Outlook and Invitation to Contribute

Embedded System Week 2020
Zain Hammadeh
Where are we using Tasking Framework currently?

- Tasking Framework
- ScOSA
- CALLISTO
- HAP
- COMPASSO
- MAIUS
- MMX
- RoFEx
- BECCAL
- GALILEO

DLR.de • Chart 2 > Tasking Framework: Outlook > Zain Hammadeh • Embedded Systems Week 2020 > 20.09.2020
Scalable On-board Computing for Space Avionics (ScOSA)

High computational demand

Commercial off-the shelf HPN

Reconfigurable Distributed System

RCN: Reliable Computer Node
HPN: High Performance Node
I/F Node: Interface Node
SpW: SpaceWire
RTU: Remote Terminal Unit
PCU: Power Converter Unit
TM/TC: Telemetry/Telecommand
ACS: Attitude Control System

* Other Satellite Bus Avionics

Höflinger – ASTRA 2017
ScOSA – Software stack

- Used hardware: FPGAs, ASICs, SoCs
- Multi-OS capable system
- FPGA as co-processor
- Reliable process communication network protocol (SpaceWire-IPC) and RMAP
- Supports Spacewire and Ethernet (EtherCat)
- Tasking framework as algorithm scheduler and communication manager
- System management and PUS services
- Application developer friendly API
ScOSA – Software stack

- Used hardware: FPGAs, ASICs, SoCs
- Multi-OS capable system
- FPGA as co-processor
- Reliable process communication network protocol (SpaceWire-IPC) and RMAP
- Supports Spacewire and Ethernet (EtherCat)
- Tasking framework as algorithm scheduler and communication manager
- System management and PUS services
- Application developer friendly API

Global Software Stack View
Reusability Flight Experiment (ReFEx)

Launch Phase
- Unlock Actuators & Re-Entry Preparation
- Re-Entry Segment Separation (SEP) & Beginning of Guided Control (BoGC)
- Firing Separation & Wing Deployment
- YoYo-Release
- Burn-out S30
- Ignition S30
- Burn-out S31 & Stage Separation
- Lift-Off

Experimental Phase
- Apogee ~ 130 km
- ~ 2 min
- ~ 87 km
- ~ 4 min
- ~ 10 min
- Entry Interface (EI) ~ 00 km
- ~ 7 min
- Touch Down
- Turning Flight & Roll Maneuver
- End of Experiment (Exit) ~ 10 - 8 km

Projects:
- 1.6 min
- 10 min

DLR.de • Chart 6  > Tasking Framework: Outlook > Zain Hammadeh • Embedded Systems Week 2020  > 20.09.2020

https://elib.dlr.de/131099/1/IAC-19.D2.6.9x49695.pdf
ReFEx – Software stack

Application Layer
- TM/TC
- Navigation
- MVM
- Guidance

Middleware Layer
- HAL
- Tasking Framework

Operating System Layer
- Clock
- IRQ
- Thread
- Uart
- Eth
- GPIO
- CAN
- BSP

Hardware Layer
- Bootloader
- BIOS
- X86 Processor Board
- Core 0
- Core 1
- Core 2
- Core 3
- DLR
- Chart 7
- Tasking Framework: Outlook
- Zain Hammadeh
- Embedded Systems Week 2020
- 20.09.2020
Outlook on Tasking Framework’s Roadmap

- C++11 Platform for Scheduler
- TaskProvider as a C++11 tuple class for better type safety
- Executor Steeling from other Scheduler
- More channel types than only FIFO channel
- User defined Boolean expression for InputArray to express more complex activation conditions
Potential research

- Timing analysis (WCET + WCRT)
- Multicore support (partitioning strategies)
- Support for GPU tasks
- Bare-metal platform
- Scheduling policy shortest laxity first
- Alternative execution model for high workload
Development process

Confidentiality

Consistency

Internally

Development

Release

Issues

Public
Development process

Confidentiality

Consistency

Internally

Public

Issues

- You can report bugs, suggest bug fixes, suggest features, etc.
- In case of contribution:
  - your contribution will be tested and evaluated
  - your name will be added to the contributor list (CONTRIBUTORS.md)
  - and to the related commit.
How to cite Tasking Framework?

- You can cite our paper
  - **OSPERT 2019**

**On-line**

- [https://elib.dlr.de/128249/](https://elib.dlr.de/128249/)

**Plain**


**Bibtex**

```
@InProceedings{Hammadeh2019,
    author    = {Zain A. H. Hammadeh and Tobias Franz and Olaf Maibaum and Andreas Gerndt and Daniel L{"u}dtke},
    booktitle = {15th annual workshop on Operating Systems Platforms for Embedded Real-Time applications},
    title     = {Event-Driven Multithreading Execution Platform for Real-Time On-Board Software Systems},
    year      = {2019},
    month     = {Juli},
    pages     = {29--34},
    url       = {https://elib.dlr.de/128249/},
}
```
Live Q&A session

Sunday 20.09.2020
at 11:00 – 12:00 am EDT
at 05:00-06:00 pm Berlin

Tutorial 4: Tasking Framework: An open-source software development library for on-board software systems

11:00am - 12:00pm, Sep 20

* This tutorial is a live Q&A session with pre-recorded videos made available before the event.

Tasking Framework is a C++ software development library and an event-driven multithreading execution platform. It is developed by the Institute for Software Technology, German Aerospace Center (DLR). Tasking Framework is dedicated to improve the reusability in developing embedded software systems and to reconcile the embedded software with model-driven software development. It can be used to develop, but not dedicated for, critical as well as non-critical embedded software on single-core as well as parallel architectures. Tasking Framework gives software developers the ability to implement their applications as task graphs with arbitrary activation patterns (periodic, aperiodic and sporadic) using a set of abstract classes with virtual methods. It is compatible with the POSIX-based operating systems, mainly Linux and RTEMS. The Tasking Framework was successfully used in, for instance, the attitude orbit