Course Module

Embedded Software is Difficult

Sebastian Fischmeister

sfischme@uwaterloo.ca

May 30, 2012

Department of Electrical and Computer Engineering
University of Waterloo

Outline

Lecture Thesis:

- Become aware of the *invisible* side: embedded computing.
- Get a glimpse of the challenges in embedded systems.
- In the worst case, just have a couple of good stories to tell.

Discuss

- What is bigger: PC market or ES market?
- What is an embedded system?
- Think of three embedded devices.

Outline

Embedded System "Definition"

A general-purpose definition of embedded systems is that they are devices used to control, monitor or assist the operation of equipment, machinery or plant. "Embedded" reflects the fact that they are an integral part of the system. In many cases, their "embeddedness" may be such that their presence is far from obvious to the casual observer. Even the more technically skilled might need to examine the operation of a piece of equipment for some time before being able to conclude that an embedded control system was involved in its functioning.

(Institute of Electrical Engineers)

Embedded System "Definition"

The Institute of Electrical & Electronics Engineers (IEEE) considers a computer system and its software embedded, if it is an integral component of a larger system and is used to control and/or directly monitor that system by using special hardware devices.

(paraphrased from IEEE P1003.13/D2.1)

Loosely defined, it is any device that includes a programmable computer but is not itself intended to be a general-purpose computer.

(W. Wolf. "Computers as Components")

Embedded System Characteristics

- Built for specific purpose
- Interaction with the environment and physical processes
- Stringent timing and resource requirements
- Mass-produced, everywhere around us

Properties of Embedded Systems

- Tightly-coupled to the physical world
- Rich in extra-functional requirements
- Heterogeneity, networked at extreme scale
- Sociological and ethical requirements
- Designed for debuggability.

Outline

Where Are They (1)

© MS Multimedia

© thewondrous.com

Where Are They (2)

© www.gitcameras.com

© w-cellphones.com

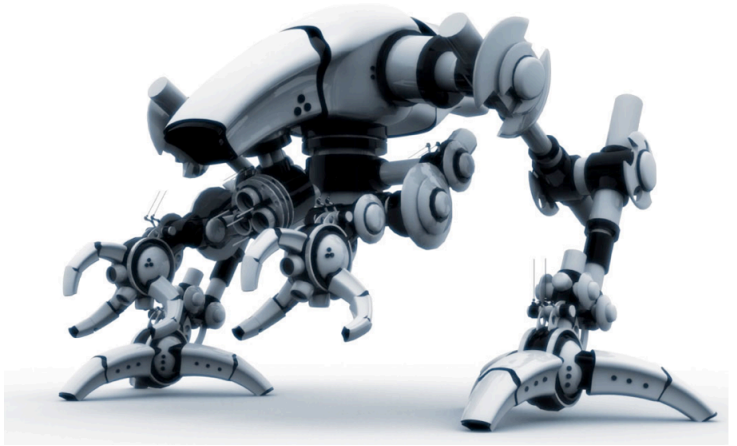
© www.tokyoezine.com

Where Are They (3)

© www.militarydesktop.com

© www.militaryphotos.net

© www.popsci.com



© www.popsci.com

Where Are They (4)

Consumer electronics

- mobile phones
- printer
- digital camera
- VCR, DVD player, HD-TV

Medical devices

- MRI
- infusion pump
- pacemaker
- artificial organs

Computing subsystems

- switches
- network cards
- serial bus chips

Mainframe computing (60's-70's)

- Relationship: $\frac{human}{computer} \approx \infty$
- Large computers executing big data processing applications

Personal computing & Internet (80's-90's)

- Relationship: $\frac{human}{computer} \approx 1$
- Personal computers executing business applications

Ubiquitous/physical computing (late 90's)

- Relationship: $\frac{human}{computer} \approx 0$
- Smart environment (=embedded systems) assists continually

Outline

Example: BMW 745i

- $\approx$ 70 processors in the car (53 8-bit, 7 16-bit, 11 32-bit)
- 2,000,000 lines of code
- Mix of operating systems
- Multiple networks (2 CAN high, 2 CAN low, serial, ...)
- Buggy?

