McMaster University

McSCert

The Pacemaker Challenge

Mark Lawford, Ph.D., P.Eng.

Professor

Associate Director

McMaster Centre for Software Certification (McSCert)

McMaster University

IFIP WG10.4

1 July, 2012

Joint effort with Alan Wassyng, Brian Larson and Tony Ulrich

Outline

- About me, my collaborators, department & McSCert
- Software Certification
- The PACEMAKER Challenge
- Conclusions & Further Work

McMaster University

McSCert

About me

- **Mark Lawford**
  - Professor, Department of Computing and Software, McMaster University, B.Sc. (Queen's), M.A.Sc., Ph.D. (Toronto), P. Eng.
  - Software Engineering; Computer Aided Inspection and Verification; Application of Formal Methods to Real-Time Systems; Supervisory Control; *Software Certification*
  - Ph.D. is in control systems under supervision of Prof. W.M. Wonham
  - Worked at Allied Signal for 4 months on a HITL Real-Time Simulator for Environmental Control Systems for 777 & F22 during PhD
  - Turned down an NSERC Postdoc to go work at Ontario Hydro as a consultant on Systematic Design Verification of the Darlington Nuclear generating Station Shutdown System for 2 years
  - Joined McMaster in August 1998 to start up Software Engineering programs & then Mechatronics Engineering programs

McMaster University

McSCert

My Main McSCert Collaborators

Tom Maibaum

- Canada Research Chair in Foundations of Software Engineering
- Professor, B.Sc. (Toronto), Ph.D. (London), FRSA, FIEE, Ceng, P.Eng.

Software Engineering; Formal Specification; Software Architecture; Architecture Description Languages; Design Methods; Software Development Tools and Methods; Formalising Electronic Contracts; Deontic Logic; Epistemology of Software Engineering

Alan Wassyng, Director of McSCert

- Associate Professor, B.Sc., B.Sc. (Hons), M.Sc., Ph.D. (Witwatersrand), P.Eng.
- Safety-critical Software; Real-time/Embedded Systems; Tabular Expressions; Software Tools for Rigorous Software Development; Timing Issues in Requirements and Software Design.
- 13 years industrial experience in Safety Critical Software Development

McMaster University

McSCert

My Department-Computing & Software

- The Department of Computing and Software offers undergraduate programs in
  - Software Engineering, including one of the first accredited undergraduate software engineering programs in Canada,
  - Software Engineering for Embedded Systems
  - Software Engineering and Game Design
  - Mechatronics Engineering
  - Computer Science

- At the graduate level, the Department offers
  - Master of Applied Science, Master of Engineering and Ph.D. programs in Software Engineering
  - Master of Science and Ph.D. programs in Computer Science.
  - Master of Engineering, a course based 1 year program in Mechatronics

McMaster University

McSCert

McMaster Centre for Software Certification

- Leading a 5 year, $22 million Ontarion Research Fund – Research Excellence project on *Certification of Software Intensive Systems* with University of Waterloo and York University (Canada).
- Working with Industry and Regulators to improve software in:
  - Biomedical (FDA)
  - Nuclear (OPG, Candu, NRC, CNSC)
  - Financial Services (LSI) &
  - now Automotive (GM, IBM & ???????)
- Focused on product not process.

McMaster University

McSCert

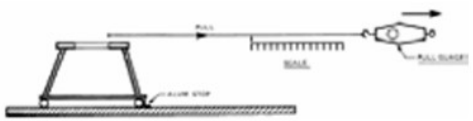
SE and Engineering

- We believe that software engineering is engineering:
  - need for principled methods
  - software should be built to same standards as any engineered artifact
  - people who produce software should be professional engineers
  - the norms and responsibilities of engineering should be applied
  - implies that rules about application of expertise should be applied

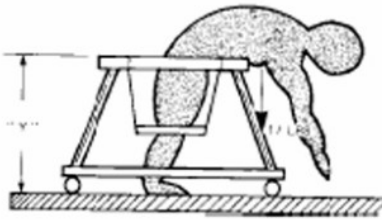
McMaster University

McSCert

Its the product, stupid.



(a) Stability Test 1



(b) Stability Test 2

Alan Wassyng, Tom Maibaum, and Mark Lawford, “On Software Certification: We Need Product-Focused Approaches”, Monterey Workshop 2008, LNCS Vol. 6028, Springer, 2010, 250-274.

McMaster University

McSCert

Software Certification

- A McSCert initiative that brings together many software engineering & safety engineering ideas to focus on the need for and the problems associated with software certification.
- Some sectors are regulated: nuclear power, civil aircraft and medical devices (highly imperfectly).
- In many areas (e.g. Automotive) there is no regulation of software
- But what do we know about software certification?

McMaster University

McSCert

What is Certification?

Def: *Certification* is the process of systematically determining, based on the principles of science, engineering and measurement theory, whether an artefact satisfies accepted, well defined and measurable criteria.

From Hatcliff et al, "A Software Certification Consortium and its Top 9 Hurdles," ENTCS, Vol. 238, No. 4, 11-17, 2009.

- By other names
  - Dependability through Assuredness (from Security viewpoint of DHS, DOD, Open Group &OMG
  - Compliance (from finance side – Sarbanes-Oxley)

McMaster University

McSCert

The Software Certification Consortium (SCC)

- We have established a North American consortium to pursue research and political aims related to software certification.
- We are working with regulators, like the US FDA and NRC, to improve existing “certification” based on known and future software engineering methods.
- We want to make certification product, not process based – A 5 Star frontal crash safety rating is based on the vehicle, not the manufacturing process!

