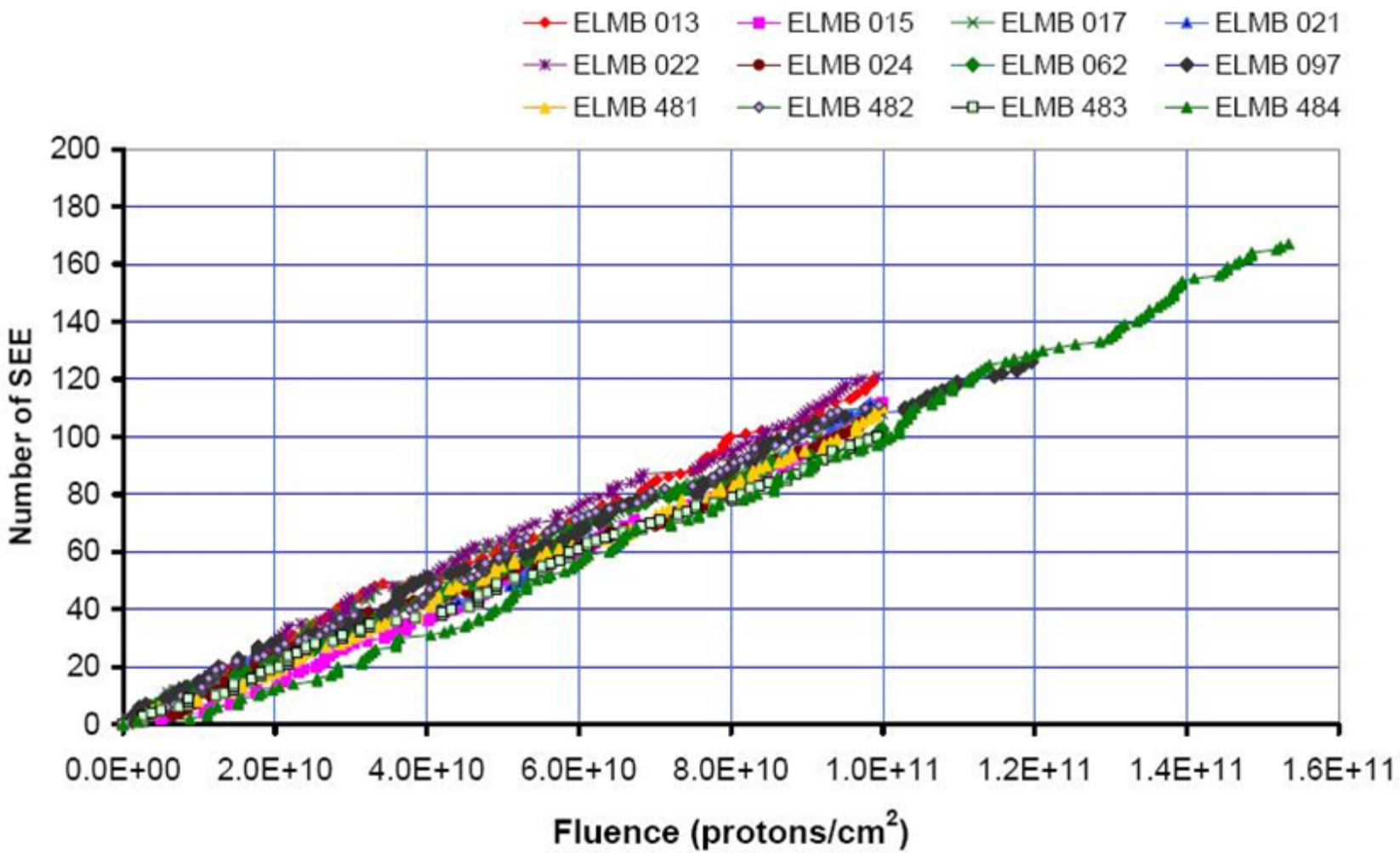
- No destructive SEE (latch-up, gate rupture or burnout) seen in the final ELMB design
  (ELMB proto: 1 latch-up seen during test, could be fixed by powercycle)
- Checking for
  - systematic errors: find bit inversions in the predefined bit patterns
  - functional errors: anomalies in behaviour
- Systematic errors: checking for bit flips in
  - Microcontroller onchip SRAM: 2 kBytes
  - Microcontroller onchip EEPROM: 2 kBytes
  - Microcontroller onchip FLASH: 56 kBytes
  - Microcontroller registers: 10 bytes
  - CAN-controller registers: 40 bytes
  - ADC device registers: 33 bytes