Embedded Software Growth

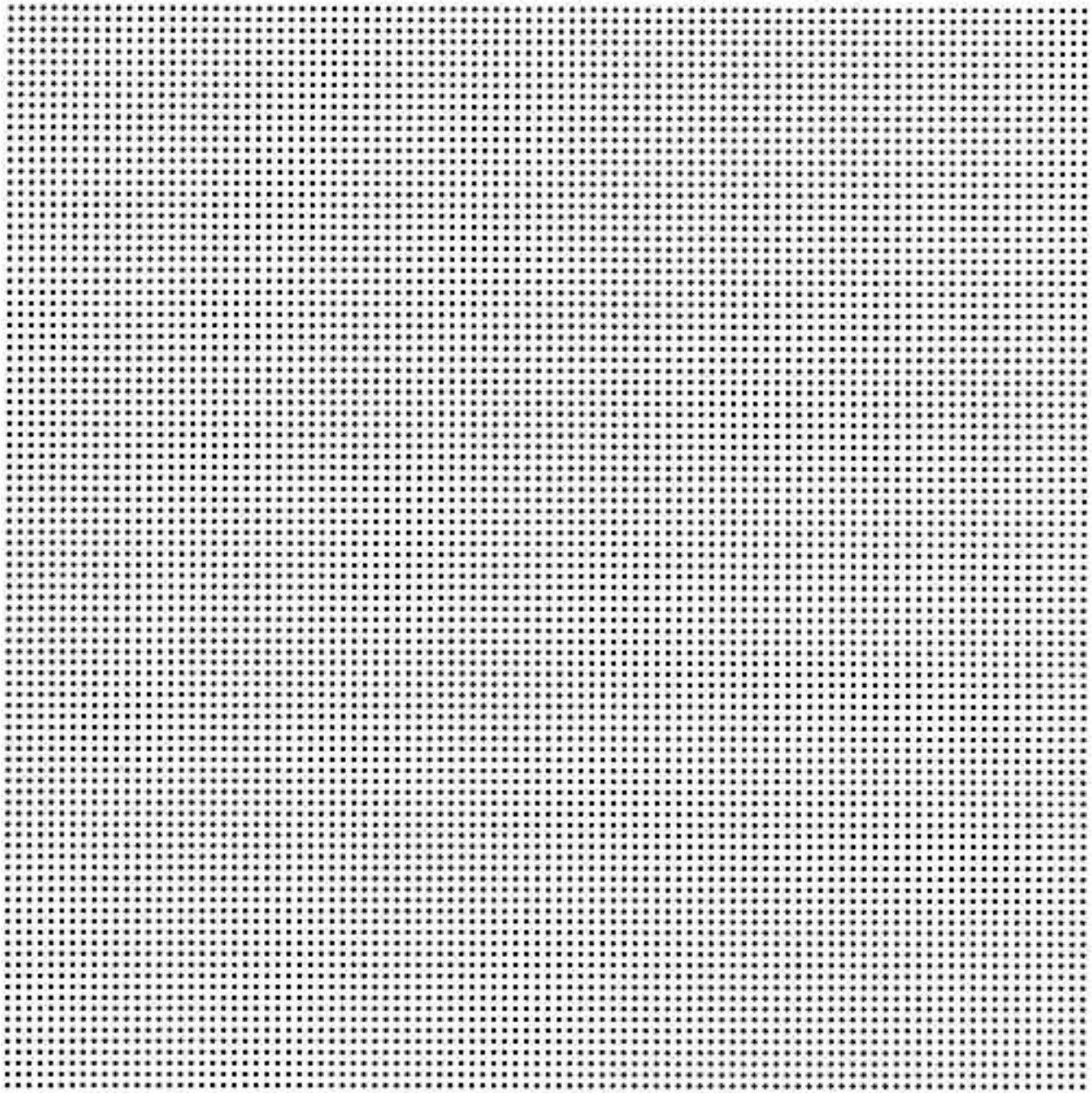
- F-22 Raptor (1997): 1.7M lines of code
- F-35 Joint Strike Fighter (2006): 5.7M lines of code
- Boeing 787 (200x): 6.5M lines of code

- Mariner (1962): 30 lines of code
- Voyager (1977): 3,000 lines of code
- Mars Exploration Rover (2003): 4M lines of code

- GM engine control (1982): 50,000 lines of code
- Current generation: 100M lines of code
- Next generation premium car: 300M lines of code

Can we Comprehend 300M LOC?

Spotting the bug?



Why does This Problem Exist?

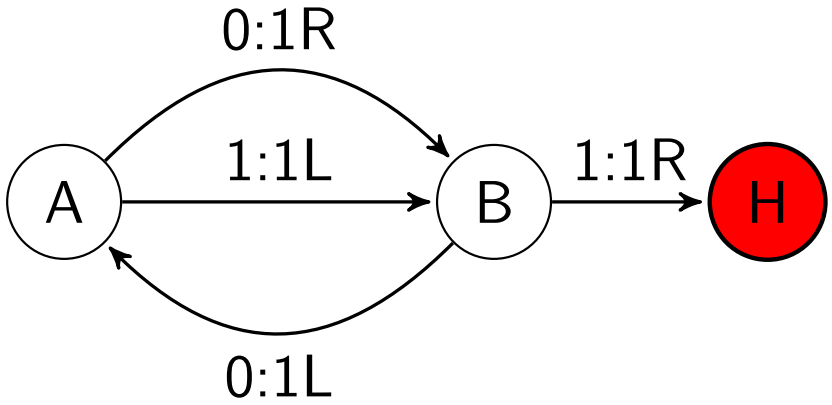
Software size and complexity is the challenge!

