I3DS SENSOR SUITE FOR SPACE ROBOTICS

Kristoffer Nyborg Gregersen, PhD
Ada-Europe 2018, Lisbon
About me

• SINTEF Digital – Department of Mathematics and Cybernetics
• Research manager for automation and real-time systems
• PhD in cybernetics from NTNU on real-time system support in Ada with execution time for interrupts
• We are hiring research scientist!
SINTEF is one of Europe’s largest independent research organisations

- **2000 Employees**
- **75 Nationalities**
- **4000 Customers**
- **NOK 3.1 billion Revenues**
- **NOK 450 MILL International sales**
I3DS project

• Space Robotics Cluster (H2020)
  • Future robotics platform for ESA
  • Led by the PERASPERA project
  • 6 Operational Grants (OG's)

• I3DS – Integrated 3D Sensors
  • Thales Alenia Space is coordinator
  • SINTEF leads software development, integration and interface definition
Project status

• Project phases defined as for ESA projects
• Kick-off in November 2016, finish November 2018
  • SRR in February 2017
  • PDR in July 2017
  • CDR in February 2018
• Now finishing integration phase of project lead by SINTEF
• Validation of sensor suite in demonstrators from July 2018
• Full integration with other OG's in next SRC calls, proposal submitted!
Project motivation and goal

• Develop and demonstrate a modular sensor suite for space robotics
  • Tight cooperation with OG's for middleware, autonomy and sensor fusion
  • TASTE framework with AADL and ASN.1 messages for system integration
  • Sensors, Instrument Control Unit (ICU) and software at TRL5

• Motivation: Reduce development time of space missions
  • Abstract away device specific details with standard interface for sensor class
  • Allows to use latest sensors available without changing other software
  • Reuse sensor interfaces, processing components and ICU hardware
Hi-Res Camera GBit
TIR Camera Gbit
ToF Camera GBit
COTS Hi-Res Camera GBit
WA Illumination Trigger & RS232
Pattern Projector Trigger

Stereo Camera GBit
Radar SpW
LIDAR Gbit & RS232
IMU RS485
Star Tracker SpW
Tactile & Torque LVDS
Demonstrator with orbital use-case

Medium-range Approach
> 200m
Inspection / Observation
~ 20m
Rendezvous
< 20m
Berthing / Docking
~ 3m
Servicing
0m

TAS-F in Cannes
Demonstrator for planetary use-case
Lab-bench at SINTEF

- Integration of sensors with ICU and testing of functionality and real-time properties
- Recording of coherent data from all sensors moving on trolley
- Sent for mechanical integration at PIAP this week (June 2018)
Software architecture

Device specific interfaces

Standard sensor class interfaces (ASN.1)

Sensor A
Sensor B
Sensor C

OG4
ICIU

OG1/2/3
OBC

Smart Sensor
Sensor interfaces

• Common ASN.1 commands and queries
  • State changes inactive ↔ standby ↔ operational
  • Set sample rate and sample batch size of sensor
  • Get configuration, state and temperature of sensor

• Each *sensor class* has its own message interface definitions
  • Camera, ToF camera, LIDAR, radar, star tracker, IMU, analogue sensors
  • Class specific commands, queries and measurements
  • Implemented by concrete sensor drivers and emulator

• Example: Camera class with shutter time, gain, flash, image frames...
ASN.1 compiler

- Use ASN1CC in project
- Same as used in TASTE framework
- Coded in F# and running on mono
- Outputs code in C and Ada/SPARK
- SPARK allows for formal verification
- Currently use C version due to issues in Ada implementation
- uPER encoding for standard messages
- Devices interfaces with ACN

https://github.com/ttsiodras/asn1scc
Computation and throughput load

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Sample size</th>
<th>Rate</th>
<th>Throughput*</th>
</tr>
</thead>
<tbody>
<tr>
<td>HR camera</td>
<td>2048<em>2048</em>2 Bytes = 8 MiB</td>
<td>10 Hz</td>
<td>84 Mb/s</td>
</tr>
<tr>
<td>Stereo camera</td>
<td>2<em>2048</em>2048*2 Bytes = 16 MiB</td>
<td>10 Hz</td>
<td>168 Mb/s</td>
</tr>
<tr>
<td>ToF camera</td>
<td>640<em>480</em>5 Bytes = 1500 KiB</td>
<td>10 Hz</td>
<td>15 Mb/s</td>
</tr>
</tbody>
</table>

*absolute best case

- Lens distortion correction
- Histogram equalization
- CLAHE
- Bilateral filtering
- Stereo rectification
- Point-cloud generation
Instrument Control Unit

- ICU build on the Xilinx Zynq UltraScale+ MPSoC
  - Mixed-criticality real-time system
  - Quad-core ARM Cortex A53 with Xilinx PetaLinux
  - Two ARM Cortex R5 real-time processors
  - FPGA for bespoke hardware modules
  - ARM Mail GPU for processing with support for OpenCV