ELMB SEU systematic test SRAM (1)

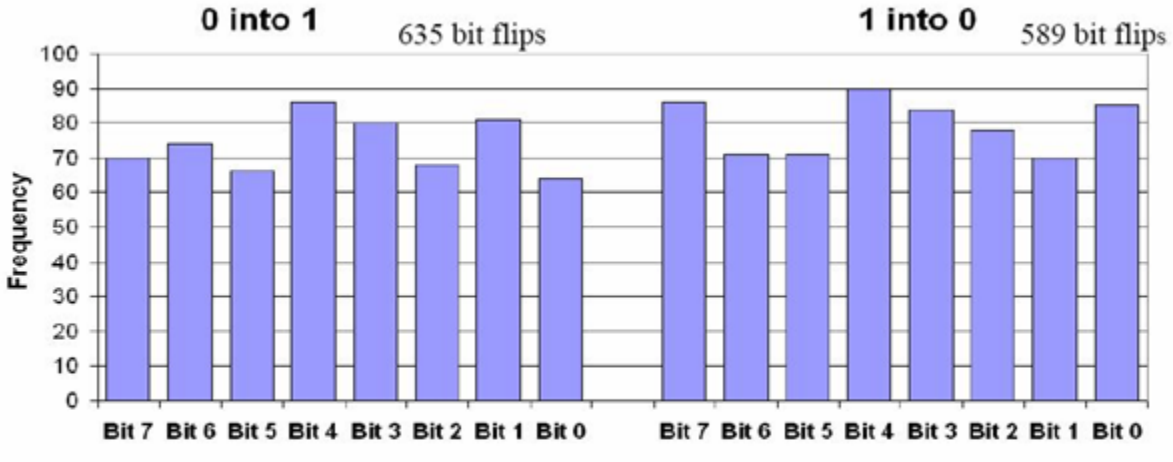


ELMB SEU systematic test SRAM (2)

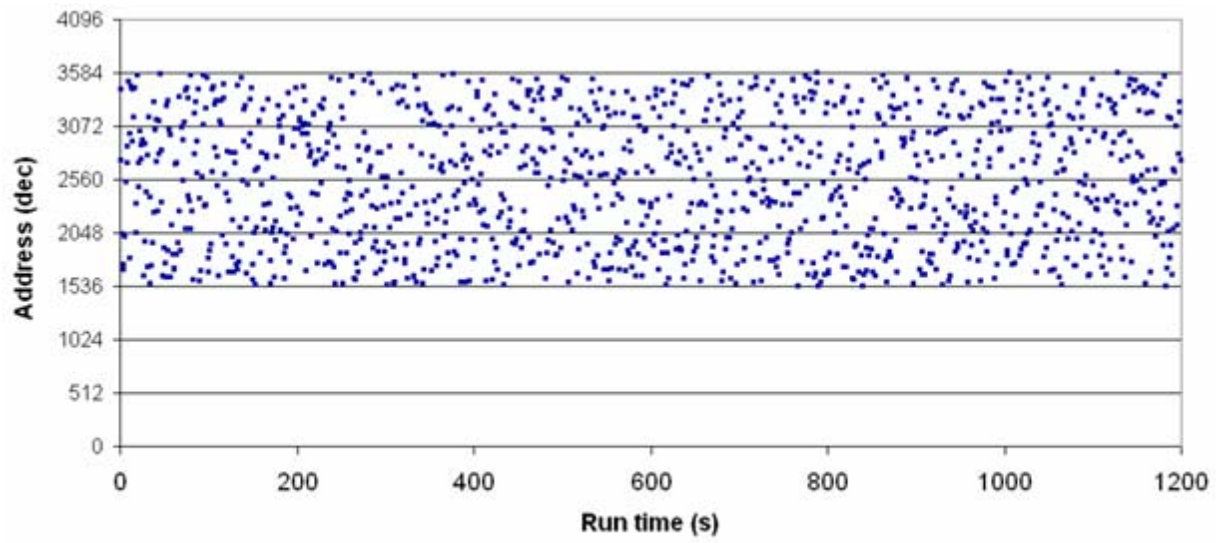
SEUs in 2 kByte of SRAM, per individual ELMB (Apr 2003 test)



ELMB SEU systematic test SRAM (3)



Number of changes from 0 to 1 and 1 to 0 in the 8 bits of the SRAM bytes The number of SEEs was 1224 single bit flips evenly distributed over bits and polarity.