The bridge from Tokyo to Vancouver:

Busy Beaver (1)

$S(n, m)$: the largest number of steps taken by an n-state, m-symbol machine started on an initially blank tape before halting.

	A	**B**
0	1RB	1LA
1	1LB	1RH

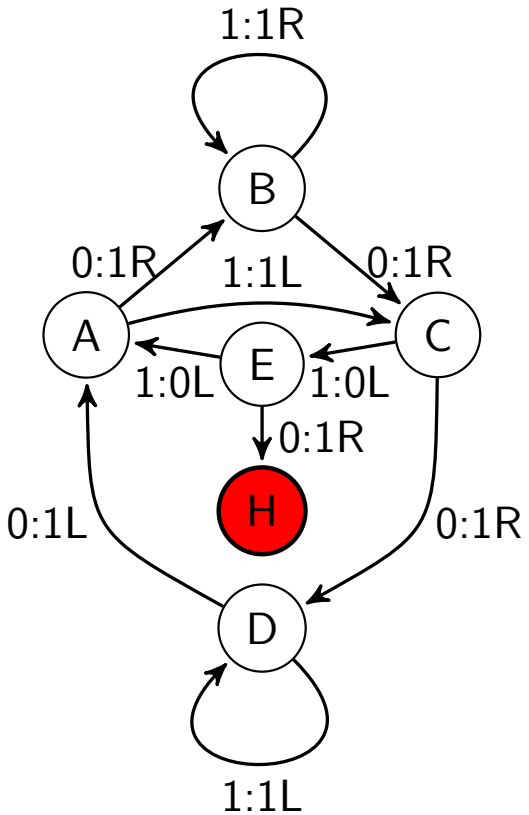


`http://bit.ly/IJF9dO`

0 0 1 1 **1** 1 0 0 (6 steps, four "1"s total)

Busy Beaver (2)

$S(5,2) =$



$\geq 47,176,870$

$S(2,5) = \geq 1.9 \times 10^{704}$

Properties of Embedded Software

Basic: correct, dependable, safe

- **Resource constrained:** 32kB of program memory, 1kB of RAM
- **Low power:** device should last for years not hours. Example: pacemaker
- **Production cost sensitive:** recurring production costs matter more than non-recurring engineering costs. Example: tools can be expensive