McMaster University

McSCert

SCC's Objectives

- To promote the scientific understanding of software certification and the standards on which it is based;
- To promote the effective deployment of software certification standards;
- To promote public, government and industrial understanding of the concept of software certification and the acceptance of the need for certification standards for software related products;
- To co-ordinate software certification initiatives and activities to further the 3 objectives above

McMaster University

McSCert

A CAUTIONARY TALE

The FDA decides to ask for assurance cases for infusion pump submissions

FDA Staff: Guidance for Industry and FDA Staff Total Product Life Cycle: Infusion Pump - Premarket Notification [510(k)] Submissions DRAFT GUIDANCE.

U.S. Department Of Health and Human Services: Food and Drug Administration, Center for Devices and Radiological Health (April 2010)

McMaster University

McSCert

Clarifications and a Disclaimer

- I think that suggesting Assurance Cases for infusion pumps was the right choice.
- The FDA has made an effort to engage industry about their expectations regarding the guidance
- The following highlights some possible problem with the guidance as it stands.
- *The opinions here and elsewhere in the slides are my own and should not be construed to be the opinions of any other body.*

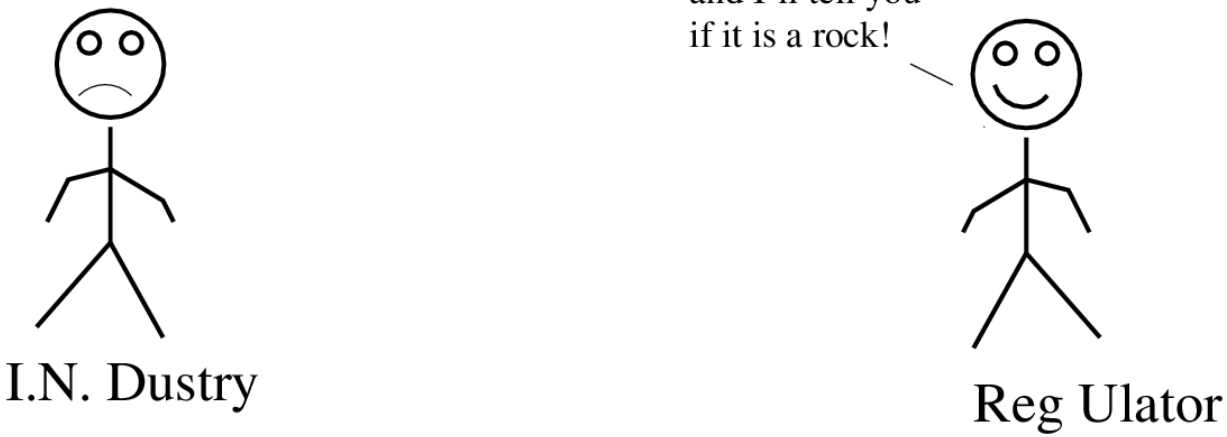
McMaster University

McSCert

Our Tale Begins

Problem: Standards have an implicit assurance case

Idea: Require an explicit Assurance Case!



McMaster University

McSCert

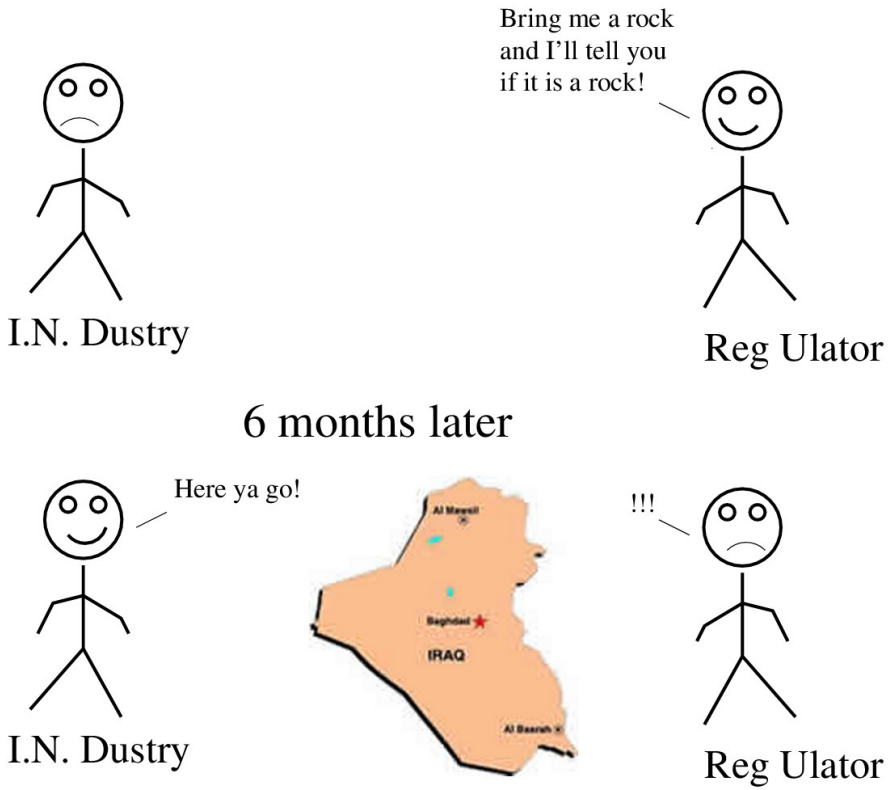
FDA Primes the Pump

- Creates Generic Infusion Pump
- Posts
  - hazards analysis,
  - safety requirements
  - Simulink model;
- Waits for the Assurances cases to start rolling in!

McMaster University

McSCert

Bringing Determinism to the Process?



