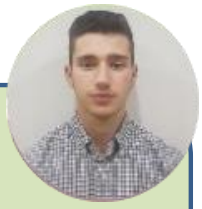
M2 ERTS PROJECT : Implementation of ED-H on Xenomai

DEBUS ALEXY
RAYELLA NIRANJAN

02/12/2021

Our team



Debus Alexy

C dev.

Rayella Niranjan

C dev.

Table of contents

1. Context

Meeting with the professor (M.Queudet) on october the 15th :

The interest here is to put into applications a dynamic-priority energy aware scheduling strategie ED-H through the use of a Real-time kernel.

We will use Xenomai, I mean a dual kernel configurations : a Linux kernel (using Ubuntu distributions) supplementing by a RT co-kernel (Cobalt)

A Master student recently integrated EDF into Xenomai…

2. Key question

We need to have ED-H working without affecting integrity of other scheduling policies (fixed priority, EDF).

How could we manage to integrate/validate ED-H on Xenomai ?

3. Approach

1) Learn about Xenomai
2) Validate/Assert EDF implementation on Xenomai
3) Understand EDF implementation
4) Learn about ED-H
5) Implement ED-H In Xenomai
6) Validate/Assert ED-H implementation on Xenomai

The project will be split into sprints (1 sprint = 3-4 weeks)

4. Primary objectives

Primary objectives are established at the beginning of the project

1) Installation of modified Xenomai and validation for EDF integration
Metric : report for modifications added to have a functional installation + tests to validate xenomai-EDF (report + source code)
Estimated time : 10h

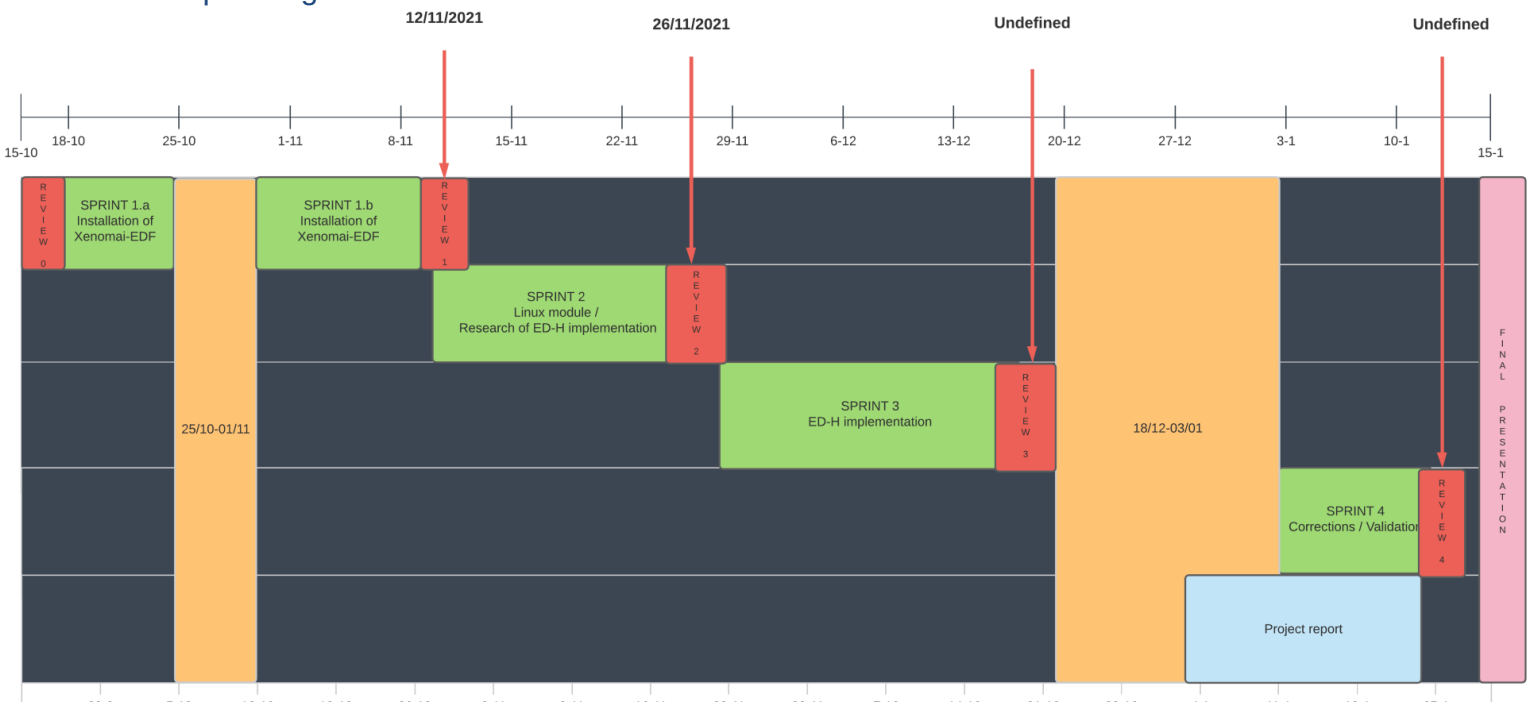
2) Development of a linux module to grab battery information and transfer to Cobalt
Metric : guideline for the creation/usee of the linux module + source code for the linux module
Estimated time : 25h

3) ED-H implementation and validation on Xenomai
Metric : guideline for the creation/use of the linux module + source code for the linux module
Estimated time : 50h

Remaining time : 15h (Redaction of the final report, PWP presentation, handle git repo.)

