VSB TECHNICAL UNIVERSITY OF OSTRAVA

EUNIS - CZ

ČSSI

IT Cluster

PEOPLE FOR IT

IT for Practice 2022 Conference Program

Thursday, October 13th 2022

Safety standards for real-time control

Roman Danel

Department of Applied Economics

Faculty of Economics

VŠB – TU Ostrava

Outline

- Real Time control introduction
- Real-time operating systems (RTOS) overview
- Safety standards & SIL
- Reccomendation for RTOS
- Case Study

Real-time system

- Real Time
  - Soft
  - Hard
  - Critical Systém
- According internal control
  - RTOS controlled by **time** (polling) – Example: MARS
  - RTOS controlled by **events** (interrupts) – Example: QNX
- According the kernel architecture:
  - Kernel in the form of an extra-code library – Example: VxWorks, eCos
  - Standalone kernel – Example: QNX, OS/9

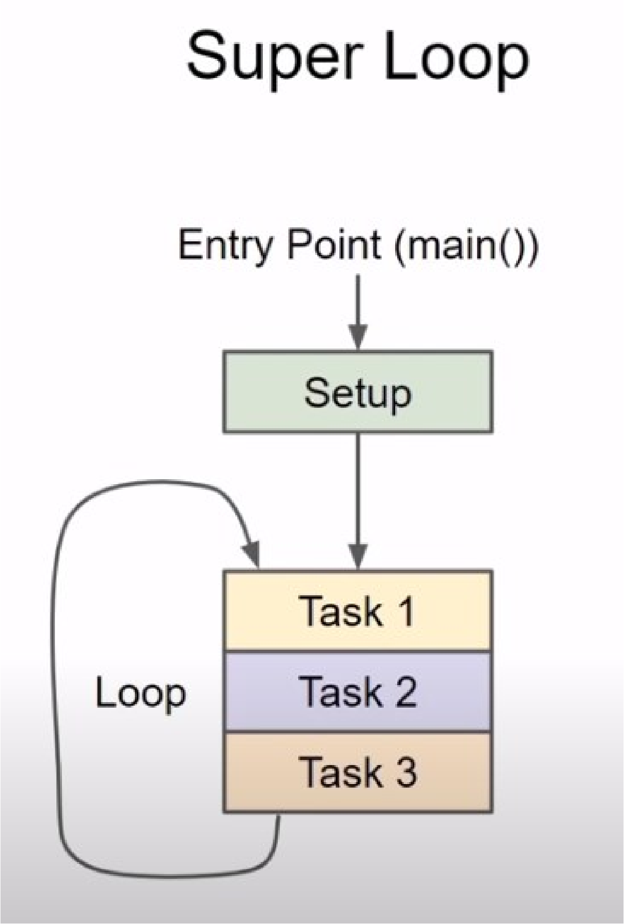
RTOS overview

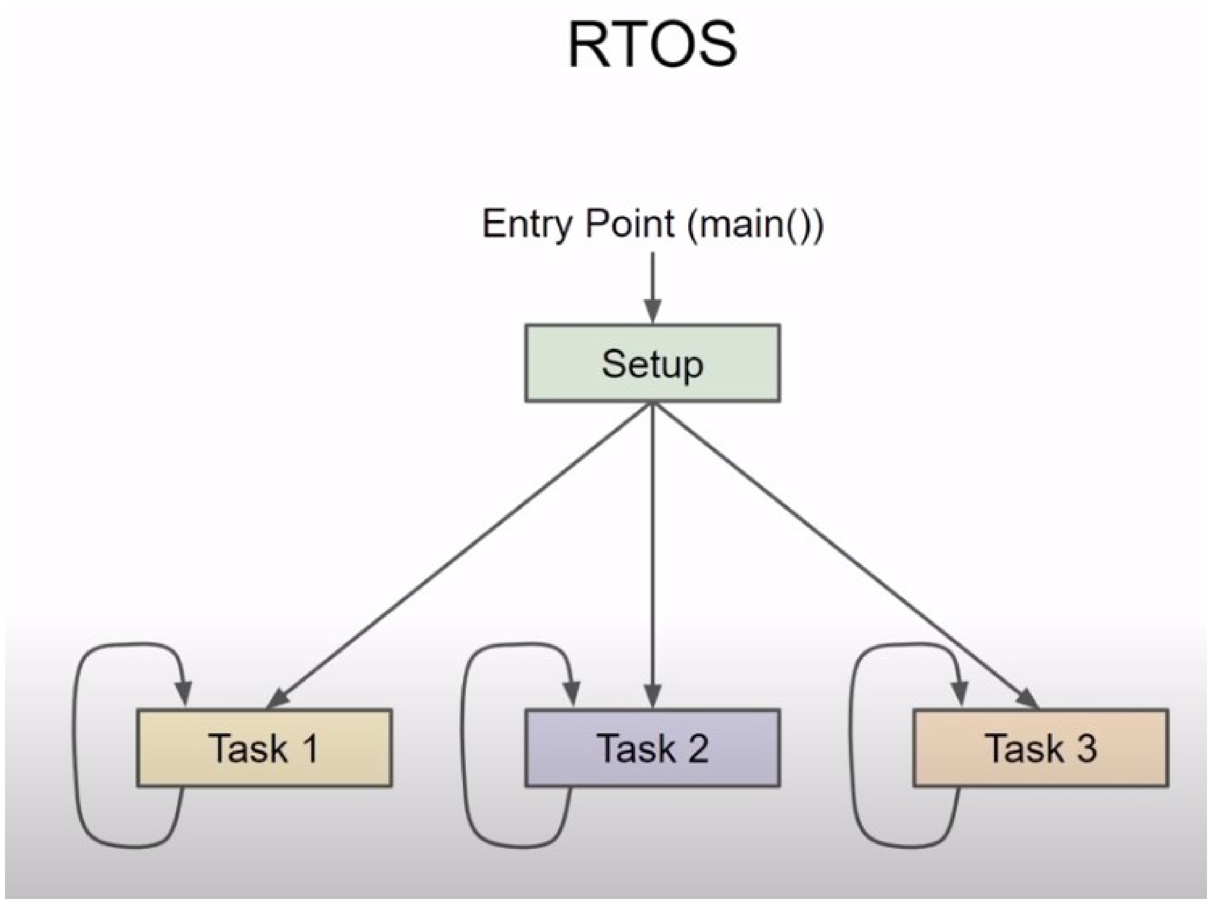
Operating system	Producer	Homepage
RT Linux		http://cs.uccs.edu/~cchow/pub/rtl/doc/html/GettingStarted/
RTAI	University of Milano, Italy	https://www.rtai.org/
ENEA OSE	ENEA AB	https://www.enea.com/products-services/operating-systems/enea-ose
LynxOS	Lynx Software Technologies	http://www.lynx.com
VxWorks	Wind River Inc.	https://www.windriver.com/products/vxworks/
INTEGRITY	Green Hills software	https://www.ghs.com/products/rtos/integrity.html
RTX64	IntervalZero	https://www.intervalzero.com/en-products/en-rtx64/
Xenomai		www.xenomai.org

RTOS overview

Operating system	Producer	Homepage
Real-Time Linux	Linux Foundation	https://wiki.linuxfoundation.org/realtime/start
embOS	German company Segger	https://www.segger.com/products/rtos/embos/
Keil RTX	ARM	https://www.keil.com/arm/rl-arm/kernel.asp
