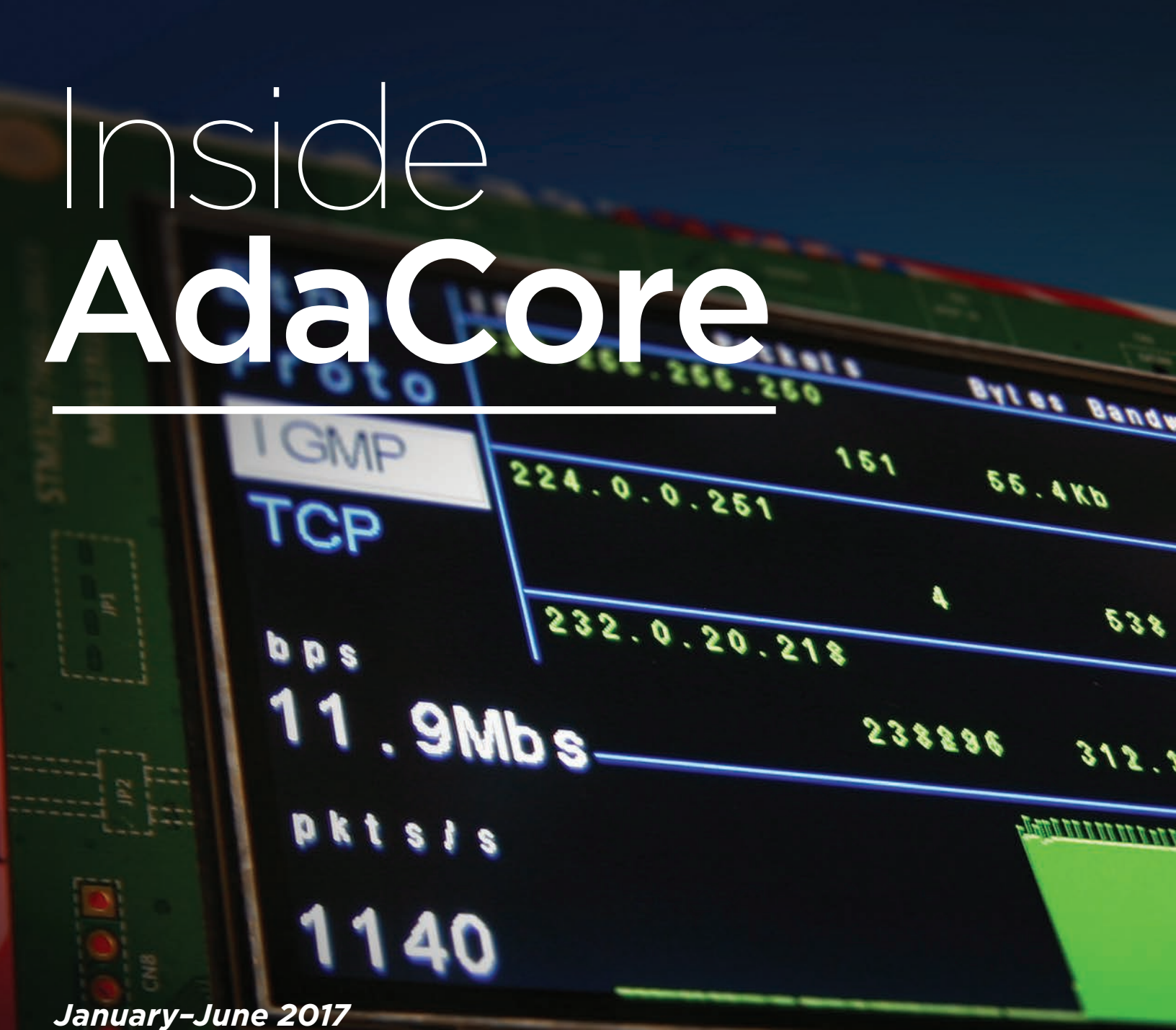


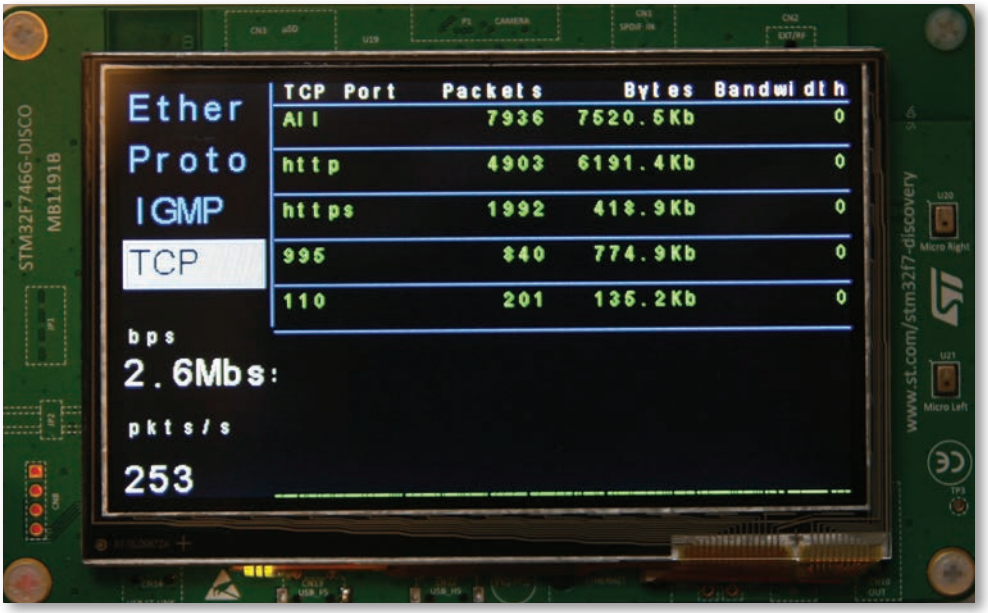
Inside AdaCore

January–June 2017

- *Make with Ada* Prize Winners Announced
- Tech Days 2016 Conducted in Paris and Boston
- Interview with Quentin Ochem
- Version 17.1 Product Releases
- Spotlighting a GAP Member:
  The Australian National University (Canberra)
- GNAT Pro Supports AddressSanitizer

Make with Ada Prize Winners Announced

The winners of the first annual *Make with Ada* programming competition were announced at the end of November. The challenge posed by this competition, which ran from June through September 2016, was to use Ada or SPARK to develop an embedded software project targeted to an ARM Cortex M or R processor. Open to individual contributors and small teams, *Make with Ada* attracted entries from around the world, and the submitted projects ranged from food dehydrators to robots to electronic power supplies. AdaCore sponsored this competition to demonstrate how Ada and SPARK can help produce reliable embedded software, and the entries were evaluated against the criteria of dependability, inventiveness, openness, and collaborativeness.



Above: LCD display from the 1st-place project EtherScope, showing TCP traffic. Photo: Stephane Carrez

The 1st place prize of €5000 was awarded to Stephane Carrez for his EtherScope project running on an STMicroelectronics STM32F746 board (Cortex M7). The software analyzes and monitors Ethernet traffic by reading Ethernet packets, conducting real-time analysis, and displaying the results on a 480x272 touch panel. A video of EtherScope is available at **youtu.be/zEtA-S5jvfY**. All source code for the application is available on github at **github.com/stcarrez/etherscope/**.

The 2nd place prize of €2000 was awarded to German Rivera for his Autonomous Car Framework to develop control software for the NXP cup race car (**community.nxp.com/docs/DOC-1284**). The development system is a miniature race car chassis coupled with a FRDM-KL25Z board and a TFC-shield board. The car kit includes two DC motors for the rear wheels, a steering servo for the front wheels, and a line scan camera.

The 3rd place prize of €1000 was awarded to Shawn Nock for his Bluetooth Beacons "iBeacons" project. The target was a Nordic Semiconductor nRF51822 System-on-a-Chip (a Cortex-M0 part with integrated 2.4GHz Radio) with a custom development board.

