# SweetAda: a Multi-architecture Embedded Development 01000 Framework 0110 A showcase of the power of Ada, anywhere you may dream

0001011010001101

Gabriele Galeotti & Fernando Oleo Blanco

2.5ms

3.0ms

**3.5**ms

2.0ms

5ms

# Objectives

SweetAda brings a lot to the table, we will showcase the following topics

- Present SweetAda
  - What it is
  - What it supports
  - How to work with it
- Showcase SweetAda
  - SweetAda on an FPGA
  - SweetAda on...

It started from a firmware for a small M68k embedded board...

It started from a firmware for a small M68k embedded board...

An idea... run Ada everywhere, when the CPU starts

It started from a firmware for a small M68k embedded board...

An idea... run Ada everywhere, when the CPU starts

It started from a firmware for a small M68k embedded board...

An idea... run Ada everywhere, when the CPU starts

#### Issues started to appear

- Toolchain?
- RTS (Run Time System)?
- Build system/framework?

It started from a firmware for a small M68k embedded board...

An idea... run Ada everywhere, when the CPU starts

### Issues started to appear

- Toolchain?
- RTS (Run Time System)?
- Build system/framework?

## Goal

It should be capable of running even on an S/390 mainframe! Emulated at least

# Soooo... Did SweetAda deliver?

Oh yes, it does deliver

## Currently supported architectures

~\$ ls SweetAda/cpus

AArch64 ARM AVR M68k MicroBlaze MIPS NiosII OpenRISC PowerPC RISC-V SPARC SuperH System390 x86 x86-64

# Soooo... Did SweetAda deliver?

Oh yes, it does deliver

## Currently supported architectures

~\$ ls SweetAda/cpus AArch64 ARM AVR M68k MicroBlaze MIPS NiosII OpenRISC PowerPC RISC-V SPARC SuperH System390 x86 x86-64

## Currently supported targets

~\$ ls SweetAda/platforms							
Altera10M50GHRD	Dreamcast	MemecFX12	PC-x86	QEMU-R2D-PLUS	Spartan3A-EK		
Amiga-FS-UAE	FRDM-KL46Z	ML605	PC-x86-64	QEMU-RISC-V	Spartan3E-SK		
Android	GEMI	MPC8306-SOM	PK-S5D9	QEMU-STM32VLDISCOVERY	STM32F769I		
ArduinoUNO	HiFive1	MPC8306-Switch	QEMU-AArch64	Quadra800	System390		
Atlas	IntegratorCP	MPC8315e	QEMU-AVR	RaspberryPi3	Template		
DE10-Lite	LEON3	MSP432P401R	QEMU-MIPS	REF405EP	VMIPS		
DECstation5000.133	M5235BCC	MVME162-510A	QEMU-OpenRISC	SBC5206	XilinxZynqA9		
DigiConnectME	Malta	NEORV32	QEMU-PPC64	SPARCstation5	ZOOM		

#### Some even have subplatforms (different implementations)

~\$ ls SweetAda/platforms/NEORV32
platform-DE10-Lite platform-GHDL platform-ULX3S-Litex

# A few extra goodies

## Build your own toolchain

~\$ ls SweetAda/toolchains
Binutils.sh GCC.sh GDB.sh GNATTOOLS.sh linux-python-config.sh MANUAL.txt

Scripts to build GCC/GNAT for all of these targets, though it is not fully automated

# A few extra goodies

## Build your own toolchain

~\$ ls SweetAda/toolchains
Binutils.sh GCC.sh GDB.sh GNATTOOLS.sh linux-python-config.sh MANUAL.txt

Scripts to build GCC/GNAT for all of these targets, though it is not fully automated

## lt's own small C-lib

~\$ ls SweetAda/clibrary
ada\_interface.h clibrary.gpr ctype.c c\_wrappers.ads gnat.adc stdio.h string.c strings.h
assert.c clibrary.h ctype.h errno.c Makefile stdlib.c string.h
assert.h configuration.in c\_wrappers.adb errno.h stdio.c stdlib.h strings.c

# A few extra goodies

## Build your own toolchain

~\$ ls SweetAda/toolchains
Binutils.sh GCC.sh GDB.sh GNATTOOLS.sh linux-python-config.sh MANUAL.txt

Scripts to build GCC/GNAT for all of these targets, though it is not fully automated

## lt's own small C-lib

~\$ ls SweetAda/clibrary								
ada_interface.h	clibrary.gpr	ctype.c	c_wrappers.ads	gnat.adc	stdio.h	string.c	strings.h	
assert.c	clibrary.h	ctype.h	errno.c	Makefile	stdlib.c	string.h		
assert.h	configuration.in	c_wrappers.adb	errno.h	stdio.c	stdlib.h	strings.c		

