“art2kitekt”

A modeling and analysis tool for aerospace domain

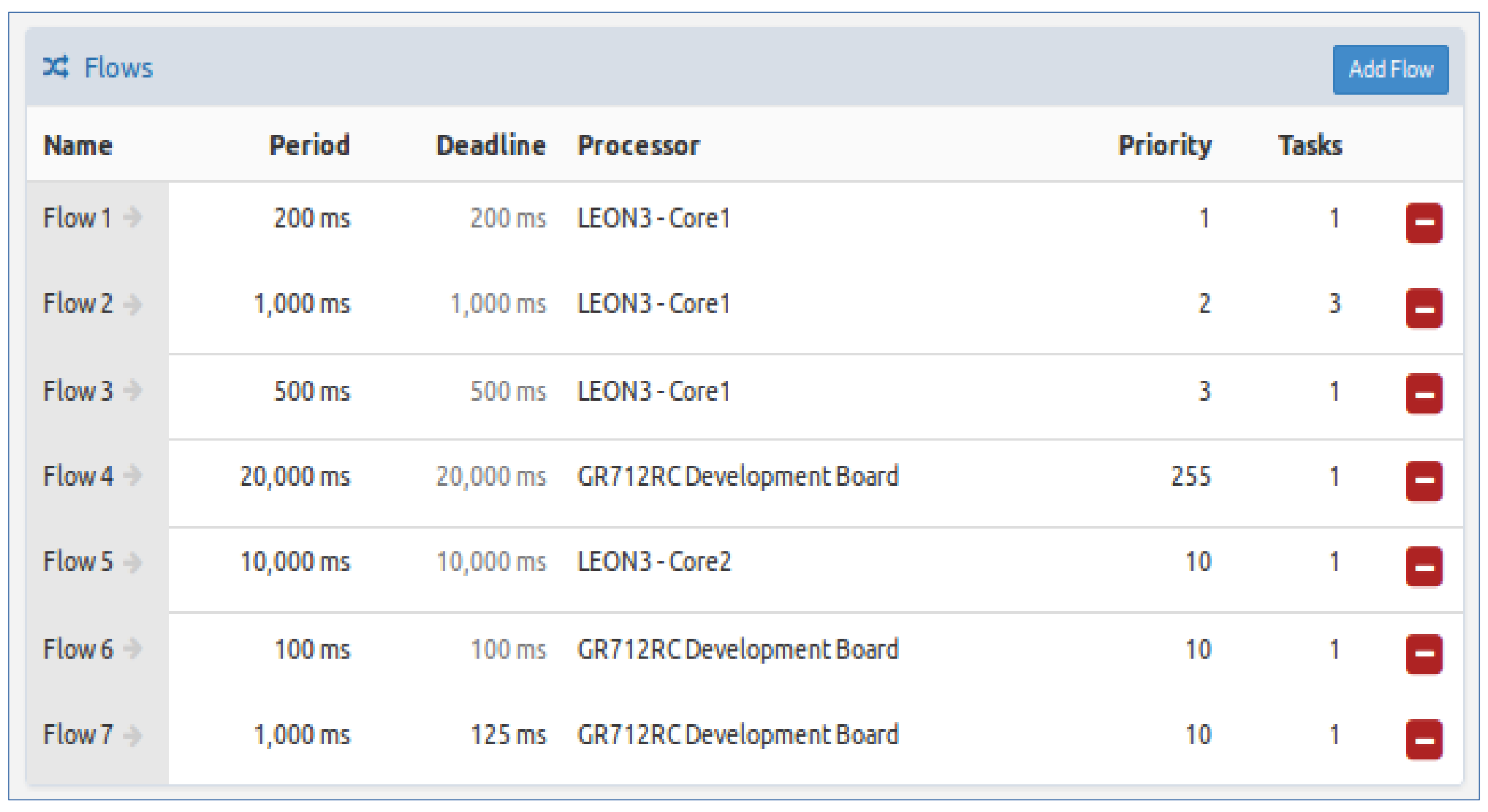
WP02 - WP09

Abstract

- A software tool for the design of HW and SW of mixed criticalilty, real-time systems with the following main features:
  - A tool for **helping the engineer** to design and analyse a high-integrity system composed of HW and SW elements
  - Elementary analysis and assistance, as well as **complex algorithms** to find the best plan for scheduling tasks and allocating resources.
  - It will automate a part of the design phase and generate **evidences to pass the certification process** of the high-integrity systems.
  - Based on profiles, it can be **tailored for the specific needs** of a given application domain or company.
  - Analysis **results** presented in both reports and source code form.