Two special prizes (Crazyflie 2.0 nano drones) were also awarded. The Robert Dewar Special Prize for dependability went to German Rivera for his IoT Networking Stack project for the NXP FRDM-K64F board. The Lady Ada Lovelace Special Prize for inventiveness was awarded to Sébastien Bardot for his Explorer and Mapper Robot project, which uses a Nucleo F411RE board based on an STM32F411RE chip from STMicroelectronics.

One of the goals of *Make with Ada* was to show that embedded system developers with little or no previous experience with Ada could quickly become productive with the language and its toolset. This objective has been met; as expressed by prize winner Shawn Nock on his blog at **nocko.se/blog/make-with-ada-redux/**: "I was able to become productive in Ada in about 8 hrs (starting from zero experience). . . . Developing the beacon [project] in Ada (a language I don't know well) took roughly the same time as the similar functionality in C and I have more confidence in the Ada code."

Information about the 2017 competition will be available during Q2 2017 at **www.makewithada.org/**.

Tech Days 2016 Conducted in Boston and Paris

AdaCore's Tech Days conferences in 2016 took place in Boston during September 21–22, and in Paris on October 6. These annual events provide attendees with in-depth information about the company's current and future products, and an opportunity to meet AdaCore staff and also other AdaCore customers. A wide range of topics were covered, including a roadmap of planned features in the major product lines (GNAT Pro, CodePeer, SPARK Pro, QGen), an integrated look at verification solutions that incorporate AdaCore's tools, and brief snapshots of several selected technologies: the libadalang semantic analysis utility, the Ada drivers library for bare metal platforms, and the "integrated unit testing" approach that simplifies the tool qualification effort for QGen. The Boston Tech Days included training on CodePeer, SPARK Pro, and QGen from the company's technical experts as well as an open session where customers discussed their experience with AdaCore products. The Paris Tech Days event featured user presentations from Thales and Altran.

Both events were very well received. Attendee responses to a post-conference survey cited "meeting the AdaCore team, and the larger community that share our interest in Ada and software assurance" and "getting insights into AdaCore's very proactive and dynamic nature" as particularly valuable.

A selection of video presentations from this year's events will be available at **www.adacore.com/techdays/presentations/**.

Version 17.1 Product Releases

New Product Numbering and Release Scheme

To simplify and unify the version numbering and annual release cycle for AdaCore products, a new set of conventions is being adopted in 2017. Each product will have a version number of the form *YY.n* where *YY* identifies the year, and *n* is either 0 (the preview release in October of the preceding year), 1 (the major release in February), or 2 (the corrective release in July). Thus 17.1 is the version number for each AdaCore product being released in February 2017.

GNAT Pro

- Based on GCC 6 and GDB 7.10
- Improved elaboration order algorithm
- Support for alternativee and faster linker "gold" under native GNU/Linux
- Extended Ravenscar profile available on bare metal platforms
- Support for SMP on leon3
- XMLAda, GNATcoll and AWS come precompiled with native GNAT Pro packages
- Debugger improvements
  - Exception message now displayed when stopping on an exception
  - Support for task switching when debugging a core file
- GPRbuild
  - New default output to improve usability
  - --build-script switch to generate standalone build scripts
  - Support for encapsulated libraries under GNU/Linux 64bits
  - Export from library can be limited to symbols in its interface
- GNATtest
  - Improved support for stubbing
  - Improved GNATcoverage integration
- GPS
  - Improved debugger support: new default perspective, variables tree view, redesigned breakpoint view, ability to set breakpoint before starting the debugger, improved call stack view, enhanced Python scripting, improved support for embedded ddebugging via OpenOCD support
  - Redesigned handling of bookmarks
  - New preferences dialog, including key shortcuts, plug-ins, and help messages
  - Menu bar for floating windows
- GNATbench
  - Support for Eclipse Neon (4.6.x)
  - Support for VxWorks 653 3.1

QGen

- Enhanced user interface
- Improved code generation and support for additional blocks
- Processor-In-The-Loop support
- QGen subset checkers
- Model-Level debugging
- Better integration with external code