- Runs sensors interfaces and pre-processing algorithms

- The Xilinx Zynq UltraScale+ is a complex platform!
Processing System

Application Processing Unit
- ARM® Cortex™-A53
- Floating Point Unit
- Memory Management Unit
- Embedded Trace Macrocell
- 32kB I-Cache w/ECC
- 32kB D-Cache w/ECC
- GIC-400
- SCU
- CIIF/SMU
- 1MB L2 w/ECC

Memory
- DDR4/3L
- LPDDR4/3
- 32/64 bit w/ECC
- 256kB OCM with ECC
- 64kB L2 Cache

Graphics Processing Unit
- ARM Mali™ 400 MP2
- Geometry Processor
- Pixel Processor
- Memory Management Unit

Real-Time Processing Unit
- ARM Cortex™-R5
- Vector Floating Point Unit
- Memory Protection Unit
- 128kB TCM w/ECC
- 32kB I-Cache w/ECC
- 32kB D-Cache w/ECC
- GIC

Platform Management Unit
- System Management
- Power Management
- Functional Safety

Configuration and Security Unit
- Config AES
- Decryption
- Authentication
- Security Boot
- Voltage/Temperature Monitor
- TrustZone

System Functions
- Multichannel DMA
- Timers, WDT, Resets, Clocking & Debug

High-Speed Connectivity
- DisplayPort v1.2a
- USB 3.0
- SATA 3.1
- PCIe® 1.0 / 2.0
- PS-GTR

General Connectivity
- GigE
- USB 2.0
- CAN
- UART
- SPI
- Quad SPI NOR
- NAND
- SD/eMMC

Programmable Logic

Storage & Signal Processing
- Block RAM
- UltraRAM
- DSP
- GTH
- GTY
- 100G EMAC
- PCIe Gen4

System Monitor
- General-Purpose I/O
- High-Performance HP I/O
- High-Density HD I/O

High-Speed Connectivity
- Interlaken
ICU mixed criticality real-time system

- Xilinx PetaLinux on Cortex A53
  - Interfaces to GigE vision cameras and COTS sensors
  - System interface to OG1/2/3 over GigaBit Ethernet
  - Use ZMQ library for high-performance messaging over TCP/IP
  - C++ framework developed for sensor interfaces, clients and emulators
  - OpenCV used for image processing, e.g. stereo image to point-cloud

- Embedded and real-time software on Cortex R5
  - ADC polling for tactile and F/T sensors, IMU, triggers, and SpW interfaces...
  - Ada/SPARK used for sensor control and processing

- Communicate in memory-buffers between cores using OpenAMP
Cortex A53 with Xilinx PetaLinux
Cortex R5 #1 with device interfaces
Cortex R5 #2 with SpaceWire interface

SpW link

SpW / RMAP device driver

Memory map

ARM Cortex-R / Core #2
Embedded

Message routing

Time service

OpenAMP

SpW driver

ARM Cortex-A
GNU/Linux
Ada and SPARK 2014 on the Cortex R5

- Use GNAT Pro Developer for ARM with SPARK 2014 tools
  - No Ravenscar run-time for the Zynq UltraScale+ MPSoC
  - Adapted zero footprint (ZFP) run-time from TMS570 (Cortex R4/5)
  - GNAT Pro makes it very easy to reconfigure and recompile run-times
- Develop static library with functionality in Ada/SPARK 2014
- Xilinx SDK for FreeRTOS, device interfaces, linking and programming
- Match ABI and compile flags for GNAT and Xilinx SDK (both use GCC)
Functionality made in SPARK 2014

• Interfaces for IMU and ADC sensors
  • Sensor state machines and command handling
  • Process, accumulate, and send sensor measurement data
  • IMU needs temperature-dependent correction with calibration data
  • ADC readings are converted to physical floating-point value
  • Hard real-time demands, sampling at 200 Hz and 1000 Hz

• Formal proof of correctness for sensor state machine

• Flow control and dependencies for data processing

• SPARK 2014 allows us to develop code with confidence!
Forward to a certifiable software system

• All critical functionality and Space Wire/Fibre on Cortex R5 and FPGA

• Want to have certified Ravenscar run-time on the Cortex R5
  • Need to integrate GNAT Pro with Xilinx SDK or the other way around? Reuse drivers?
  • Bonus: FPGA gives great opportunities for specialized support hardware (e.g. TMU)

• Mixed-criticality with the Cortex A53 and PetaLinux
  • Xilinx reVISION provides FPGA-accelerated OpenCV and more
  • PetaLinux with accelerated OpenCV for heavy camera pre-processing
  • Could use hypervisor such as XtratuM for improved isolation

• Coming GNAT for ARM64 allows use on PetaLinux too!
Conclusions

• Future robotics applications require high-performance computer platforms for AI, image processing, machine learning etc.

• The Xilinx UltraScale+ MPSoC with accelerated OpenCV and real-time functionality with Ravenscar and SPARK 2014 is very promising!

• Need good integration between Xilinx tools generating hardware, PetaLinux, hypervisor, and GNAT Pro for safety-critical real-time code

• We want to use this mixed-criticality real-time system and SPARK for our autonomous robots and high-performance edge computing!
This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 730118
Technology for a better society