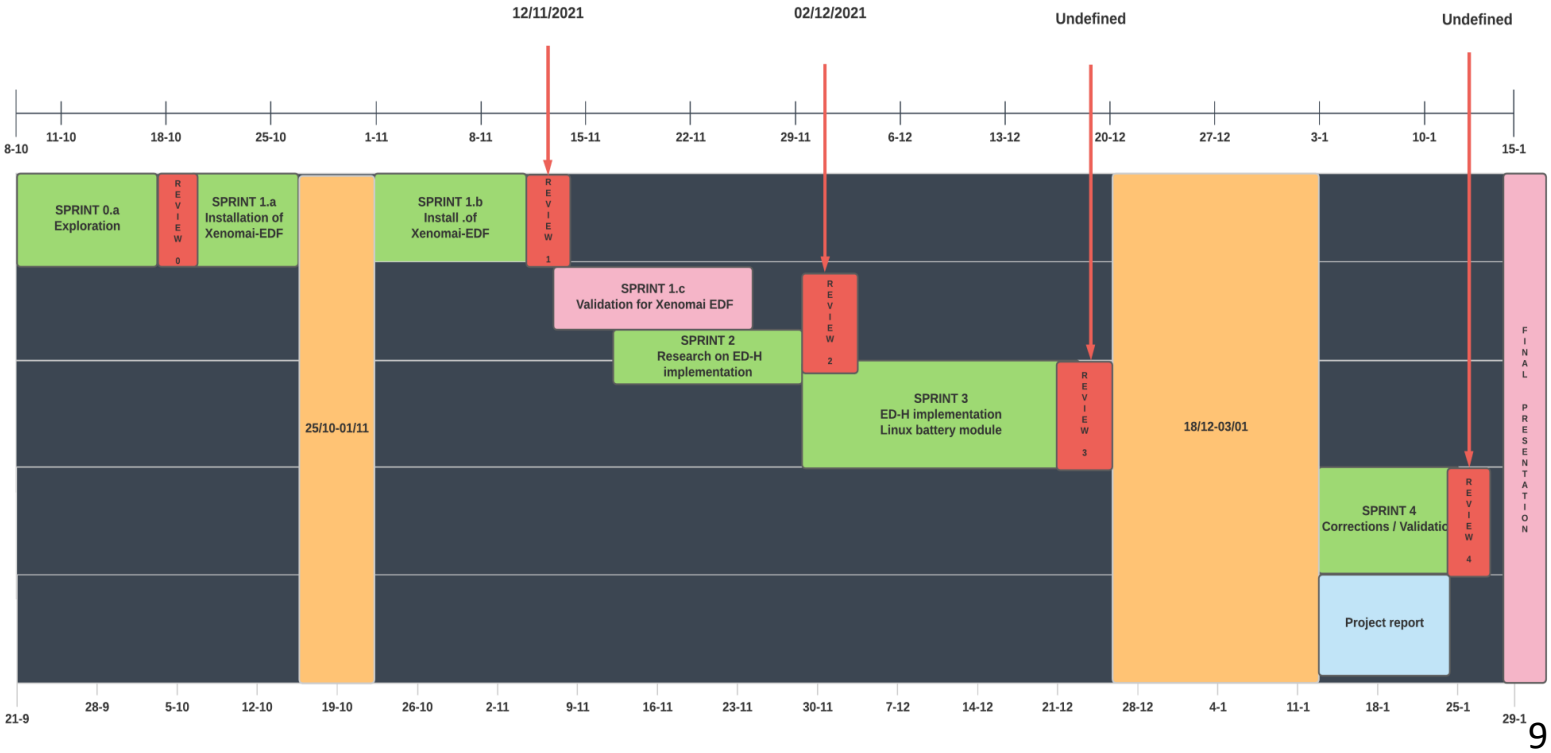
5. Planning

Provisional planning:



5. Planning

Effective planning :



6. Progress

0) Development of linux module to grab battery information
Results: Start implementation of a simple linux module (using kernel header file linux/power_supply)
2 weeks (08/10 - 17/10)
- In charge : Debus Alexy

1) Installation of Xenomai EDF
Results: Successfully installed Xenomai and validate EDF for no-premption test cases
3 weeks (18/10 - 11/11
- *In charge : Debus Alexy & Rayella Niranjan*

2) Validation of Xenomai EDF
Results: Validate EDF for premption test cases
3 weeks (13/11 - 26/11
- In charge : Debus Alexy

7. Project evolution

Actual : (13/11 – 01/12)

1. Study of EDH (Theory for EDH)

Future (02/12 - 17/12) :

Read EDF source code / EDH implementation

7. Presentation of EDH

Energy Harvesting : Energy sources from environment , which can be harvested and converted them in to electric sources . It can be used to power wireless systems .