CodePeer

- "No False Positives" mode, to suppress likely false alarms
- Improved precision and new messages
  - Loop unrolling and better handling of static array bounds
  - Improved analysis of unchecked conversions, access checks, test always true
  - Global objects used in subprograms displayed in annotations
- User Interface
  - Ability to review multiple messages at once from GPS
  - More information displayed when reviewing messages
  - More powerful and easier to use handling of baseline runs
  - Ability to exclude complete source directories from analysis
  - Faster loading of messages

SPARK Pro

- Improved proof automation
  - CodePeer engine available as an additional prover
  - Generation of loop invariants for frame conditions
  - Library of lemmas for non-linear arithmetic
  - Finer granularity in proof of conjunctions
- Improved proof interaction
  - Improved counterexamples (arrays, quantifiers, enumerations)
  - New mode statistics for proof results
  - Display mode for proof
  - Replay mode for proof
- Support for type invariants

GNATcoverage

- Support for ARM bare metal, native Windows 32bits and Windows 64bits
- Improved object coverage
- Support for incremental coverage analysis (consolidation of partial results)
- Support for incidental and stub coverage confinement

Quentin Ochem

Lead of Business Development and Technical Account Management

Quentin, tell us about your background and how you came to be involved with Ada and AdaCore. What is your current role?

My first exposure to programming was as a hobby in high school. I played pretty extensively with QuickBasic, C++, and Java for about 3 years, which led me to select computer science as a major in college (Aix-Marseille Université). At first I was upset when I realized I was going to have an entire semester on Ada: this "weird obsolete language that was used for nothing but blowing up space rockets". This was back in 2000. But I fell in love with Ada on the spot. What attracted me was the possibility to specify "stuff" in the code. All of the sudden I wasn't manipulating integers or floats but days, oranges, meters or francs (and, a bit later, euros)! I subsequently learned that my university had good contacts with AdaCore (then ACT Europe). Arno Charlet happened to have studied at the same place some years earlier. Following a suggestion from one of our teachers, I participated in a programming contest organized by Ada-Europe. However, I only reached second place because "I didn't document my code enough". Meh. After that, Arno brought me in for an internship in Paris in 2002, then a second one in New York in 2005, after which I was hired. I started as an engineer, working on technologies such as GNATbench, GNAT-AJIS and the GPS smart-completion engine. I guess my documentation skills didn't improve enough, as I gradually got transferred to the commercial team. I started as a sales engineer (or technical account manager) and eventually took the lead of these activities worldwide. Today, I'm also responsible for business development; in other words reaching out to new areas for the Ada language and AdaCore's technology. It's ironic that years after my initial exposure to Ada, I find myself in the role of explaining to people why it isn't a "weird obsolete language that was used for nothing but blowing up space rockets".

What kinds of challenges and opportunities do you foresee in your role as Business Development Lead?

Business Development means spending a lot of time discussing with people for whom we almost—but don't quite—have the right solution. Challenges and opportunities are actually the same thing here—it's about identifying patterns of requirements that we can serve better.

As a specific example, we're finding a growing number of projects that would "freeze" on a specific version of our technology. This might be due, for example, to certification requirements, or to long-lasting maintenance contracts. But after a while, the customer would start to request fixes to critical issues, and we only knew how to meet this need on an ad hoc basis. We will soon be releasing an off-the-shelf offering to address this issue. It will provide product versions for which we can perform critical updates on a standard basis.

I also have encountered software developers from traditional C-based environments who are looking for solutions that can increase the reliability of their code. They are typically working in domains that are outside of our traditional Aerospace and Defense market; examples are medical devices, industrial Internet of Things, automotive and industrial automation. They're not looking at Ada in particular, but rather at what's available to help improve their software; Ada just happens to be a good match. A technology that has a track record of outperforming in safety-critical areas for decades can be extremely appealing. This is a relatively new trend for us, which we're getting ready to address.

And so, the whole game from a business development perspective is twofold. First talk to people, understand their constraints, and come up with products and solutions that match what they need. In this case, that would be a product that is very ARM-oriented, with a subset of the tools and services that we usually provide. The next step, when that product is confirmed to be a match, is to bring it to a larger set of users. This is a very exciting time for us!