Problem: Standards have an implicit assurance case

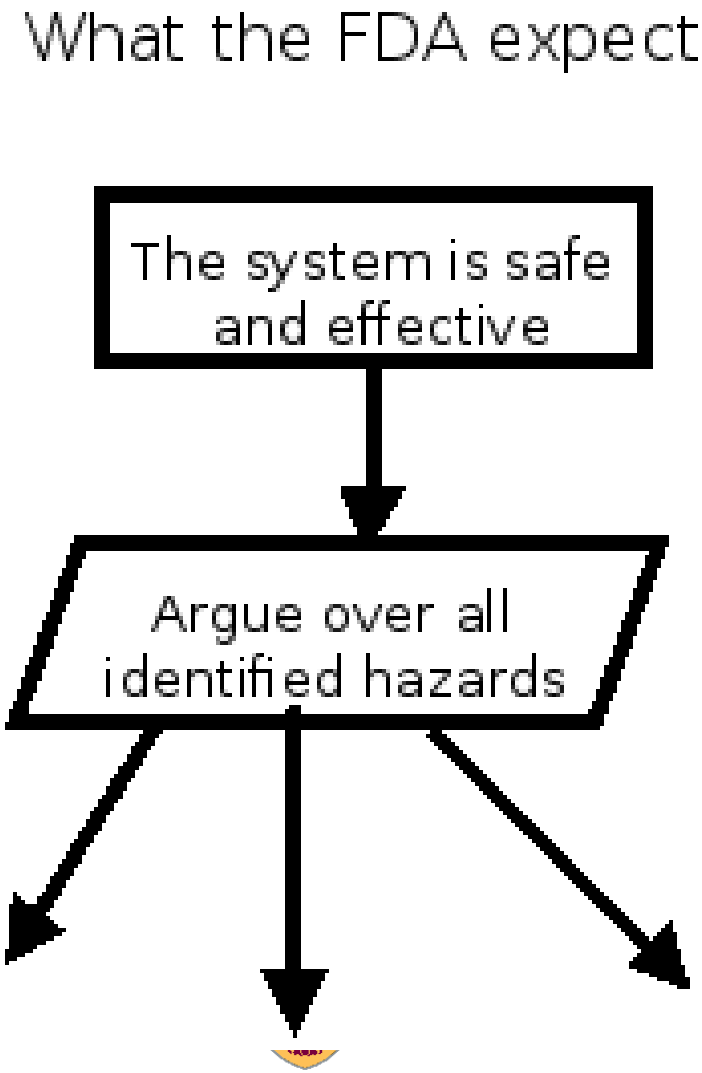
Idea: Require an explicit Assurance Case!

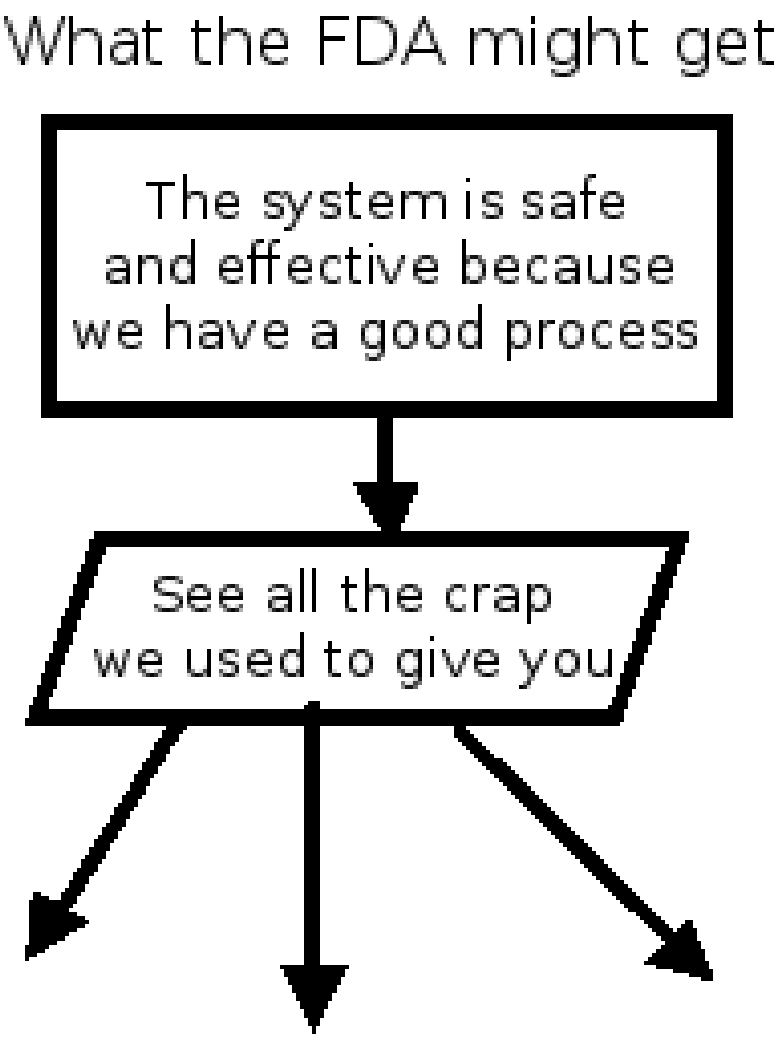
Problem: Assurance cases aren't standardized

McMaster University

McSCert

When every assurance case is a one off





The Solution?

- Dialog between the industry and the regulator to agree on what is expected
  - OPG did this on the Darlington Nuclear Generating Station – it works!
- Make the assurance case behind your standards explicit
- Standardize your assurance cases

Alan Wassyng et al, ``Software Certification: Is There a Case against Safety Cases?," LNCS 6662, 2011, 206-227

McMaster University

McSCert

Why have a challenge?

- Let's settle the arguments!
  - Product vs. Process
  - Standards vs. Assurance cases
  - Spark Ada vs. Event B
- The emergence of Cyber Physical Systems means certification is only going to get harder
- Foster technology transfer & stimulate research
- Help regulators & manufacturers sleep at night
- *Bring determinism to the certification process!*

McMaster University

McSCert

Bootstrapping Product Based Software Certification

- There is a lack of evidence about what evidence is needed for certification
- There is a lack of HQP educated in CS, (Control) Systems & Formal Methods
- There is a lack of focused research on certification
- We need to educate undergraduate, graduate & researchers about Formal Methods & what it takes to apply them in practice
- We need to stimulate further research on certification of Software Intensive Systems

McMaster University

McSCert

What is the Pacemaker Challenge?