Addresses of the SRAM where the SEEs were located versus run-time

Summary ELMB systematic SEUs (1)

- SRAM and device registers sensitive to SEU
- Not a single error found in EEPROM or FLASH
- Comparison of number of bit flips counted

  (scaled to a total fluence of **$1.1*10^{12}$ p/cm^2**) :

0.50 μm technology

	SRAM	EEPROM	FLASH	CAN	ADC	μC regs
ELMB proto	7733	0	0	61	73	---
ELMB	1233	0	0	27	2	0
ELMB (prod)	1122	0	0	54	5	1

microcontroller
in 0.35 μm technology

change of technology(?)

Summary ELMB systematic SEUs (2)

- Comparison of number of bit flips per byte
  (scaled to a total fluence of **$1.1*10^{12}$ p/cm^2**, between brackets number of bytes in test) :

	SRAM (2048)	EEPROM (2048)	FLASH (57344)	CAN (40)	ADC (33)	µC regs (10)
ELMB proto	3.78	0	0	1.53	2.21	---
ELMB	0.60	0	0	0.68	0.06	0
ELMB (prod)	0.55	0	0	1.35	0.15	0.10

- Fluence ca. 20 times more (0.5 vs $11*10^{11}$p/cm2) than required for ATLAS
  - assume using 1 Kbyte (SRAM+registers) in an application then leads to ca. 600/20 = 30 SEUs per ELMB (in 10 years)
  - ATLAS has ca. 3000 ELMBs in the cavern → 90000 SEUs in 10 years = ca. 25 SEUs/day

ELMB functional SEUs

- Detect and count 'anomalous' behaviour, and categorize according to the action taken to recover from it
  - automatically (e.g. by CAN-controller hardware, ADC time out)
  - soft reset (manually)
  - power cycle (manually)
- Anomalous behaviour (detected by observing), for example:
  - no replies or other messages: SEU in some CAN-controller register?
  - ADC gain change: SEU in ADC gain setting register ?
  - change in timer period between periodic action
  - Watchdog Timer reset
  - CAN bus errors (automatically handled by CAN-controller hardware)

Summary ELMB functional SEUs

- Number of *functional* errors counted:

	ELMB proto test (6/2001)	ELMB test (4/2003)	ELMB (prod) test (11/2003)
Power cycling	15	0	1
Software reset	19	1	1
'Automatic' recovery	78	13	3

(scaled to a total fluence of **$1.1*10^{12}$ p/cm^2**)

- So 2 resets required due to SEUs (11/2003 test), for a fluence ca. 20 times more (0.5 vs $11*10^{11}$p/cm2) than required for ATLAS, i.e. 0.1 resets per ELMB in 10 years
  - ATLAS has ca. 3000 ELMBs in the cavern → 300 resets in 10 years = less than 3 resets or powercycles per month
- Improvement from proto to production version due to both hardware (technology change) and software measures taken after initial rad test

ELMB software and SEU

- Knowing that
  - EEPROM (and Flash) is not sensitive to SEU
  - SRAM and device registers are sensitive to SEU
- Write the software such as to take this into account to try to minimize the impact of SEUs on the ELMB's operation
  - points listed in the next slides
    in more or less arbitrary order

ELMB software and SEU: various points (1)