Any hobbies or outside interests that you'd like to share?

My twin children (a boy and a girl) were born in February 2016, so whatever time my wife and I had for hobbies has shrunk to a bare minimum! But when everyone else is asleep, I'm an avid consumer of TV series. I'm constantly amazed by the quality of the writing and dramaturgy that gets injected into those things, so much so that movies feel like something in the past. Since my "day job" at AdaCore does not really involve coding, I have also returned to my teenagehood hobby: programming. I'm having a lot of fun playing with Unity 3D and writing small video games. The power of the engine they give out is truly amazing!

---

professional training

Public Ada Course in the UK, April 2017

A 5-day course on Ada fundamentals will be held during April 24–28, 2017, in Yeovil, UK. The course, which comprises live lectures and hands-on lab exercises with AdaCore's GNAT Ada toolsuite, covers the essentials of software development with Ada. Among the topics are strong typing, packages, generics, safe concurrency (including the Ravenscar profile), and systems-level programming. The only prerequisite is familiarity with some programming language (such as C, C++ , or Java); no previous exprience with Ada or the GNAT technology is required. Attendees will receive John Barnes' *Programming in Ada 2012* textbook and an ARM Cortex based single board computer from STM Microelectronics. Supplementary material, including AdaCore's *Ada for the C++ and Java Developer* booklet, will also be distributed. For registration information and a detailed course outline. please visit **www.adacore.com/public-ada-training/.**

Quentin Ochem photograph by Gary Matoso

Spotlighting a GAP Member

The Australian National University (Canberra, Australia)

Ada is used in several courses at the Australian National University (ANU), and recent experience has confirmed Ada's benefits as a teaching language in undergraduate computer science programs.

In 2015 the senior-year course "Real-Time and Embedded System", conducted by Dr. Uwe Zimmer, was extended to cover hardware-level real-time programming concepts at a higher abstraction level than is provided by more customary low-level programming constructs. According to Dr. Zimmer, "AdaCore's support for the Ada Ravenscar profile on the STM32 family was exactly what was needed for this transition." The students implemented a complex distributed real-time system on their individual hardware configurations, connecting their devices in different network topologies. This was a challenging assignment, but it demonstrated that undergraduate students were able to design and implement real-time software at an abstraction level that reflected the essential aspects of the problem space in a hardware-independent fashion, while maintaining the necessary hardware-level control.

Owing to the success of this course, Dr. Zimmer is planning to keep the approach and to further expand the class in 2017.

Ada is also a principal language in a second-year course at the ANU, "Concurrent and Distributed Systems". Ada has been used in this course since 2000, and the Ada tasking model has been instrumental in conveying the principles of structured concurrent programming. For example, students have come to appreciate Ada's protected object feature for state-based mutual exclusion as simpler and more reliable than constructs such as monitors and signals/semaphores as are found in other languages. Dr. Zimmer reports that the students in this course have consistently come away with a solid understanding of the issues raised by concurrent systems and how to solve them.

In brief, Ada continues to show its value at the ANU as a language for teaching effective software engineering techniques for the most demanding types of applications. Students use Ada to successfully conceptualize complex real-time and concurrent systems at a high level of abstraction while retaining the necessary low-level control.

GNAT Pro Supports AddressSanitizer

The GNAT technology now supports the gcc AddressSanitizer utility, which detects run-time memory errors such as out-of-bounds references and use-after-free bugs. Although many of these errors would be prevented because of Ada's static or dynamic checks, others could arise in low-level programs where the checks are absent.

To use AddressSanitizer, pass the `-fsanitize=address` switch to the compiler; the relevant instrumentation for run-time monitoring will be added to the object file. Here are some examples:

Buffer overflow

Data objects of different types may be overlaid in Ada using an address clause, but if their sizes are different any attempt to access a memory location that is in the larger object but not the smaller will be reported as an error by AddressSanitizer:

```
C : Character;
X : aliased array (1..3) of Integer;  -- 12 bytes
Y : aliased String (1..16);           -- 16 bytes
for Y'Address use X'Address;          -- Overlay the two objects
...
Y (13) := 'a'; -- Assigns to memory beyond X
C := Y (14);   -- References memory beyond X
```

Both attempted assignments will be reported as errors by AddressSanitizer, since they are assigning to or from a memory location outside one of the overlaid objects.