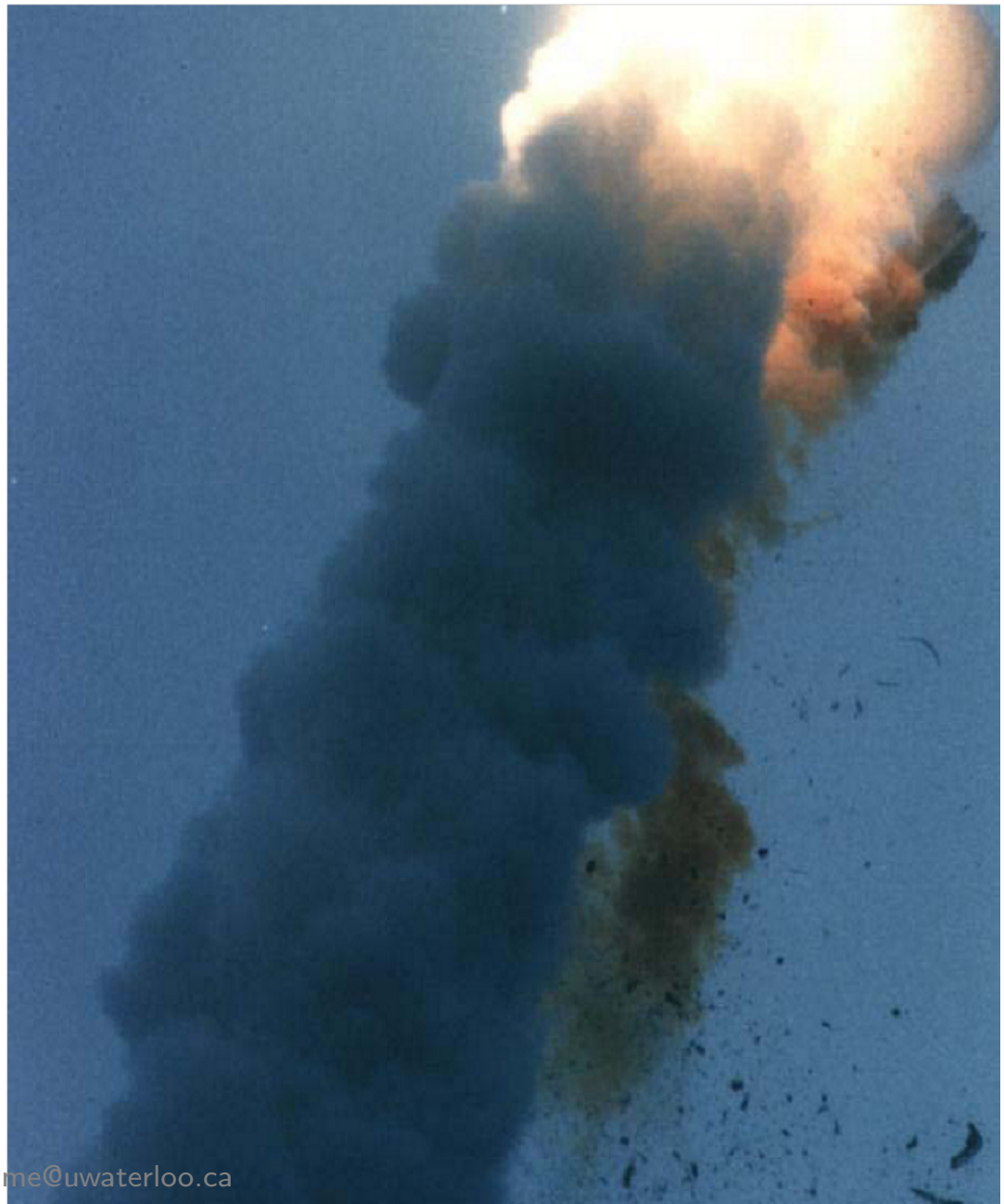
Outline

Do you think writing embedded software is easy?