- A hardware reference platform developed at the University of Minnesota with help from Brian Larson of Boston Scientific
- A 10 year old informal (English prose) pacemaker requirements document (35 pages) from Boston Scientific:

3.6.3 Pace-Now State

Commanded emergency bradycardia pacing (Pace-Now) shall be available. The Pace-Now Pace parameter values are as follows:

1. The mode Pace-Now pace parameter shall have a value of VVI.
2. The lower rate limit Pace-Now pace parameter shall have a value of 65 ppm ±8 ms.

What is the Pacemaker Challenge?

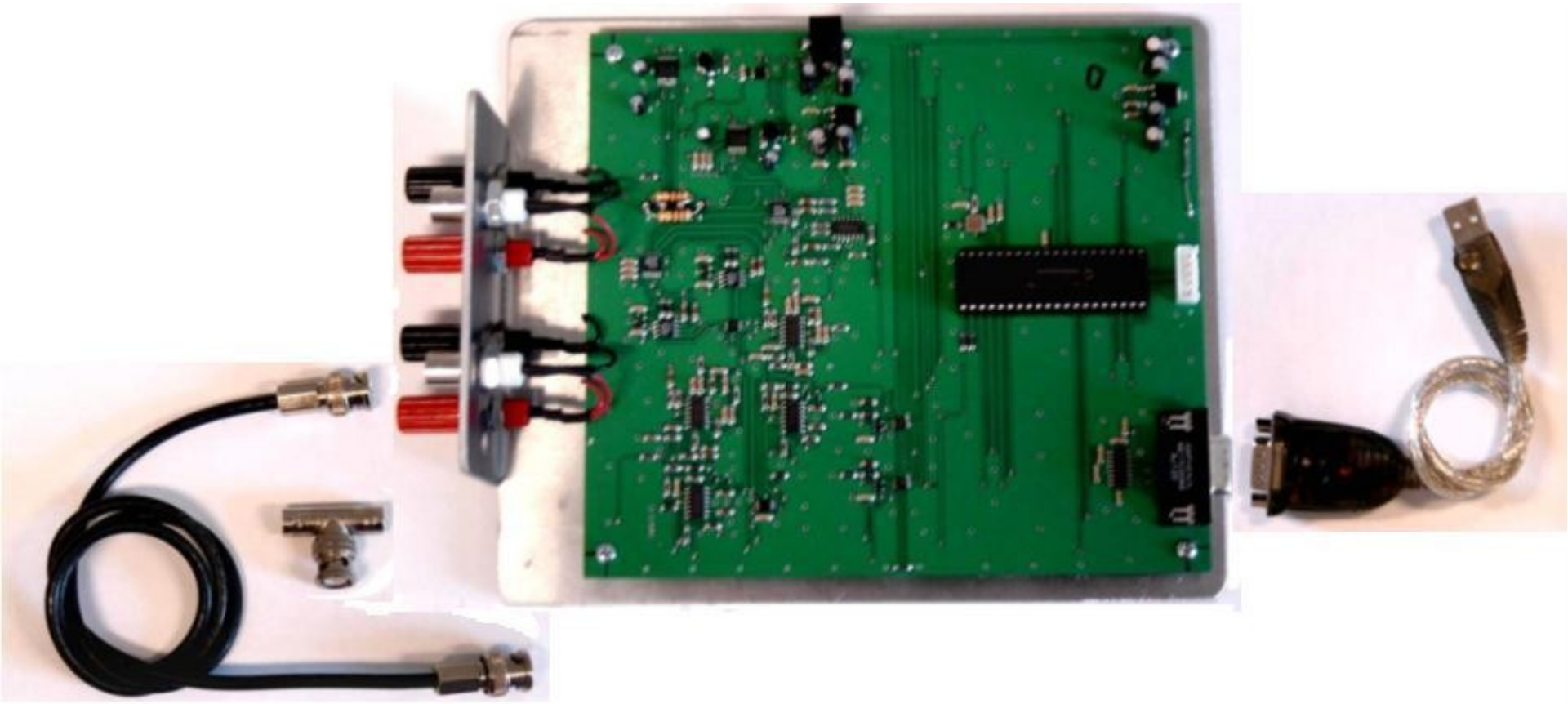
- A hardware reference platform developed at the University of Minnesota with help from Brian Larson of Boston Scientific
- A 10 year old informal (English prose) pacemaker requirements document (35 pages) from Boston Scientific:

3.6.3 Pace-Now State

Commanded emergency bradycardia pacing (Pace-Now) shall be available. The Pace-Now Pace parameter values are as follows:

1. The mode Pace-Now pace parameter shall have a value of VVI.
2. The lower rate limit Pace-Now pace parameter shall have a value of 65 ppm ±8 ms.