- CAN communication (for ELMB main link to the outside)
  - low-level bus traffic error handling and message integrity fully taken care of by the CAN-controller hardware (very robust protocol)
  - I use an intermediate message buffer in SRAM with buffer management variables due to lack of buffering in controller:
    3x message counter and 3x 'next message' pointer with majority voting
  - periodic refresh of registers, where possible and without interfering or loss..
- CAN protocol (*CANopen*)
  - 'Life Guarding': if haven't received message from the 'host' for a while, reset and re-initialize the CAN-controller (in case configuration got corrupted)
  - host: read back items you wrote; read items you just read again; if possible..
- Settings/variables that are configurable but do not change often (i.e. are long-lived in the running program, are globals but not constants)
  - a working-copy is stored in EEPROM and is read right before each use, so use 'rad-hard' EEPROM as extension/replacement of SRAM
  - Example:
```
LifeTimeFactor = eeprom_read( EE_LIFETIMEFACTOR );
if( LifeTimeFactor > 0 && LifeGuardCntr >= LifeTimeFactor )
  { ……
```
  - Beware: EEPROM has limited number of write/erase cycles (100000) !

ELMB software and SEU: various points (2)

- Minimizing use of SRAM, don't use more bits than necessary for variables, using masks
  - Example, boolean-like variable needs only 1 bit:

```
unsigned char boolean;   /* In C no type 'bool' */
if( (boolean & 1) == 1 ) /* 7 bits don't care   */
   { /* true… */ }
```

- Microcontroller and ADC registers
  - Microcontroller's I/O registers ('direction', i.e. input or output) and ADC configuration settings refreshed from values in Flash or EEPROM
  - ADC regularly reinitialized/recalibrated (ELMB: optionally before every readout cycle of all inputs)
- Watchdog Timer
  - if you have one, use it (on ELMB: always on, not software controllable)
- Opto-couplers (slow type) in interface between micro and a device
  - the bit signal width is configurable (in this case between 10 and 255 µs) as performance and delays may change under influence of radiation (setting stored in EEPROM of course!)

ELMB software and SEU: various points (3)

- Some additional notes
  - avoid to take a (drastic) decision based on a single reading
  - try to avoid writing to EEPROM (i.e. change the configuration of the ELMB app) during radiation, since stuff in ELMB EEPROM is now considered error-free, reliable data
  - remote in-system-programming very useful because bugs and unanticipated 'effects' can be fixed
    - beware: flash programming capability deteriorates due to radiation, so a working module can be 'killed' trying to upgrade/reprogram it
    - ELMB power-up: *Bootloader* (= flash upgrade app) is active for a few seconds, allowing upgrade even when user application is corrupt
    - don't upgrade during radiation
  - now we wait for LHC to come online and see what happens…

Conclusion

- ELMB qualified for radiation environment up to certain levels –exceeding the requirements– by extensive and rigorous testing, including TID, NIEL and SEE
- ELMB sensitivity to SEUs considered more than acceptable (but not for safety-critical applications in the cavern…)
- Coding practices in the ELMB software contribute to more radiation-tolerant behaviour of the embedded application software with respect to SEUs
- References
  - ATLAS rad-hard electronics webpage http://atlas.web.cern.ch/Atlas/GROUPS/FRONTEND/radhard.htm
  - ELMB info: documentation, schematics, source code http://elmb.web.cern.ch
  - ELMB rad tests reports http://atlas.web.cern.ch/Atlas/GROUPS/DAQTRIG/DCS/iwn.html

Key Messages

- The ELMB is composed of standard COTS components, no special rad-tolerant or rad-hard components were used.
- A lot of effort and preparation goes into radiation tests on a hardware module plus its software, such as the ELMB, which -in fact- is actually a fairly simple module, and still only part of a larger system.
- Changing a component in the design or a change of technology by the component manufacturer may have a large impact on the SEU sensitivity of a module design, so radiation tests should be repeated (example: change of processor type on the ELMB).
- In the SEU tests of the ELMB we found that SEUs occur in SRAM and device registers. The software has been written with a number of adaptations to take this into account, such as a majority voting scheme, register refresh, and others.
- In the SEU tests of the ELMB we did not find any SEUs in EEPROM or FLASH memory. The software has been adapted to take advantage of this fact by using the EEPROM as a kind of rad-tolerant extension to the SRAM for storing long-lived variables.
- The extra precautions taken in writing the ELMB software have contributed to mitigate the effects of SEUs in the module and increase the overall tolerance of the ELMB to SEUs.