## A few drivers...

~\$ ls SweetAda/dr	ivers					
am7990.adb	etherlinkiii.ads	ide.ads	pc.adb	piix.adb	uart16×50.adb	xps.ads
am7990.ads	ethernet.adb	Makefile	pc.ads	piix.ads	uart16×50.ads	z8530.adb
blockdevices.adb	ethernet.ads	mc146818a.adb	pci.adb	pl011.adb	upd4991a.adb	z8530.ads
blockdevices.ads	goldfish.adb	mc146818a.ads	pci.ads	pl011.ads	upd4991a.ads	
configuration.in	goldfish.ads	ne2000.adb	pcican.adb	pl110.adb	vga.adb	
etherlinkiii.adb	ide.adb	ne2000.ads	pcican.ads	pl110.ads	vga.ads	

How to make all this possible

## GCC/GNAT compiler, the driving force

Needs no introduction nor explanation. GCC with all the architectures it supports, is the central pillar

How to make all this possible

## GCC/GNAT compiler, the driving force

Needs no introduction nor explanation. GCC with all the architectures it supports, is the central pillar

## Good old make

Used as the skeleton for the build. It selects targets, runtimes, toolchains, flags, autogenerates/processes files, prints diagnostics...

How to make all this possible

## GCC/GNAT compiler, the driving force

Needs no introduction nor explanation. GCC with all the architectures it supports, is the central pillar

## Good old make

Used as the skeleton for the build. It selects targets, runtimes, toolchains, flags, autogenerates/processes files, prints diagnostics...

## GPRBuild, the Ada heavyweight builder

Runs the build for the final binary/kernel

How to make all this possible

## GCC/GNAT compiler, the driving force

Needs no introduction nor explanation. GCC with all the architectures it supports, is the central pillar

## Good old make

Used as the skeleton for the build. It selects targets, runtimes, toolchains, flags, autogenerates/processes files, prints diagnostics...

## GPRBuild, the Ada heavyweight builder

Runs the build for the final binary/kernel

## And some configuration files

Each platform works a bit differently. A config file is present to help with the build config, post-build and upload. Python, shell and some legacy TCL is also used

SweetAda setup

# A tipical workflow, step by step, using SweetAda I SweetAda setup

## First things first, get SweetAda & find help

git clone https://github.com/gabriele-galeotti/SweetAda & cd SweetAda make help

## SweetAda setup

## First things first, get SweetAda & find help

## Get a toolchain

- Either provide one / your own (SweetAda is that flexible)
- Or build one using script found in ~\$ ls SweetAda/toolchains

## SweetAda setup

## First things first, get SweetAda & find help

## Get a toolchain

- Either provide one / your own (SweetAda is that flexible)
- Or build one using script found in ~\$ ls SweetAda/toolchains

## You are ready to rock and roll

A tipical workflow, step by step, using SweetAda II CPU & Board setup

# A tipical workflow, step by step, using SweetAda II CPU & Board setup Build the RTS (Run Time System)

SweetAda supports both the zfp (zero-footprint profile) & sfp (small-footprint)

make CPU=RISC-V RTS=sfp rts # Or RTS=zfp
# make CPU=RISC-V RTS=sfp TOOLCHAIN\_NAME=riscv64-elf rts

# A tipical workflow, step by step, using SweetAda II CPU & Board setup Build the RTS (Run Time System)

SweetAda supports both the zfp (zero-footprint profile) & sfp (small-footprint)

make CPU=RISC-V RTS=sfp rts # Or RTS=zfp
# make CPU=RISC-V RTS=sfp TOOLCHAIN\_NAME=riscv64-elf rts

## Select a board and files for the build process

This will select the boards specific files and configuration for the build

make PLATFORM=NEORV32 SUBPLATFORM=ULX3S-Litex createkernelcfg

# A tipical workflow, step by step, using SweetAda II CPU & Board setup

Build the RTS (Run Time System)

SweetAda supports both the zfp (zero-footprint profile) & sfp (small-footprint)

make CPU=RISC-V RTS=sfp rts # Or RTS=zfp
# make CPU=RISC-V RTS=sfp TOOLCHAIN\_NAME=riscv64-elf rts

## Select a board and files for the build process

This will select the boards specific files and configuration for the build

make PLATFORM=NEORV32 SUBPLATFORM=ULX3S-Litex createkernelcfg

## Configuration of the build (flags, post-process steps, application to be built...)

make TOOLCHAIN\_NAME=riscv64-elf PLATFORM=NEORV32 SUBPLATFORM=ULX3S-Litex configure