QNX Neutrino	Canadian company QNX Software Systems	https://blackberry.qnx.com/en
Zephyr	Wind River Rocket	https://www.zephyrproject.org/
TI RTOS	Texas Instruments	https://www.ti.com/tool/TI-RTOS-MCU
FreeRTOS		https://www.freertos.org/
Mbed OS	ARM	https://www.arm.com/products/development-tools/embedded-and-software/mbed-os
PikeOS	SYSGO GmbH	https://www.sysgo.com/pikeos
Sciopta	SCIOPTA Germany	https://www.sciopta.com/safetykrn/index.html
Azure RTOS	Microsoft	https://azure.microsoft.com/en-us/products/rtos/

Approach for RT application





Safety Standards

IEC 61508	Safety standard for electrical and electronic devices – defines SIL (Safety Integrity Level); safety life cycle; criteria for software testing
ISO 26262	Standard for automotive; Road Vehicles – Functional Safety
IEC 62304	Medical devices
DO-178x	Aerospace; ‚Software Considerations in Airborne Systems and Equipment Certification’; 5 levels of ‚failure conditions‘; 5 levels of ‚design assurance‘ (A to E); DO-254 - Design Assurance Guidance for Airborne Electronic Hardware
IEC 61511	Safety instrumented systems for the process industry sector
IEC 61513	Nuclear industry
IEC 62061	Safety of machinery
IEC 60730	Safety Standard for Household Appliances
ISO 15408	Common Criteria for Information Technology Security Evaluation

Standard IEC 61508 -> SIL

SIL = Safety Integrity Level – functional safety standard

Categories of likelihood of occurrence

Category	Definition	Range (failures per year)
Frequent	Many times in lifetime	$> 10^{-3}$
Probable	Several times in lifetime	10^{-3} to 10^{-4}
Occasional	Once in lifetime	10^{-4} to 10^{-5}
Remote	Unlikely in lifetime	10^{-5} to 10^{-6}
Improbable	Very unlikely to occur	10^{-6} to 10^{-7}
Incredible	Cannot believe that it could occur	$< 10^{-7}$

