Embedded programming in Ada

Fabien Chouteau, Yannick Moy
AdaCore

High level goals

Designed for Safety & Reliability

Designed for large-scale applications to embedded systems

Designed for being as much as possible right from the first time

Programming is about communication

With:

- The compiler
- The other tools (static analyzers, provers, etc.)
- Users of your API
- Your colleagues
- Your future self...

Strong typing

Consider the following

```
void set_throttle(float percent);
```

So full power is what: 100.0 or 1.0?

Strong typing

In Ada, you'd write

```
type Percentage is new Float range 0.0 .. 1.0;
procedure Set_Throttle (Value : Percentage);
```

```
Set_Throttle (50.0);
```

produces a compiler error
or a runtime error

Contracts

It's all about specifying what's done

```
procedure Inc (Value  : in out Percentage;
               Amount :  Percentage)
with Post => Value = (if Percentage'Last > Value'Old + Amount
                      then Value'Old + Amount
                      else Percentage'Last);
```

… more to come in the next part of the presentation

Multitasking: the Ravenscar profile[1]

Ceiling locking, with a FIFO within priorities
Periodic tasks, timed events
Mutual exclusion, shared access
Synchronization
Interrupt handling
Multi-core support

[1] https://blog.adacore.com/theres-a-mini-rtos-in-my-language

Representation clause

Used to represent precisely a memory mapped object
... such as a register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ITSF	RECALPF
														rc_w0	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAMP3F	TAMP2F	TAMP1F	TSOVF	TSF	WUTF	ALRBF	ALRAF	INIT	INITF	RSF	INITS	SHPF	WUTWF	ALRB WF	ALRAWF
rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rw	r	rc_w0	r	r	r	r	r

Bits 31:18 Reserved, must be kept at reset value

Bit 17 **ITSF**: Internal tTime-stamp flag

This flag is set by hardware when a time-stamp on the internal event occurs.
This flag is cleared by software by writing 0, and must be cleared together with TSF bit by writing 0 in both bits.

Bit 16 **RECALPF**: Recalibration pending Flag

The RECALPF status flag is automatically set to '1' when software writes to the RTC_CALR register, indicating that the RTC_CALR register is blocked. When the new calibration settings

Representation clauses

```
type F_Type is (Off, Third, Two_Third, Full)
   with Size => 2;

type Reg is record
  …
  F : F_Type;
  …
end record
  with Size => 32, Volatile_Full_Access;

for Reg use record
  …
   F at 0 range 3 .. 4;
  …
end record;
```

Representation clauses

Allows you to replace

```
tmpreg = Periph->Reg;
tmpreg = (tmpreg & ~0x18) | (Value << 3);
Periph->Reg = tmpreg;
```

by

```
Periph.Reg.F := Value;
```

Embedded program proving in SPARK

Yannick Moy, AdaCore

Some embedded software should never crash or hang or be hacked... or someone dies.