## Modify the default configuration as you see fit

[...] TOOLCHAIN PREFIX: /opt/toolchains TOOL CHAIN NAME: riscv64-elf [...] rv32im/ilp32 GCC MULTIDIR: RTS: sfp GNAT.ADC PROFILE: sfp ADA MODE: ADA22 [...] USE | TBADA: Υ USE CLIBRARY: **OPTIMIZATION | EVEL:** 2 ADA GCC SWITCHES (RTS): -mcmodel=medany C GCC SWITCHES (RTS): -mcmodel=medany GCC SWITCHES (PLATFORM): -march=rv32i2p0 mc -mabi=ilp32 LOWLEVEL FILES: startup.S llkernel.S GCC SWITCHES (LOWLEVEL): linker.lds ID SCRIPT: LD SWITCHES: -m elf32lriscv --defsym riscv mtime mmap=0×FFFF400 --defsym riscv mtimecmp mmap=0×FFFF408 **OBJCOPY SWITCHES: OBJDUMP SWITCHES:** 

## Build the kernel, application into a binary

GPRBuild takes over the build

make all # or make kernel
make postbuild # run helper steps if needed to get the binary ready

## Build the kernel, application into a binary

GPRBuild takes over the build

make all # or make kernel
make postbuild # run helper steps if needed to get the binary ready

## Run the application

Each target has its own way of running SweetAda (launch emulator, upload via serial, JTAG...)

make run
# or
make session-start # make session-end

## Build the kernel, application into a binary

GPRBuild takes over the build

make all # or make kernel
make postbuild # run helper steps if needed to get the binary ready

## Run the application

Each target has its own way of running SweetAda (launch emulator, upload via serial, JTAG...)

make run
# or
make session-start # make session-end

Enjoy!

# **DEMOnstration time**

SweetAda running on...

- ULX3S FPGA, NEORV32 softcore, Litex build
  - RISC-V 32-bit IMC core, 50 MHz, Timer, UART, Leds
- FRDM-KL46Z ARM board
  - LEDS, GDB-debugging session
- PC-X86, QEMU-ROM
  - VGA, TCP-IP stack

SweetAda offers a lot to the Ada (and wider) programming community

Portability - run Ada code on a large number of architectures, boards and emulators!

- Portability run Ada code on a large number of architectures, boards and emulators!
- Developer friendly & Hackable show expand, tweak, improve and automate things as your needs evolve!

- Portability run Ada code on a large number of architectures, boards and emulators!
- Developer friendly & Hackable show expand, tweak, improve and automate things as your needs evolve!
- Fast & easy work prepare, build and upload applications with just a few commands!

- Portability run Ada code on a large number of architectures, boards and emulators!
- Developer friendly & Hackable show expand, tweak, improve and automate things as your needs evolve!
- Fast & easy work prepare, build and upload applications with just a few commands!
- SFull support supports and builds several runtimes, with exceptions, interrupts, and much more!

- Portability run Ada code on a large number of architectures, boards and emulators!
- Developer friendly & Hackable show expand, tweak, improve and automate things as your needs evolve!
- Fast & easy work prepare, build and upload applications with just a few commands!
- SFull support supports and builds several runtimes, with exceptions, interrupts, and much more!
- *A Open development* has had tremendous growth and contributions are more than welcome!

- Portability run Ada code on a large number of architectures, boards and emulators!
- Developer friendly & Hackable show expand, tweak, improve and automate things as your needs evolve!
- Fast & easy work prepare, build and upload applications with just a few commands!
- **Solution** Full support supports and builds several runtimes, with exceptions, interrupts, and much more!
- *A* Open development has had tremendous growth and contributions are more than welcome!
- MIT licensed hack, develop, integrate it however you like thanks to its liberal license!

SweetAda offers a lot to the Ada (and wider) programming community

- Portability run Ada code on a large number of architectures, boards and emulators!
- Developer friendly & Hackable show expand, tweak, improve and automate things as your needs evolve!
- Fast & easy work prepare, build and upload applications with just a few commands!
- **Solution** Full support supports and builds several runtimes, with exceptions, interrupts, and much more!
- *A Open development* has had tremendous growth and contributions are more than welcome!
- MIT licensed hack, develop, integrate it however you like thanks to its liberal license!

### Future work?

Improve the RTS, more drivers, quality-of-life improvements, documentation, board support...

# Thank you very much!

000000

0

SweetAda, vø. 1 Gabriele Galeotti & Fernando Oleo Blanco

> gabriele.galeotti@sweetada.org 100000000000

- IIII