Application Software

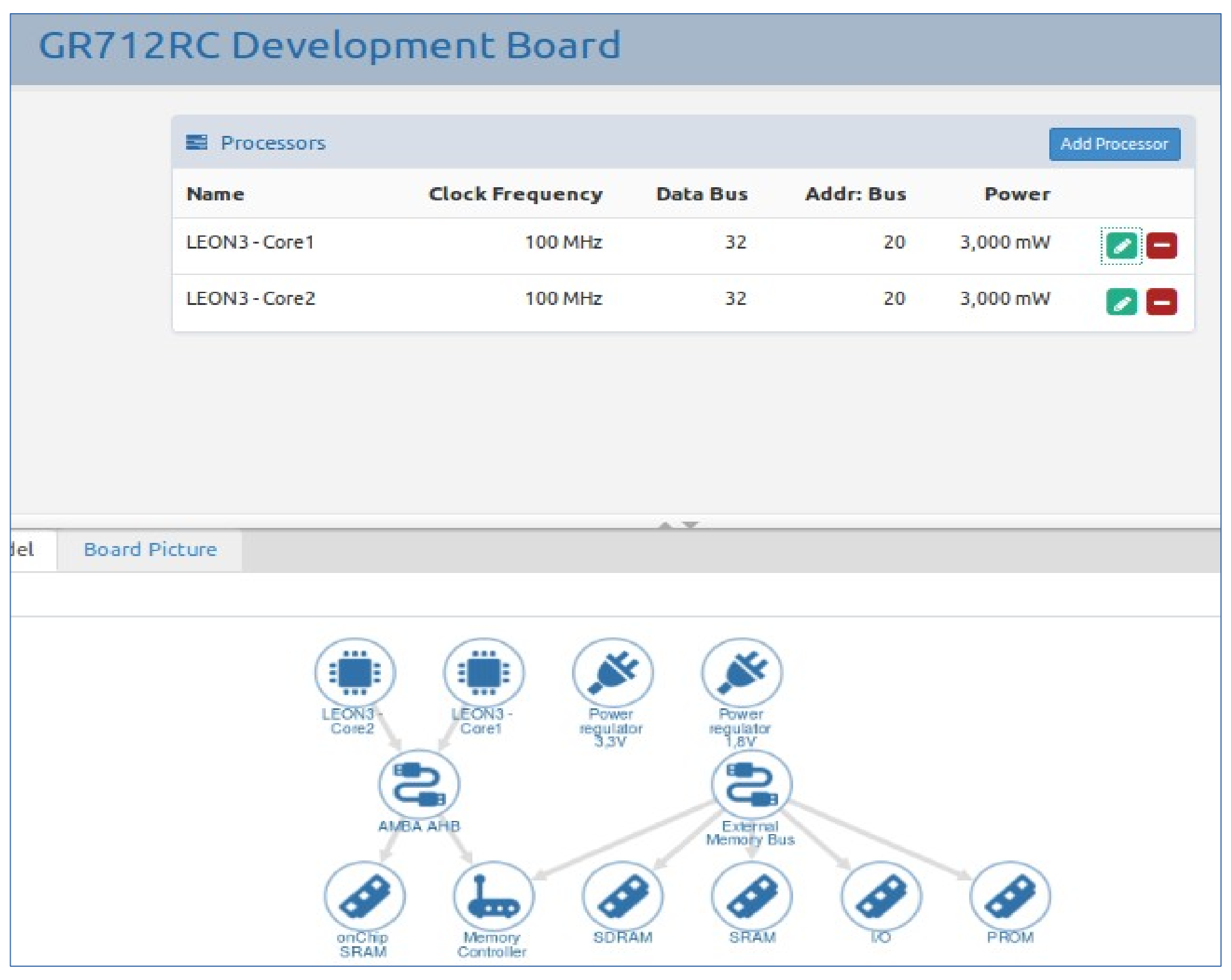
- Application software can be grouped into different subsystems.
- They can be decomposed in a set of tasks with dependencies among them, as well as properties and constraints: worst execution time, periods, deadlines.
- Also shared resources can be used by tasks.