Double-free error

If the same object is referenced by different pointers (access values) and is freed (via Unchecked_Deallocation) through one of the pointers, a subsequent attempt to free the object through the second pointer is erroneous. Such a "double free" is reported by AddressSanitizer:

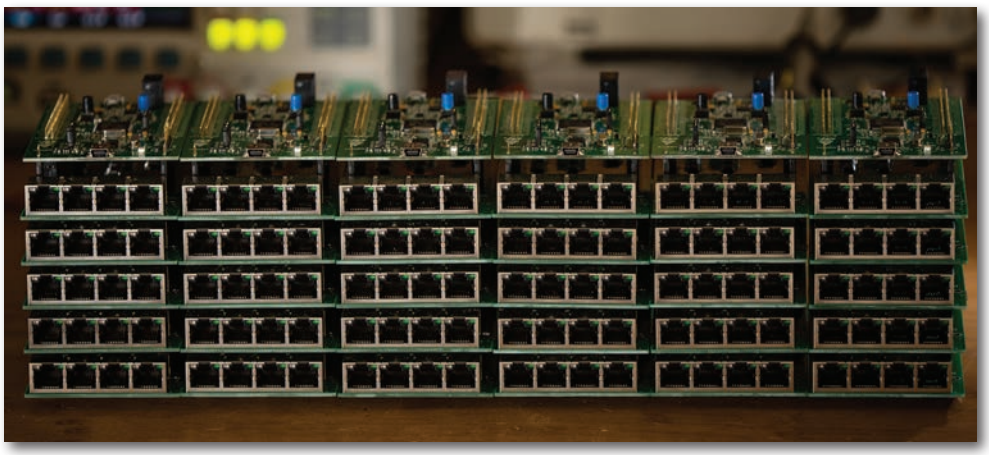
```
type Int_Access is access all Integer;
Ptr1, Ptr2 : Int_Access;

procedure Free is new
  Ada.Unchecked_Deallocation (Integer, Int_Access);
...
Ptr1 := new Integer'(100);
Ptr2 := Ptr1;
...
Free (Ptr1);  -- OK
Free (Ptr2);  -- Error: double free
```

AddressSanitizer complements other GNAT utilities for dynamic analysis such as gnatmem, which monitors a program's dynamic allocation and deallocation activity and displays information about incorrect deallocations and possible sources of memory leaks.

AddressSanitizer is supported under GNAT Pro 7.4 and later versions, on native x86 Linux platforms (32- and 64-bit).

For additional information please visit **github.com/google/sanitizers/wiki/AddressSanitizer/**.



Right: Hardware configuration for Real-Time and Embedded Systems course exercises Photo: Uwe Zimmer

GNAT Pro support for FACE

The Future Airborne Capability Environment (FACE™) Technical Standard is a government and industry open avionics standard for making military computing operations more robust, interoperable, and portable through a common avionics environment. Through active participation in the FACE consortium over the past five years and ongoing internal analysis and testing, AdaCore has been ensuring that GNAT Pro Ada meets the FACE Technical Standard requirements for the Ada Language Runtime Safety and Security profiles with multiple operating systems. Formal FACE Verification and Certification is currently being planned in coordination with operating system vendors.

AdaCore in O'PAVES Autonomous Vehicle Project

AdaCore is participating in the O'PAVES project (Open Platform for Autonomous VEhicle Systems), an effort funded by the European Union through the CPSE Labs consortium. The overall project goal is to effect a technology transfer to bring high reliability to the emerging domain of autonomous vehicle systems: in particular, to reuse or adapt technologies that have proved successful in high-assurance application areas such as aerospace and rail. O'PAVES is using AdaCore's tools to produce an open platform for developing and verifying autonomous vehicle algorithms, oriented towards students and researchers. The project got underway in November 2016 and will complete in October 2017.