Consequence categories

Category	Definition
Catastrophic	Multiple loss of life
Critical	Loss of a single life
Marginal	Major injuries to one or more persons
Negligible	Minor injuries at worst

The standard defines four safety levels:

- **SIL 1**
- **SIL 2** – requirement for system self-diagnosis, documentation
- **SIL 3** – requirement for the system to operate in the event of failure of individual devices - redundancy
- **SIL 4** – triple redundancy

Methods for SIL setting:

- Risk matrices
- Risk graphs
- Layers of protection analysis (LOPA)

Problems of RTOS

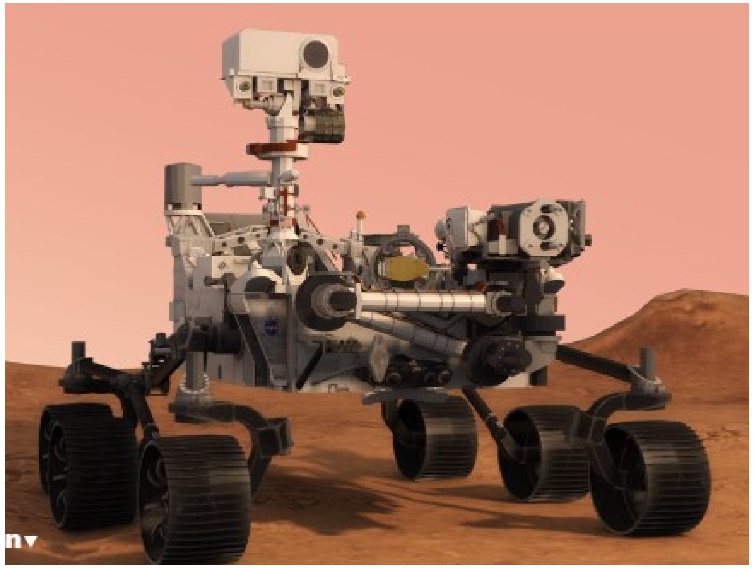
- Unexpecting functional behaviour of tasks
    - Accidental infinite loops, deadlocks, misuse of pointers…
- Application software
    - Unsafe use of RAM memory
        - Loss of memory
        - Multiple copying
        - Memory exhaustion
    - Corruption of processor data
    - Task execution and interactions
- Malfunctioning hardware

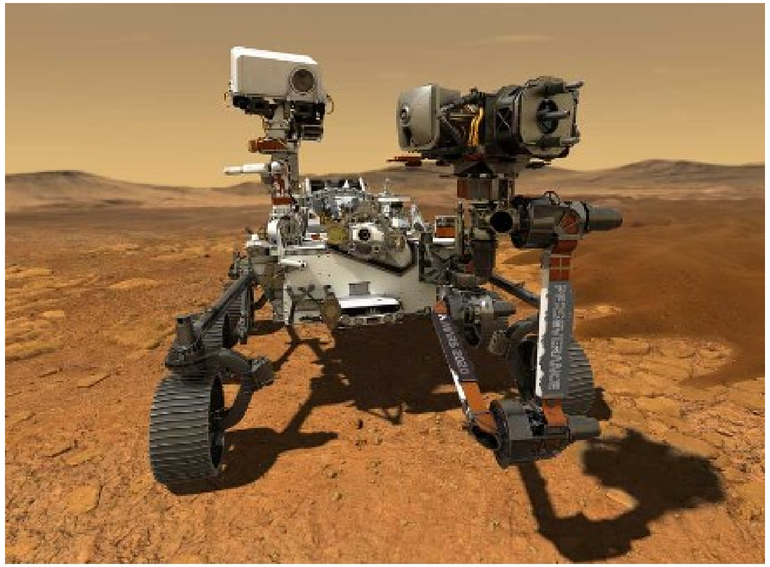
Recommendation for RT application

- Minimize the number of aperiodic tasks in design
- Change real-world random by polling for signals instead of using interrupts
- ISR as deferred server methods – as short as possible
- Keep processor utilisation as low as possible
- Measure the actual spare time so you know what‘s really happening
- Hide: Task location, CPU type, network protocol aspects, signal handing features
- Watchdog

Case Study: Perseverance Rover

- Two computers (Rover Compute Element), 3 antennas
  - Processor PowerPC 750 200 MHz, with 2 GB flash memory, 256 MB RAM
- The flight software runs on the VxWorks Operating System
- Open Souce real-time operating system RTEMS 4.5.0
  - High safety in compliance with IEC 61508 and DO-178B





IT for Practice 2022 Conference Program

Thursday, October 13th 2022

Thank you for attentions....