Name	Period	Deadline	Processor	Priority	Tasks
Flow 1	200 ms	200 ms	LEON3 - Core1	1	1
Flow 2	1,000 ms	1,000 ms	LEON3 - Core1	2	3
Flow 3	500 ms	500 ms	LEON3 - Core1	3	1
Flow 4	20,000 ms	20,000 ms	GR712RC Development Board	255	1
Flow 5	10,000 ms	10,000 ms	LEON3 - Core2	10	1
Flow 6	100 ms	100 ms	GR712RC Development Board	10	1
Flow 7	1,000 ms	125 ms	GR712RC Development Board	10	1

Execution Platform

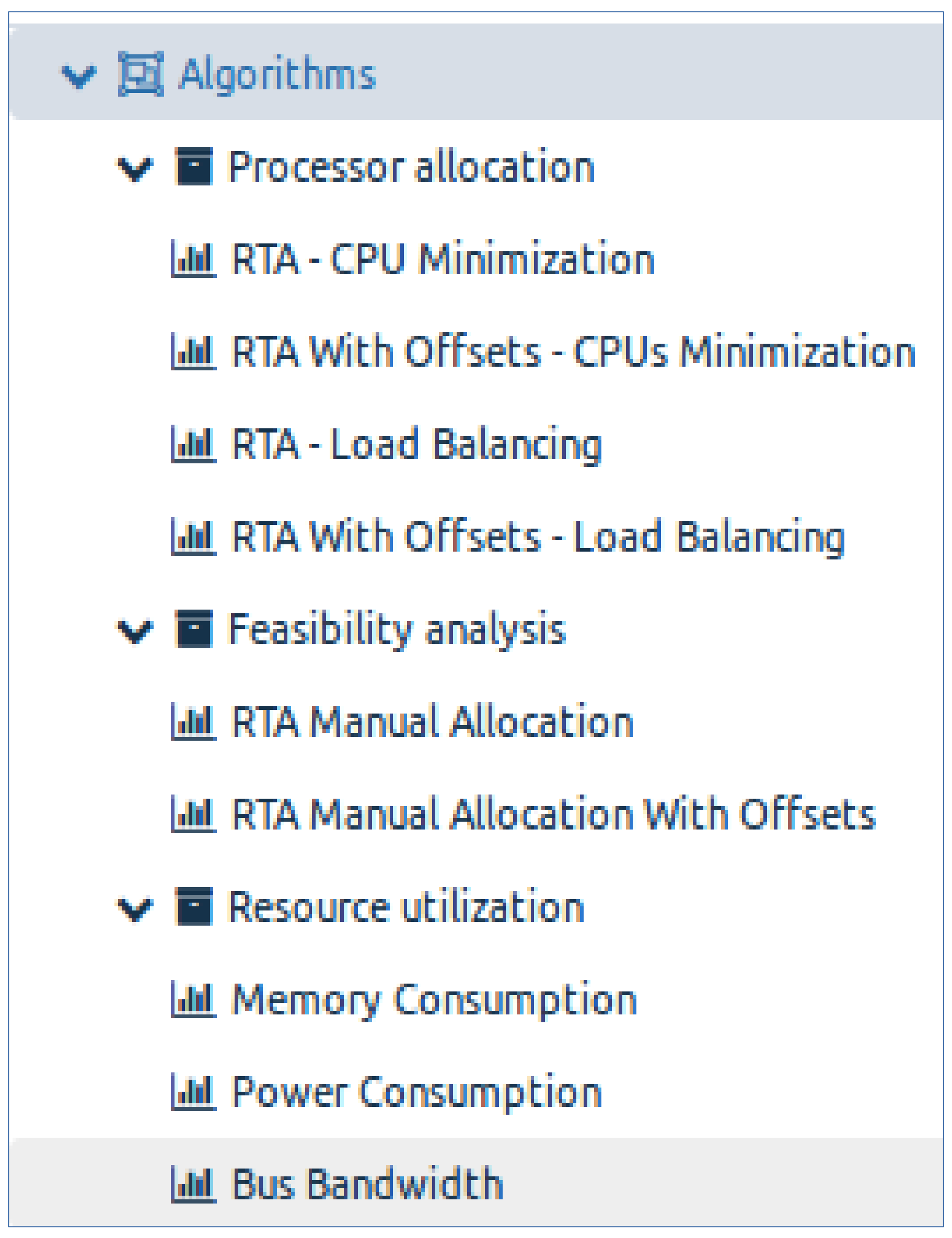
- Different execution platforms redefined and available for the engineer to configure with the corresponding parameters of the specific hardware to be used.
- The engineer is able to custom the platform in terms of CPUs, buses, memories and devices.