Digital Thermometer

Nuclear Power Stations

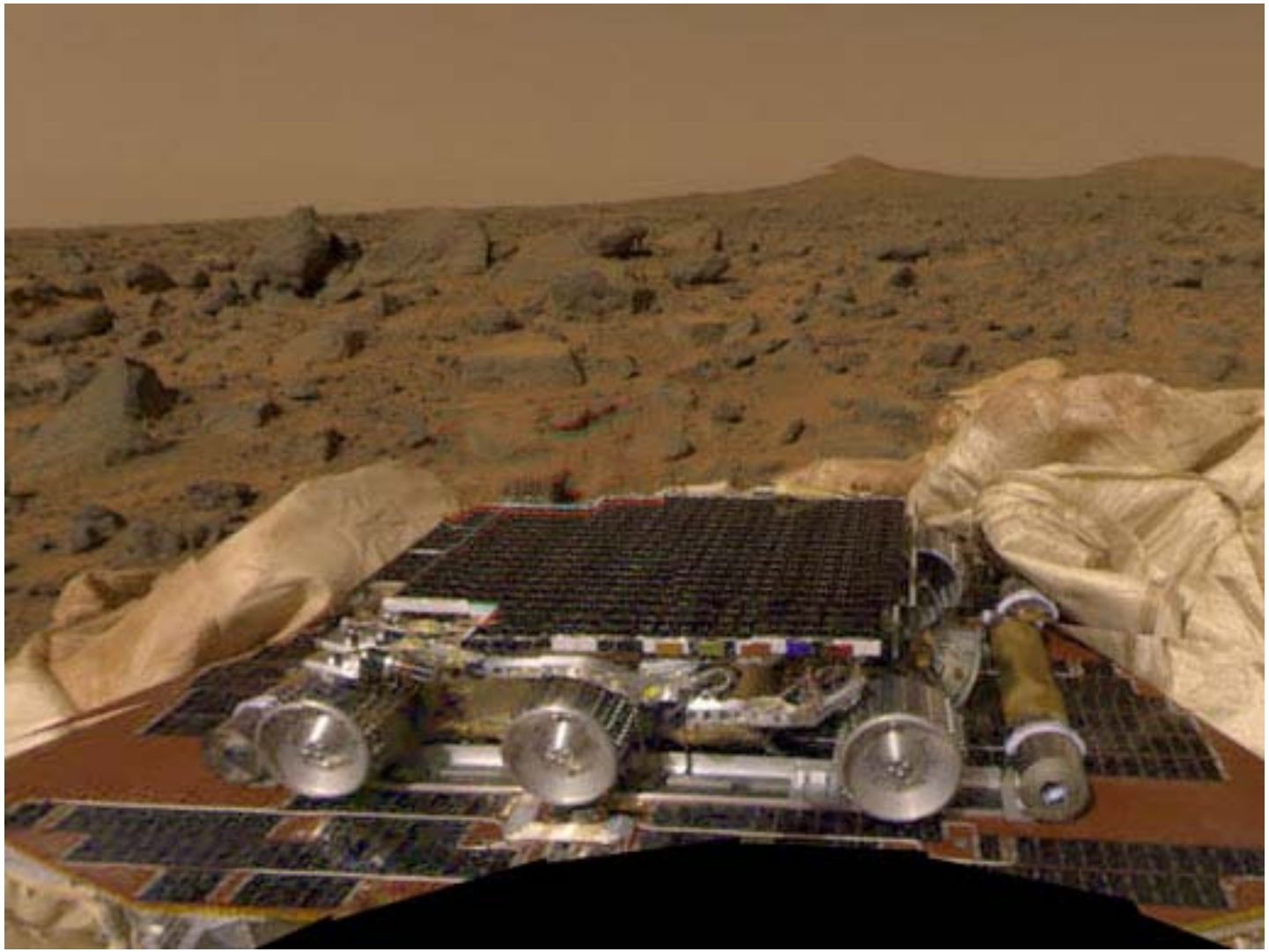
Ariane 5



Ariane 5 (1996)

- Self-destructing after 37 seconds.
- One of the most expensive computer bugs in history.
- Problem: 64-bit $\rightarrow$ 16-bit, disabled trap handler

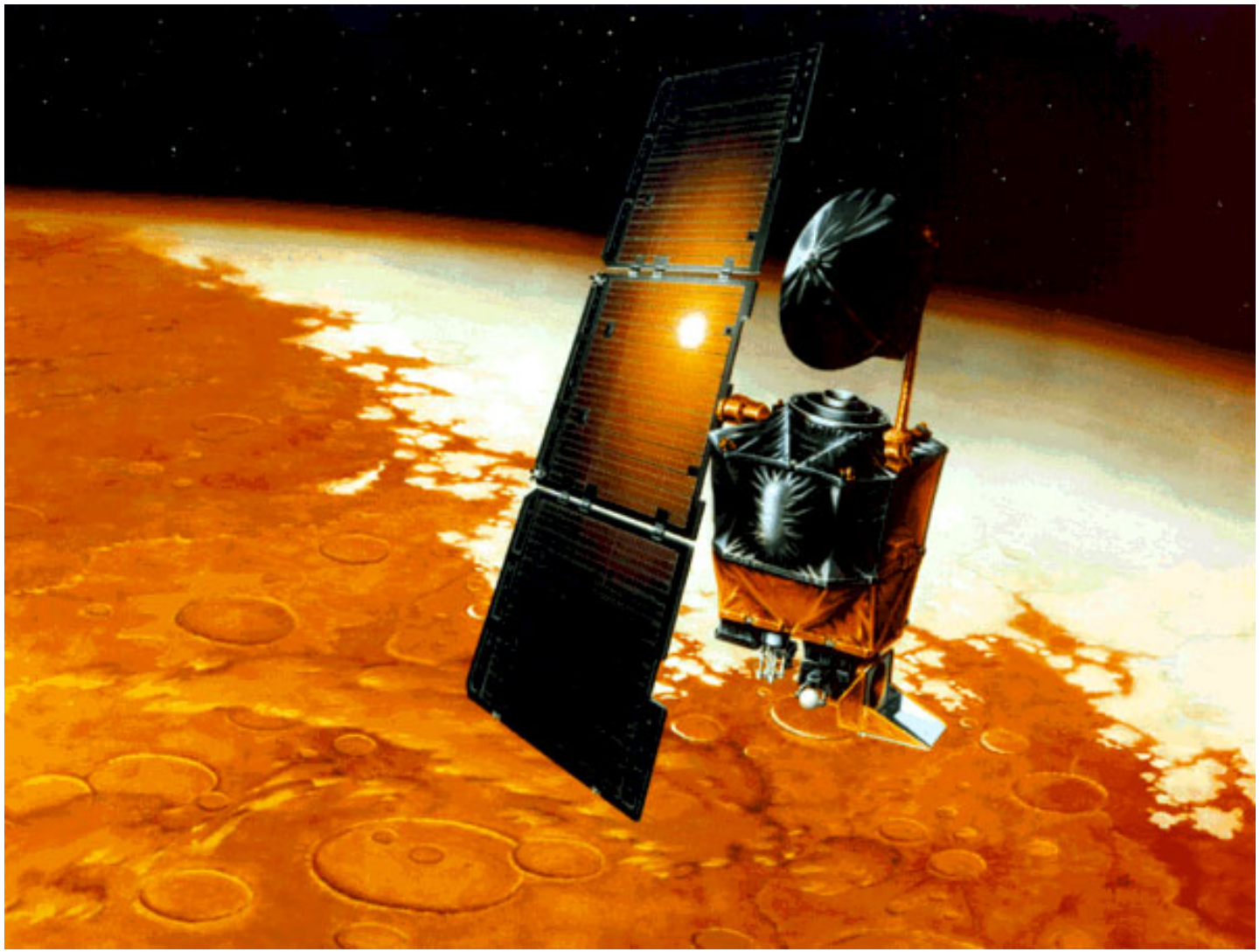
Mars Pathfinder



Mars Pathfinder (1997)

- Few days into the mission, Pathfinder started resetting.
- Problem: Priority inversion
- Fix: traces from ground model, online software update.