Building Perfect™ Software

KISS - Keep it simple, stupid

Certification processes (e.g. avionics) - re-re-reverifying

Use better programming languages and tools

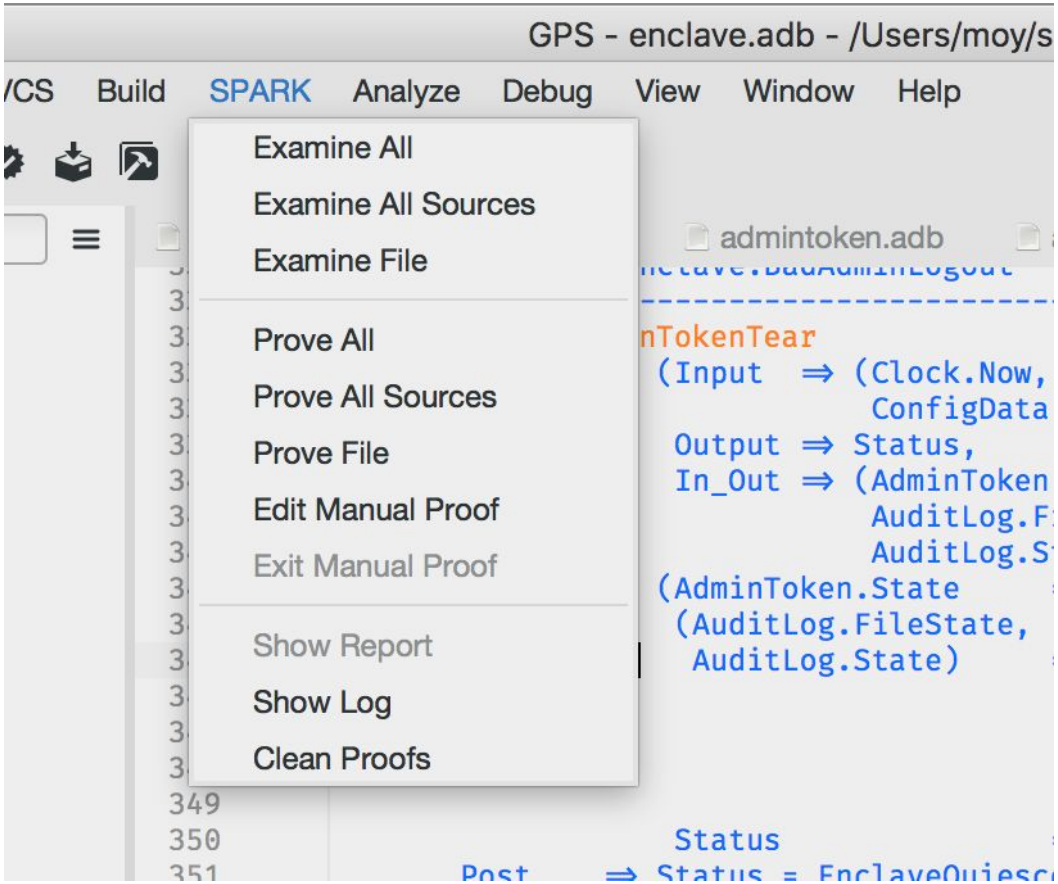
SPARK = Ada + proof

Support all Ada (OO, concurrency) except pointers (in progress)

Proof - mathematical guarantee

Made usable for (embedded) developers

Proof - the developer view



Proof - under the hood

An example of proof

```
procedure Increment (X : in out Integer)
  with Global  => null,
       Depends => (X => X),
       Pre     => X < Integer'Last,
       Post    => X = X'Old + 1;
```

data dependencies ✓
flow dependencies ✓
functionality ✓

```
procedure Increment (X : in out Integer)
is
begin
   X := X + 1;
end Increment;
```

robustness ✓

Examples of open source projects in SPARK

EwoK - secure microkernel for USB

"Software classes of attacks (e.g. buffer overflows) are mitigated using EwoK [...] providing more confidence by using the Ada safe language along with SPARK for formal verification of critical parts."

https://github.com/wookey-project/ewok-kernel

Muen - secure separation kernel

"The Muen Separation Kernel is the world's first Open Source microkernel that has been formally proven to contain no runtime errors at the source code level."

https://muen.codelabs.ch/

Beyond absence of runtime errors

Data invariants

From Muen project

```
type Table_Pointer_Type is range 0 .. 2 ** 35 - 1
  with Dynamic_Predicate =>
    Table_Pointer_Type mod MC.Page_Size = 0;

type Legacy_IRQ_Range is range 0 .. 23
  with Static_Predicate => Legacy_IRQ_Range /= 2;
```

Defensive Coding

From project github.com/Componolit/libsparkcrypto

```
function SHA512_Hash
  (Message : Message_Type;
   Length  : Message_Index) return SHA512_Hash_Type
with
   Pre =>
     Message'First <= Message'Last and
     Length / Block_Size +
      (if Length mod Block_Size = 0 then 0 else 1)
     <= Message'Length;
```