Pacemaker Hardware



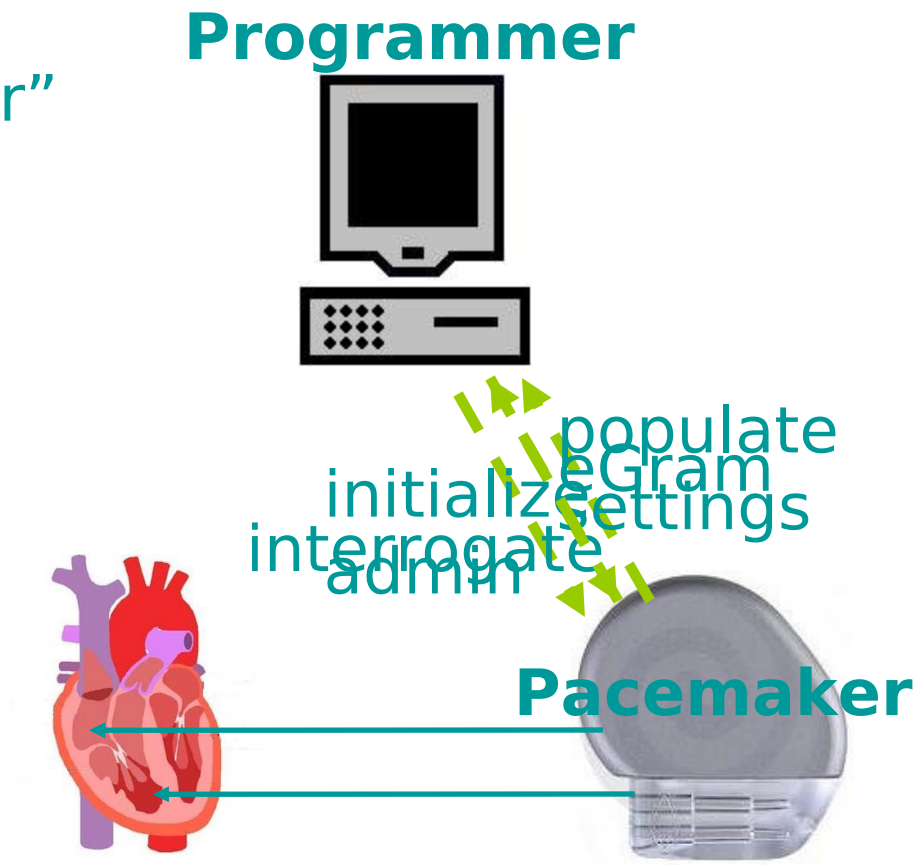
Hardware Details

- 8-bit PIC 18F4520 microcontroller with 32k program memory & 1.5 k RAM, 256 bytes EEPROM
- 64 different part #'s - total 227 discrete components
- RS-232 serial interface
- Non-standard 5 pin programming header
- You need external connections to simulate heart – we have binding posts and BNC
- Can program using Microchip's MPLAB & C18 compiler
- Also need an additional "programmer"
  - e.g. Microchip's ICD2 ($150) or PICKit2 ($50)

McSCert

Device Controller Monitor (DCM)

AKA “the programmer”



Submission Review Panel

- Rick Chapman, US Food & Drug Administran (FDA)
- Brian Larson, former Research Scientist, Boston Scientific
- Mark Lawford, P.Eng., McSCert, McMaster University
- David Tremaine, CEO, SWI
- Alan Wassyng, P.Eng., Director of McSCert, McMaster University

McMaster University

McSCert

Pacemaker Formal Methods Challenge

- Think your methods and tools are the best?
- Prove it!
- We are challenging the Formal Methods Community to “solve” the pacemaker problem
- You can go from requirements to code or anything in between and make a submission
- All submissions will be judge by a panel of academic, industry & FDA representatives

McMaster University

McSCert

Why Should you be Interested?

- It's a great "real world" problem that is not too big.
- We can make you (in)famous!
  - Win the competition and you'll get bragging rights
  - You can generate papers! (preliminary pacemaker papers already in FM 2008 & elsewhere)
- You'll run into issues that your methods don't handle well that motivate further research
- The FDA and industry are part of review panel
- If you are a company you can see how FM researchers tackle a complete real problem

McMaster University

McSCert

Current Status of the Challenge

- There is a Pacemaker Wiki: http://www.cas.mcmaster.ca/wiki/index.php/Pacemaker
- Still need to develop submission guidelines
- Pacemaker has been chosen for SCORE – ICSE programming competition
- Produced 45 prototype boards that are now on sale sold out @ $350/board + shipping
- The Pacemaker Challenge has been used at McMaster for undergraduate and graduate courses

McMaster University

McSCert

The Pacemaker at McMaster

- 23 final year software engineering undergraduate students' senior thesis project was the pacemaker
- 5 team of 4-5 students were formed
- Teams did 3 revisions of complete documented system
  - Rev. 0: A slice for VVI mode
  - Rev. 1: Relatively complete VVI, DDD, DDDR & DCM
  - Rev. 2: Complete formally document system
- Students were required to use PVS theorem prover to check correctness of at least a few tabular specifications from requirements