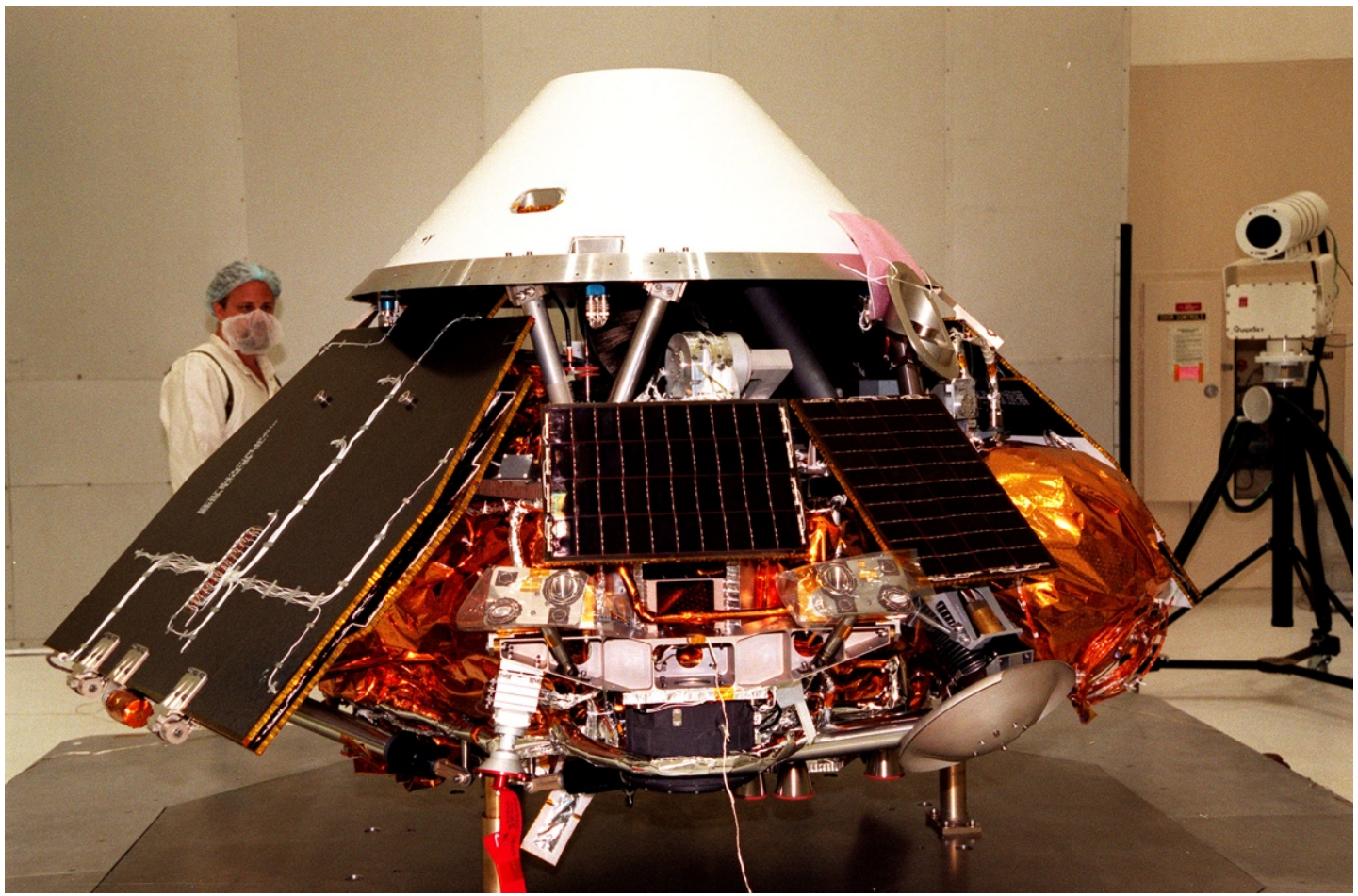
Mars Climate Orbiter



Mars Climate Orbiter ($85M, 1999)

- Study Martian weather and climate
- Problem: unit conversion problem
- Fix: *well...*

Mars Polar Lander



Mars Polar Lander ($120M, 1999)

- Study weather near the poles
- Problem: incorrectly set threshold
- Fix: *well...*

Patriot Missile Defence System

Patriot Missile (1991)

- Surface-to-air missile system for high to medium air defence
- First use in the Gulf War 1991
- Problem: clock skew
- Fix: reset the computer regularly until software patch

USS Yorktown Smart Ship

USS Yorktown Smart Ship (1998)

- Reduce manpower, maintenance, costs
- Problem: division by zero
- Consequence: database overflow killed propulsion system
- Fix: towing and software update