GR712RC Development Board

Name	Clock Frequency	Data Bus	Addr: Bus	Power
LEON3 - Core1	100 MHz	32	20	3,000 mW
LEON3 - Core2	100 MHz	32	20	3,000 mW

System Analysis

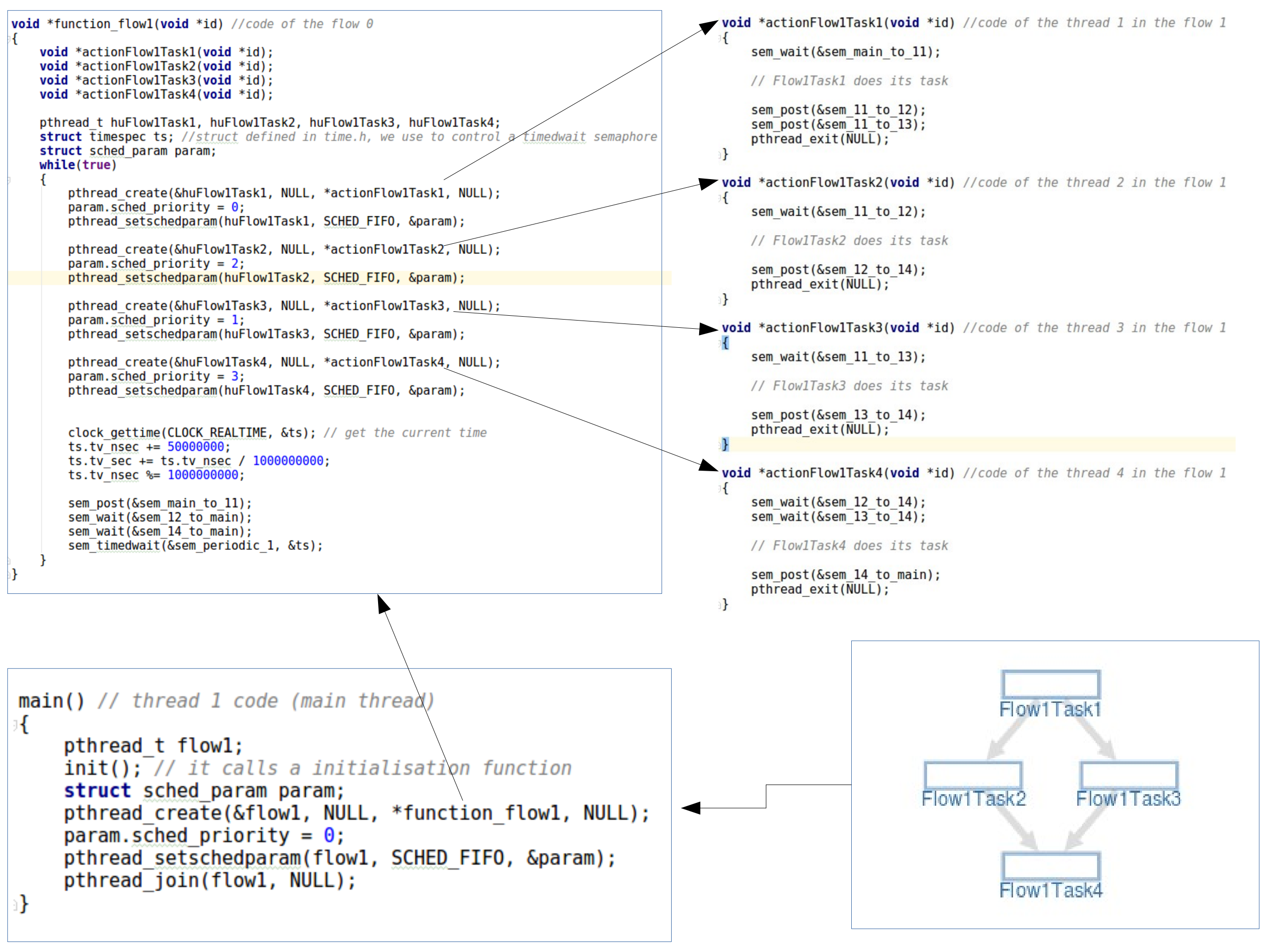


- Algorithms
  - Processor allocation
    - RTA - CPU Minimization
    - RTA With Offsets - CPUs Minimization
    - RTA - Load Balancing
    - RTA With Offsets - Load Balancing
  - Feasibility analysis
    - RTA Manual Allocation
    - RTA Manual Allocation With Offsets
  - Resource utilization
    - Memory Consumption
    - Power Consumption
    - Bus Bandwidth

- The engineer is able to specify the kind of system analysis she wants to perform. It is also possible to interact with the tool to fix any issues that makes the system unfeasible.
- The implementation of these analyses is fast enough to be executed “on the fly”, so the user can easily test several possible conditions and see the result immediately.