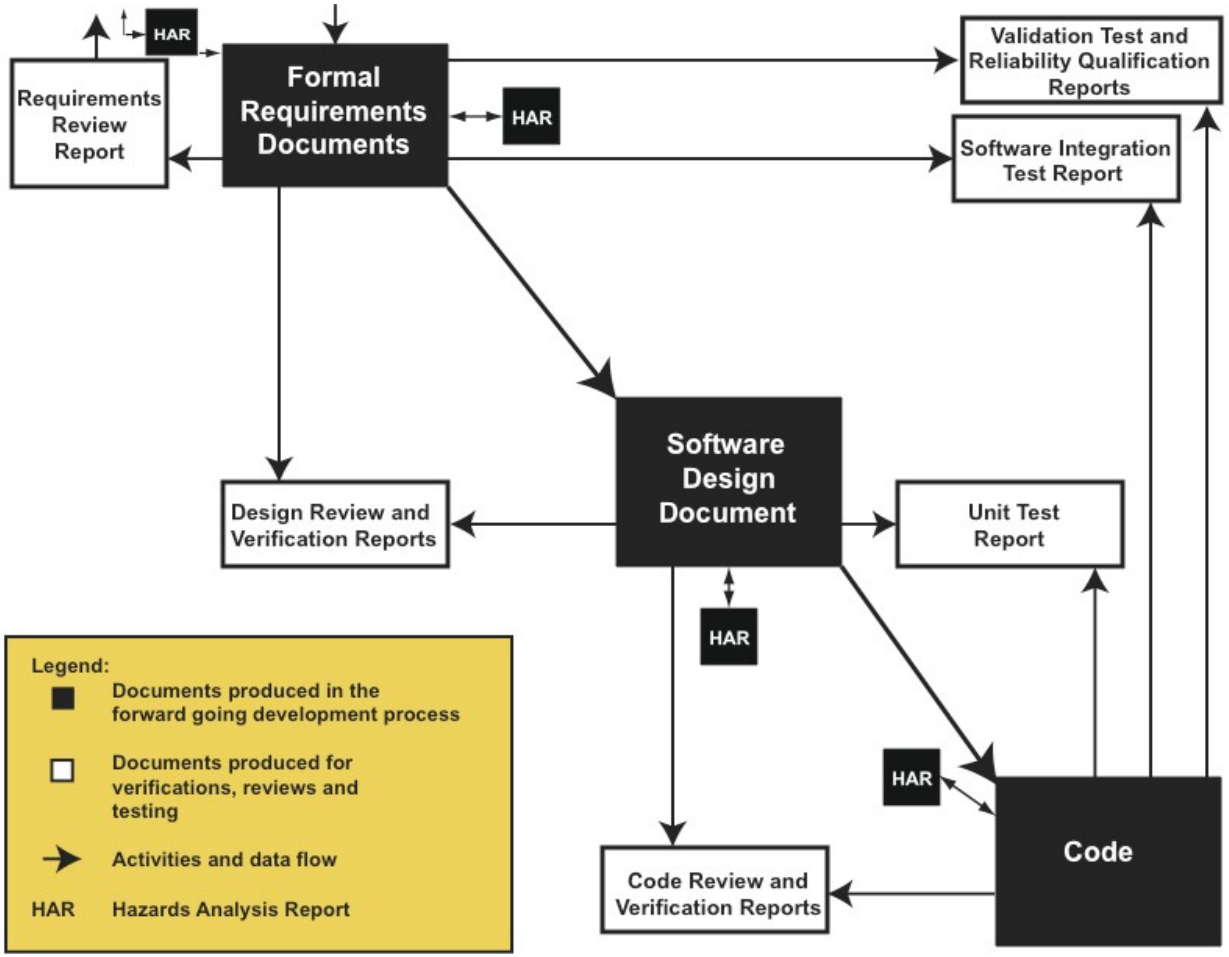
McMaster University

McSCert

Idealized Process Used

Simplified version of Wassyng & Lawford, Lesson learned from a successful implementation of formal methods, FME'2003, LNCS 2805, 2003, 133-153.

Students were encouraged to "Fake it" al la Parnas RDP



McMaster University

Pacemaker Modes

	I	II	III	IV (optional)
Category	**Chambers Paced**	**Chambers Sensed**	**Response To Sensing**	**Rate Modulation**
Letters	O–None A–Atrium V–Ventricle D–Dual	O–None A–Atrium V–Ventricle D–Dual	O–None T–Triggered I–Inhibited D–Tracked	R–Rate Modulation