Oerlikon GDF-005

Oerlikon GDF-005 (2007)

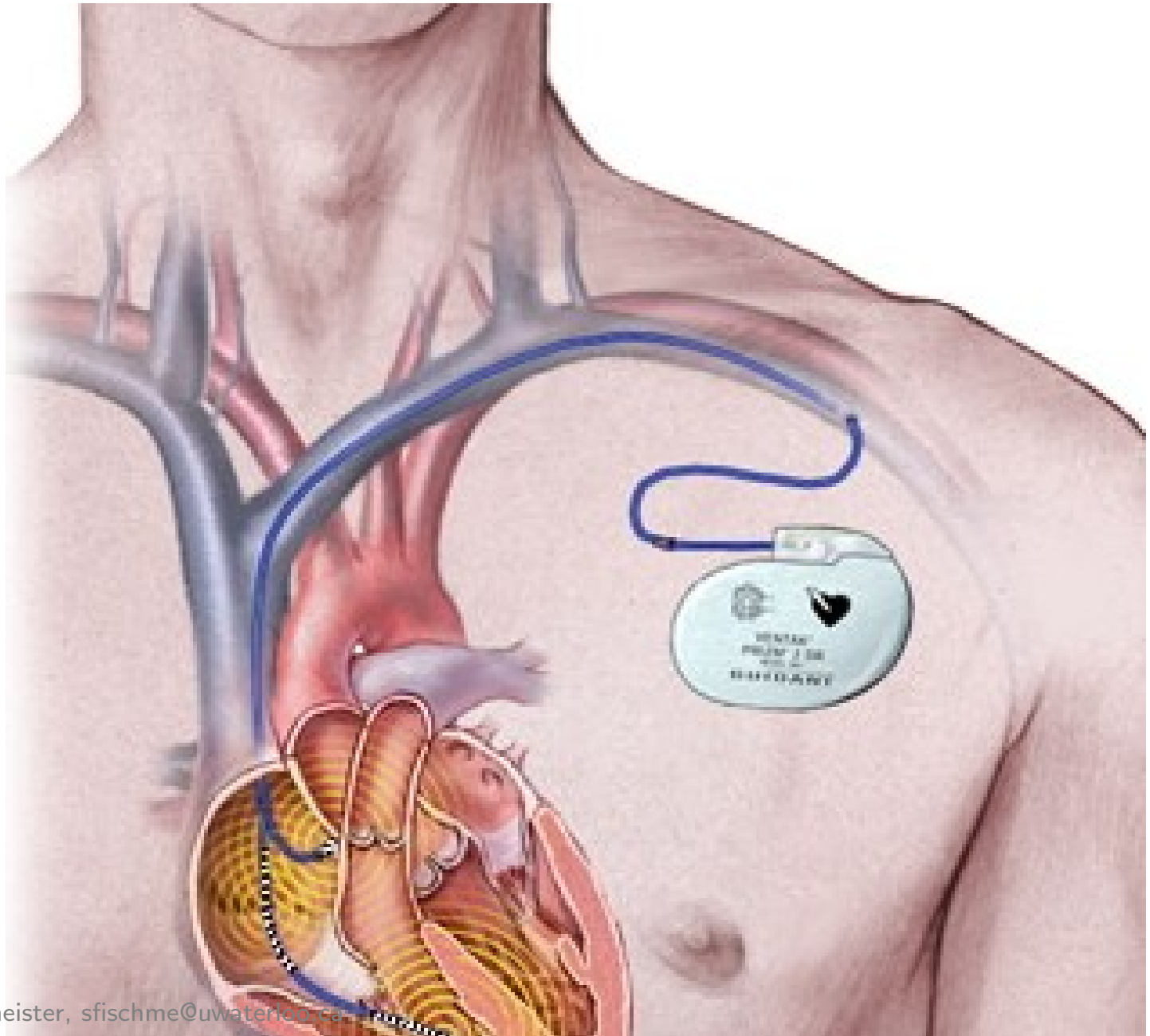
- Semi- and full-automatic ground to air defence system
- Problem: failure to detect faults
- Consequence: 9 soldiers killed, 14 injured
- Fix: none