Code Generation

- A target platform can be addressed to automatically generate the high level code to manage flows and tasks.



```
void *function_flow1(void *id) //code of the flow 0
{
    void *actionFlow1Task1(void *id);
    void *actionFlow1Task2(void *id);
    void *actionFlow1Task3(void *id);
    void *actionFlow1Task4(void *id);

    pthread_t huFlow1Task1, huFlow1Task2, huFlow1Task3, huFlow1Task4;
    struct timespec ts; //struct defined in time.h, we use to control a timedwait semaphore
    struct sched_param param;
    while(true)
    {
        pthread_create(&huFlow1Task1, NULL, *actionFlow1Task1, NULL);
        param.sched_priority = 0;
        pthread_setschedparam(huFlow1Task1, SCHED_FIFO, &param);

        pthread_create(&huFlow1Task2, NULL, *actionFlow1Task2, NULL);
        param.sched_priority = 2;
        pthread_setschedparam(huFlow1Task2, SCHED_FIFO, &param);

        pthread_create(&huFlow1Task3, NULL, *actionFlow1Task3, NULL);
        param.sched_priority = 1;
        pthread_setschedparam(huFlow1Task3, SCHED_FIFO, &param);

        pthread_create(&huFlow1Task4, NULL, *actionFlow1Task4, NULL);
        param.sched_priority = 3;
        pthread_setschedparam(huFlow1Task4, SCHED_FIFO, &param);


        clock_gettime(CLOCK_REALTIME, &ts); // get the current time
        ts.tv_nsec += 50000000;
        ts.tv_sec += ts.tv_nsec / 1000000000;
        ts.tv_nsec %= 1000000000;

        sem_post(&sem_main_to_11);
        sem_wait(&sem_12_to_main);
        sem_wait(&sem_14_to_main);
        sem_timedwait(&sem_periodic_1, &ts);
    }
}
```

```
void *actionFlow1Task1(void *id) //code of the thread 1 in the flow 1
{
    sem_wait(&sem_main_to_11);

    // Flow1Task1 does its task

    sem_post(&sem_11_to_12);
    sem_post(&sem_11_to_13);
    pthread_exit(NULL);
}
void *actionFlow1Task2(void *id) //code of the thread 2 in the flow 1
{
    sem_wait(&sem_11_to_12);

    // Flow1Task2 does its task

    sem_post(&sem_12_to_14);
    pthread_exit(NULL);
}
void *actionFlow1Task3(void *id) //code of the thread 3 in the flow 1
{
    sem_wait(&sem_11_to_13);

    // Flow1Task3 does its task

    sem_post(&sem_13_to_14);
    pthread_exit(NULL);
}
void *actionFlow1Task4(void *id) //code of the thread 4 in the flow 1
{
    sem_wait(&sem_12_to_14);
    sem_wait(&sem_13_to_14);

    // Flow1Task4 does its task

    sem_post(&sem_14_to_main);
    pthread_exit(NULL);
}
```

```
main() // thread 1 code (main thread)
{
    pthread_t flow1;
    init(); // it calls a initialisation function
    struct sched_param param;
    pthread_create(&flow1, NULL, *function_flow1, NULL);
    param.sched_priority = 0;
    pthread_setschedparam(flow1, SCHED_FIFO, &param);
    pthread_join(flow1, NULL);
}
```

Conclusion

- A tool suite named ***art2kitekt*** developed as an integrated software tool for designing and analyzing mixed criticality, real-time systems.
- It **features**: an unified framework for the whole HW-SW codesign cycle, complex analysis, certification evidences, predefined pofiles, instant feedback, code generation and intiuitive web app.
- Some remarkable **benefits** comprise: provides requisites traceaility, can be used as a certification aid, saves work with predefined profiles, it is simple and guides through the design process, after the system design a linkable and executable can be obtained.

Contributing Partners:

15Q-ITI

ITI INSTITUTO TECNOLÓGICO DE INFORMÁTICA

The research leading to these results has received funding
from the European Commission through the ARTEMIS Joint
Undertaking under grant agreement n° 621429 and from the
Spanish Ministry of Industry, Energy and Tourism.

ECSEL Joint Undertaking

Electronic Components and Systems for European Leadership