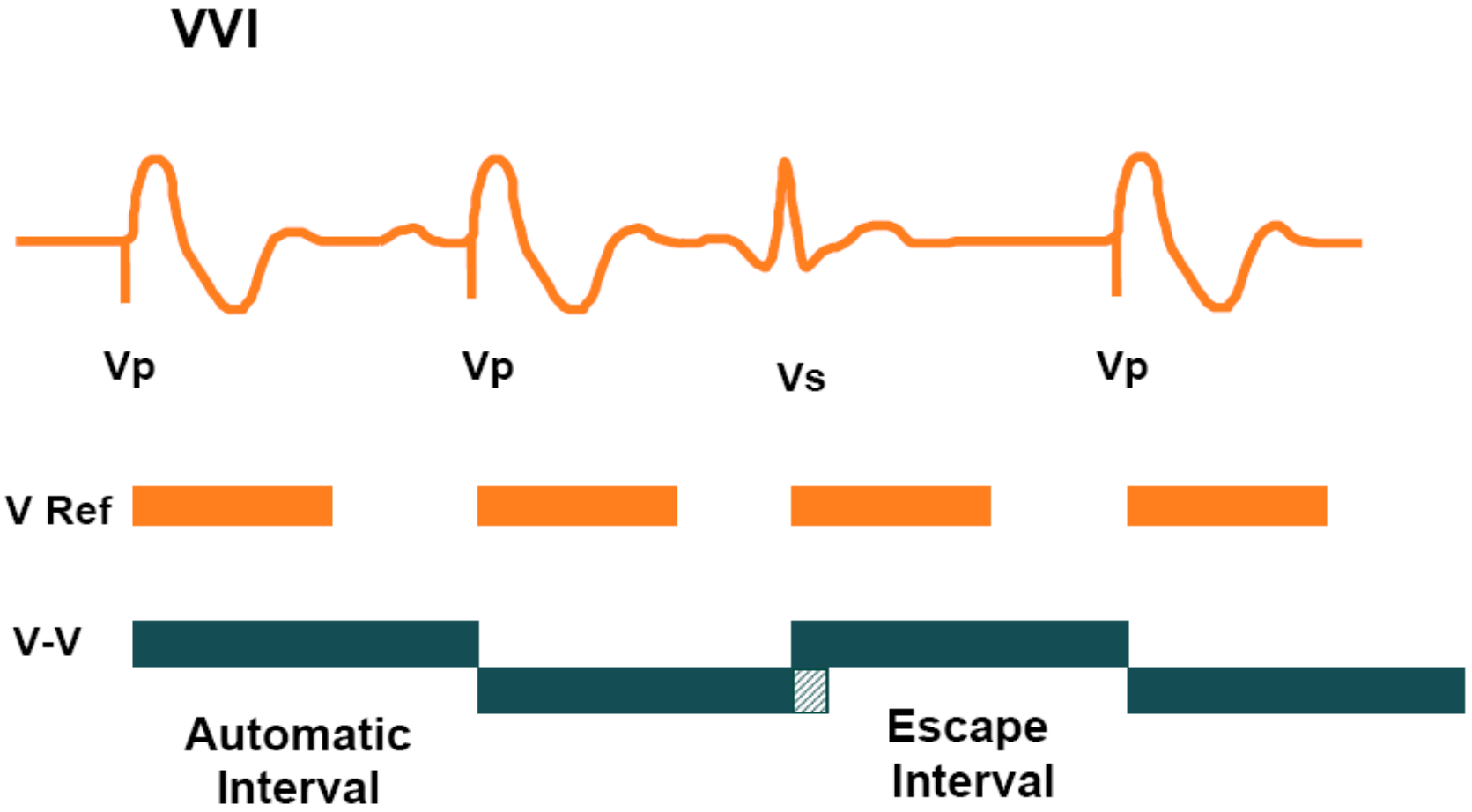
Table 2: Bradycardia Operating Modes

- VOO – “open loop” Ventricle pacing
- VVI – Ventricle paced, Ventricle sensed, response to sense inhibited
- DDDR – Both chambers paced & sensed and rate modulated depending upon activity level

McMaster University

McSCert

Pacemaker: VVI Mode



McMaster University

McSCert

Formalizing VVI Mode

Condition	Result
	c_vp
No spontaneous heart beat has Held For escape interval & No pace initation has Held For the automatic interval	Pace
Pacing has Held For the pace width	Stop Pacing
Otherwise	No Change

Formalizing VVI Mode

Condition	Result
	c_vp
(f_blanking OR m_vs_{-1} OR NOT m_vs) Held For(k_escape) & (c_vp_{-2} = k_vPacedAmp OR c_vp_{-1} = 0) Held For(k_automatic)	k_vPaceAmp
(c_vp_{-1} = k_pacedAmp) Held For(k_pacedWidth)	0
NOT[(f_blanking OR m_vs_{-1} OR NOT m_vs) Held For(k_escape) & (c_vp_{-2} = k_vPacedAmp OR c_vp_{-1} = 0) Held For(k_automatic)] & NOT[(c_vp_{-1} = k_pacedAmp) Held For(k_pacedWidth)]	No Change

McMaster University

McSCert

A more complicated requirement

Condition			*Results* f_bVPace
f_bInVSP	$\neg\ f\_bInPAVB$	$(f\_bVPosEdge = False) HeldFor-(p\_nVSP - p\_nPAVB)$	False
		$\neg(f\_bVPosEdge = False) HeldFor-(p\_nVSP - p\_nPAVB)$	True
	$f\_bInPAVB$		False
$\neg\ f\_bInVSP$			False

This table is taken from one of the student groups SRS!

McMaster University

McSCert

What Happened?

- Hardware was late – project start in Sept. 07 and hardware was available Jan 31, 2008
- Hardware had “issues” they discovered and had to try to diagnose and workaround.
- Rev. 0 of the documents was terrible – but students had no complete examples
- Students had to learn process, document style, hardware, embedded programming & pacemaker domain

McMaster University

McSCert

Outcomes

- All 5 groups produced working firmware/DCM combinations for VVI, DDD and DDDR
- Each group had unique features
  - Software workaround for reed switch problem
  - DCM event logging for audit trail
  - Remote patient database with full encryption
- What is weak: Hazards analysis, integration and system level testing
- We are still using it in UG software dev course

McMaster University

McSCert

Student Feedback

- At start:
  - Not happy with application, support, expectations
- At the end - Overwhelmingly positive:
  - “I learned more in this course than I did in all the courses in the previous 3 years.”
  - “I would spend all my time on this course if I had my choice.”

McMaster University

McSCert

Example Student Submission

- 3855 SLOC of C code (including comments) for pacemaker firmware + 2004 SLOC unit tests
- Object code uses 9410 x 16 bit words of the 16384 x 16 bit words of available program memory
- DCM is 9298 SLOC of C#

McMaster University

McSCert

Lessons Learned from McMaster Experience

- It is possible to tackle the Pacemaker problem with reasonably trained undergraduate students using formal methods where appropriate
- Integrated simulation environment for executable formal tabular specifications would have improved earlier revisions
- Start with VOO, then VVI then other modes
- Real-world system and hardware constraints will expose implicit assumptions in your formal models

McMaster University

McSCert

Why has the Challenge failed?

Large Man with Dead Body: Who's that then?

The Dead Collector: I dunno, must be a ~~king~~ FM Researcher.

Large Man with Dead Body: Why?

The Dead Collector: He hasn't got ~~sh*t~~ code all over him.

McMaster University

McSCert

Why has the challenge failed

- Lack of Test suite/harness
  - Almost got a “test heart” from Boston Scientific
  - But Were no good open source heart models
  - It takes a lot of time to get a good test suite (ask Ken) and they are highly proprietary
- Awkward development board
  - PIC18F4520 has awful C compilers
  - RS-232 & separate power supply puts off a lot of people