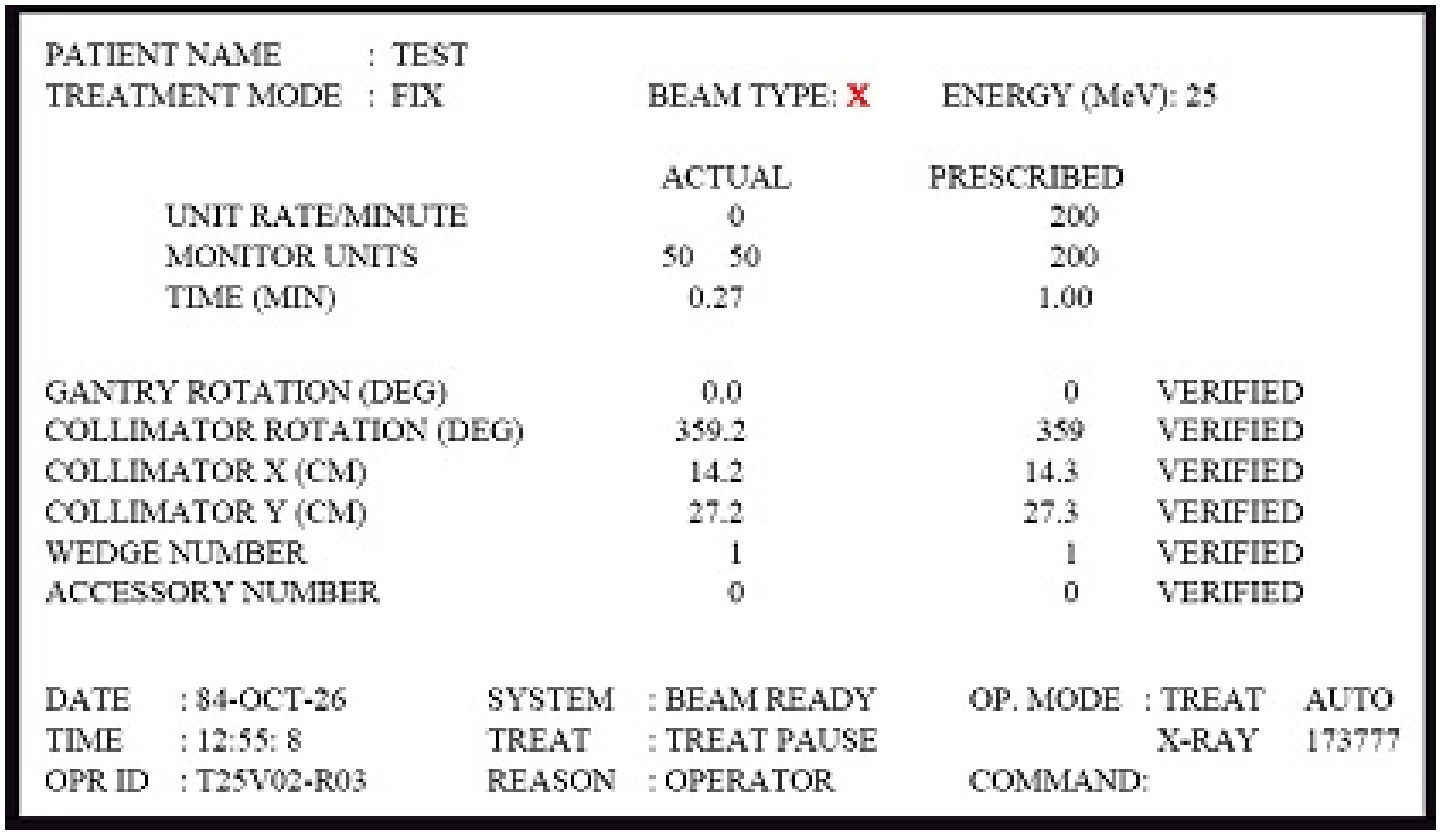
Quantas Flight 72

Qantas Flight 72 (2008)

- Goal: fly from Singapore to Perth
- Problem: incorrect Air Data Inertial Reference Unit
- Fix: an Operators Information Telex how to handle the situation

Pacemakers





PATIENT NAME : TEST
TREATMENT MODE : FIX BEAM TYPE: X ENERGY (MeV): 25

	ACTUAL	PRESCRIBED	
UNIT RATE/MINUTE	0	200	
MONITOR UNITS	50 50	200	
TIME (MIN)	0.27	1.00	
GANTRY ROTATION (DEG)	0.0	0	VERIFIED
COLLIMATOR ROTATION (DEG)	359.2	359	VERIFIED
COLLIMATOR X (CM)	14.2	14.3	VERIFIED
COLLIMATOR Y (CM)	27.2	27.3	VERIFIED
WEDGE NUMBER	1	1	VERIFIED
ACCESSORY NUMBER	0	0	VERIFIED

DATE : 84-OCT-26 SYSTEM : BEAM READY OP. MODE : TREAT AUTO
TIME : 12:55: 8 TREAT : TREAT PAUSE X-RAY 173777
OPR ID : T25V02-R03 REASON : OPERATOR COMMAND:

Therac-25

- Radiation therapy machine
- Problem: many $\rightarrow$ race conditions, overflow, missing safety interlocks
- 3 patients died!
- Fix: software updates

Someone has to build these systems!
Your life (will) depend on them.

Chances to learn about building safety-critical systems:

- ECE455 Embedded Software
- SE499: Independent project
- FYDP
- Undergraduate research assistant