Correct API usage

From Muen project

```
procedure Clear_State (Id : Skp.Subject_Id_Type)
  with Refined_Post => Descriptors (Id) = SK.Null_Subject_State;

procedure Restore_State
 (Id   :   Skp.Subject_Id_Type;
  Regs : out SK.CPU_Registers_Type)
  with Refined_Post => Descriptors (Id).Regs = Regs;
```

Functional correctness

From project github.com/jcdubois/mσth/tree/spark

```
procedure os_sched_wait (task_id      : out os_task_id_param_t;
                         waiting_mask :     os_mbx_mask_t)
   with
      Pre => os_ghost_task_list_is_well_formed and
             os_ghost_mbx_are_well_formed and
             os_ghost_current_task_is_ready,
      Post => os_ghost_task_list_is_well_formed and
              os_ghost_task_is_ready (task_id);
```

[Continued…]

```
function os_ghost_task_list_is_well_formed return Boolean is
--  The mbx fifo of all tasks need to be well formed.
    (
    --  The list might be empty. This is legal.
    (os_sched_get_current_list_head = OS_TASK_ID_NONE and
     -- then all element are disconnected (not in a list)
     (for all task_id in os_task_list_rw'Range =>
        -- no next
        os_task_list_rw (task_id).next = OS_TASK_ID_NONE
        -- no prev
        and os_task_list_rw (task_id).prev = OS_TASK_ID_NONE
        -- and all tasks are in not ready state
        and not (os_ghost_task_is_ready (task_id))
     ...
```

Want to learn Ada or SPARK?

BETA

LEARN.
ADACORE.COM

Search docs

About

Courses

Books

Edit on GitHub

Learn.adacore.com is an interactive learning platform designed to teach the Ada and SPARK programming languages.

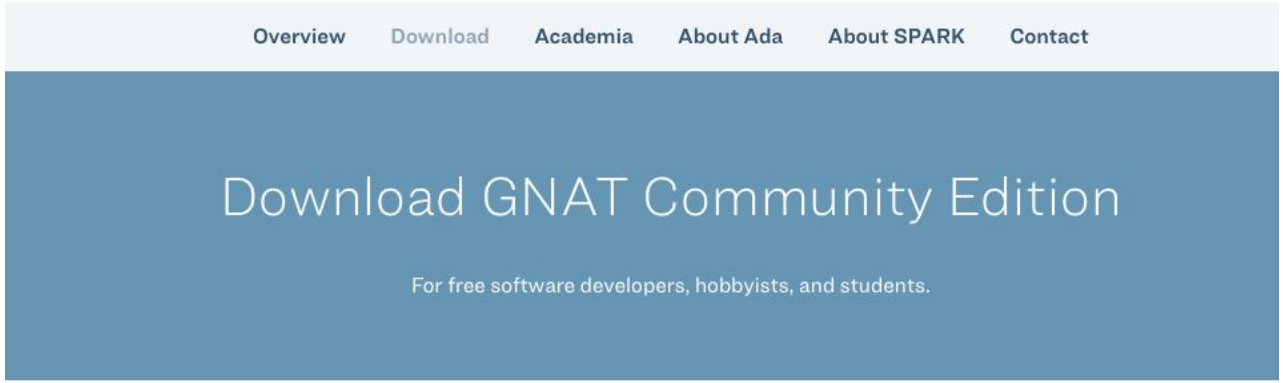
```
with Ada.Text_IO; use Ada.Text_IO;

procedure Learn is

   subtype Alphabet is Character range 'A' .. 'Z';

begin

   Put_Line ("Learning Ada from " & Alphabet'First & " to " & Alphabet'Last);

end Learn;
```

Reset Prove Run

```
Running...
  Learning Ada from A to Z
Success!
```

Want to try Ada or SPARK?

https://www.adacore.com/community



Supported boards

- STM32 Discovery boards
  - STM32F411E-disco
  - STM32F429I-disco
  - STM32F469I-disco
  - STM32F746G-disco
  - STM32F769I-disco
- Raspberry Pi2 (Bare metal)
- Micro:bit
- TI TMS570
- HiFive1 (RiscV)
- … and more[1]

[1] https://github.com/adacore/bb-runtimes