McMaster University

McSCert

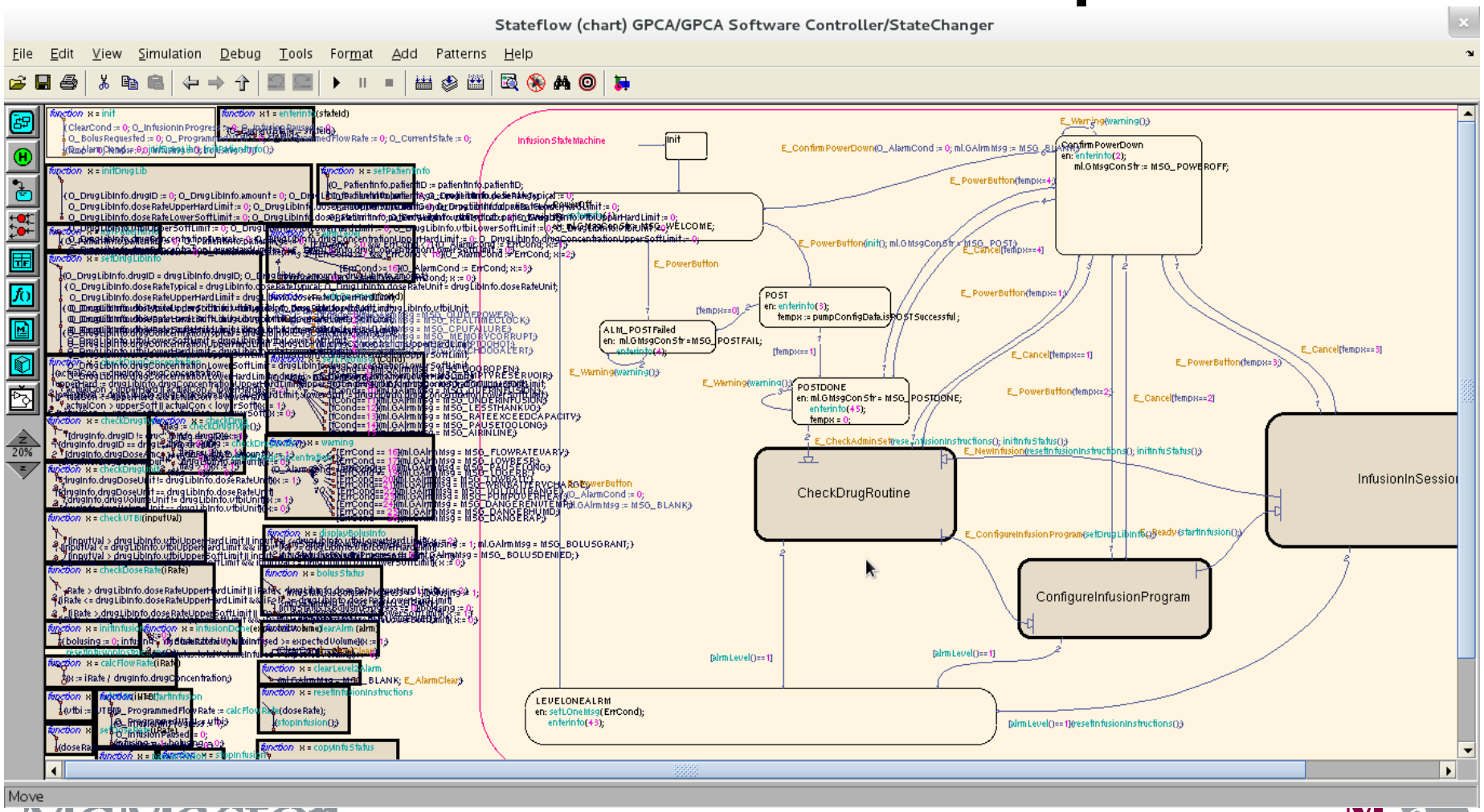
If you build it, they won't come

But if you integrate it, they might ...

McMaster University

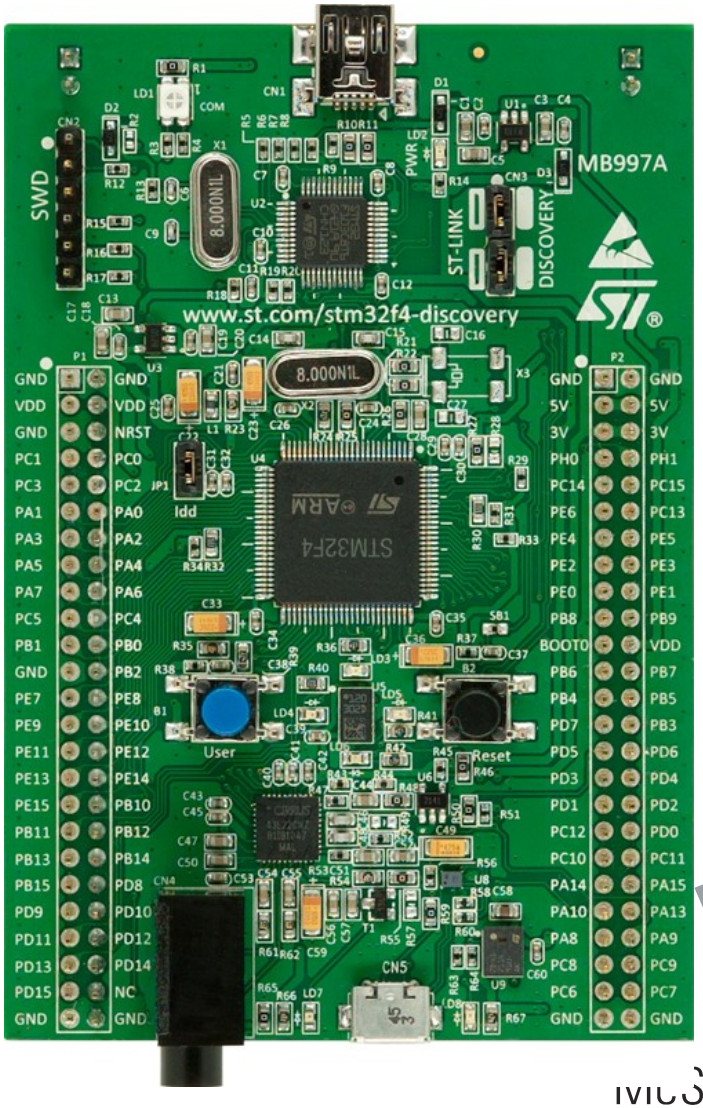
McSCert

A Simulink Model is NOT the Requirements



Where Do We Go From Here?

- Update platform to use "modern" dev kit (e.g. STM32F4 Discovery)
  - Create Simulink blocks to handle interfacing
  - Provide same device interface code
- Should we make it a wireless link to add in security issues?



McMaster University

Questions?

Before IFIP WG 10.4

After IFIP WG 10.4

McMaster University

McSCert