SIGAda HILT Workshop Summary

ACM SIGAda's HILT 2016 (High Integrity Language Technology) workshop was held on October 6 and 7 in Pittsburgh, with a focus on Model-Based Development and Contract-Based Programming. Increasing the rigor in model-based development was a common thread across many of the talks. Including formalized contracts (such as those found in Ada 2012) in the generated code can help developers verify that the final application satisfies the original system requirements. AdaCore was a Gold Sponsor of HILT 2016, and Tucker Taft served as workshop co-chair.

NIST Report on Software Vulnerabilities

A November 2016 report from the National Institute of Standards and Technology in the US cited several technical approaches that offer high potential in helping to meet the goal stated in the title of the report: *Dramatically Reducing Software Vulnerabilities*. The approaches include formal methods, contract-based programming, static analysis, and model-based development, all of which are directly supported by AdaCore's major product lines (SPARK Pro, GNAT Pro, CodePeer, and QGen, respectively). The report is available at **nvlpubs.nist.gov/nistpubs/ir/2016/NIST.IR.8151.pdf**.

SPARK Webinar

On December 12, Yannick Moy (SPARK Product Manager at AdaCore) and Rod Chapman (Director, Protean Code Ltd) conducted a webinar "Building High-Assurance Software without Breaking the Bank". This webinar presented the SPARK language and toolset technology and showed how it meets today's software development and verification requirements. It also summarized SPARK experience from a number of government and commercial projects, and explained how an organization can incorporate SPARK into an existing infrastructure as a practical complement to testing. A recording of the webinar is available at **adaco.re/fx**, and information on SPARK may be found at **www.spark-2014.org/.**

calendar highlights / January–June 2017

For up-to-date information on conferences where AdaCore is participating, please visit **www.adacore.com/events/**.

VMCAI 2017 (18th International Conference on Verification, Model Checking, and Abstract Interpretation)
January 15–17, 2017 / Paris, France

AdaCore is a sponsor of this event.
conf.researchr.org/home/VMCAI-2017

SSS '17 (Safety-critical Systems Symposium)
February 7–9, 2017 / Bristol, UK

AdaCore is an exhibitor at this event.
scsc.org.uk/e438

Embedded World 2017
March 14–16, 2017 / Nuremberg, Germany

This conference is a major international exposition for embedded systems, encompassing topics such as security for electronic systems, distributed intelligence, the Internet of Things, e-mobility, and energy efficiency. AdaCore is an exhibitor at this event.
www.embedded-world.de/en

Certification Together International Conference 2017
March 21–23, 2017 / Toulouse, France

AdaCore is an exhibitor at this event.
www.certification-together.com

Tech Days 2017 Australia
March 21, 2017 / Canberra, Australia
March 24, 2017 / Adelaide, Australia

AdaCore is an exhibitor at these events. (Note: these are not related to AdaCore's Paris and Boston Tech Days conferences.)
dedicatedsystems.com.au/techdays/

Public Ada Training
April 24–28, 2017 / Yeovil, UK

AdaCore is conducting an Ada Fundamentals course, which combines live lectures with hands-on workshop exercises. For further information please refer to a companion article in this newsletter or visit the website:
www.adacore.com/public-ada-training/

TU-Automotive Detroit 2017
June 7–8, 2017 / Novi MI, USA

This major conference and exhibition is dedicated to automotive technological innovation. AdaCore is a sponsor and an exhibitor.
www.tu-auto.com/detroit/

Ada-Europe 2017
June 12–16, 2017 / Vienna, Austria

AdaCore is a sponsor and an exhibitor at this event.
www.auto.tuwien.ac.at/~blieb/AE2017/

Inside AdaCore is published twice a year simultaneously in New York and Paris by AdaCore.

150 W. 30th Street, 16th floor
New York, NY 10001, USA
tel +1 212 620 7300
fax +1 212 807 0162

46 rue d'Amsterdam
75009 Paris, France
tel +33 1 49 70 67 16
fax +33 1 49 70 05 52

info@adacore.com
www.adacore.com

AdaCore