EDH Scheduling Algorithm : Objective of EDH scheduling algorithm Authorize job execution as long as there occurs no starvation in future.

1. Static Analysis : Energy, Deadline constraints.
2. Dynamic Analysis : Dynamic Slack energy with respect to current time .

7. Presentation of EDH Simulator Functions

Feasibility check :

*Untitled - Notepad

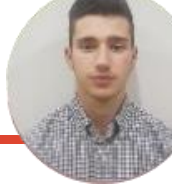
File Edit Format View Help

```
function generate-ftable
 for each task do
    T ask[i].nextarrival ← T ask[i].arrival
    Task[i].nextdeadline ← T ask[i].deadline
 end
Bool pending-next-arrive,pending-same-arrive = TRUE
while pending-next-arrive do
  if Task[i].nextarrival ≥ Hyperperiod then
      pending-next-arrive = FALSE
  end
while pending-same-arrive do
     endIndex ← f ind − min − end
    StartIndex ← f ind − min − start
    Earliestarr ← T ask[StartIndex].nextarrival
    Earliestdeadline ← T ask[endIndex].nextdeadline
   if Earliestdeadline ≤ Hyperperiod then
       Feasibility-checkEarliestarr,Earliestdeadline
   end
  if Earliestdeadline ≥ Hyperperiod then
     pending-same-arrive = FALSE
  end
end
Task[StartIndex].nextarrival+ = T ask[StartIndex].period end

function Feasibility-check (Earliestarr, Earliestdeadline)
Calculate hw, gw, SSE, SSTW
if SSE < 0 or SSTW < 0
  then
    Bool feasibility-pass = false;
  end
return feasibility-pass
```

Ln 31, Col 1 | 100% | Windows (CRLF) | UTF-8

7. Project evolution



Actual : (13/11 – 01/12)

1. Validation for Xenomai EDF (Preemption for 2/3/4 tasks)
2. Research for XDDP-based communication
3. Read/Understand EDF source code

Future (02/12 - 17/12) :

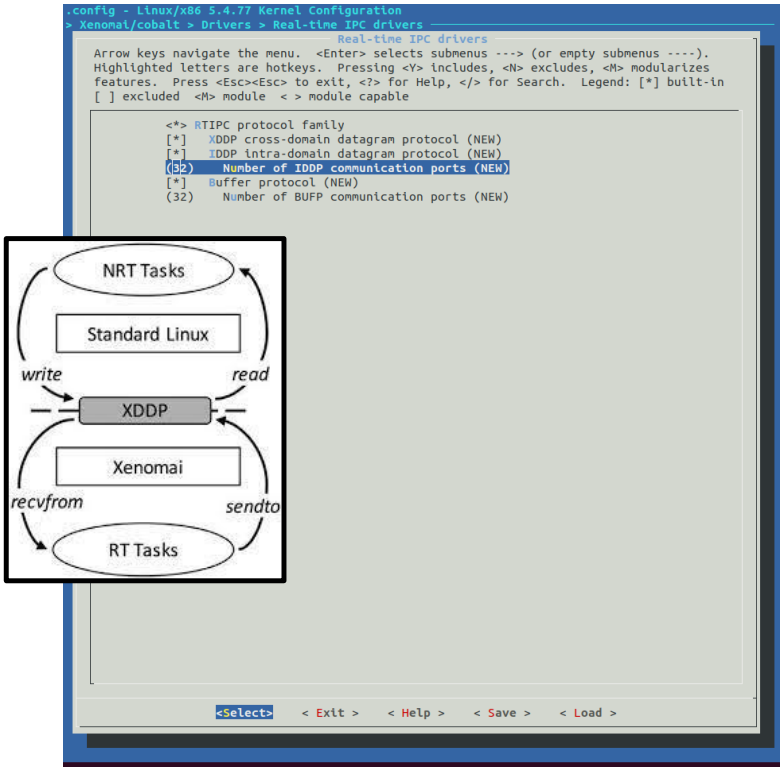
Source code for EDH (including Battery linux module)

8. Presentation of sprint 2

1) Need to compilate linux kernel
With XDDP protocol enabled

On Xenomai kernel space :
RW on port /dev/rtp0

On standard linux kernel space :
RW on port
/proc/xenomai/registry/rtipe/xddp/0



8. Presentation of sprint 2

Added lines in .config for linux kernel compilation

```
#
# Real-time IPC drivers
#
CONFIG_XENO_DRIVERS_RTIPC=y
CONFIG_XENO_DRIVERS_RTIPC_XDDP=y
CONFIG_XENO_DRIVERS_RTIPC_IDDP=y
CONFIG_XENO_OPT_IDDP_NRPORT=32
CONFIG_XENO_DRIVERS_RTIPC_BUFP=y
CONFIG_XENO_OPT_BUFP_NRPORT=32
# end of Real-time IPC drivers
```

8. Presentation of sprint 2

File Schedqueue.h

```c
typedef struct list_head xnsched_queue_t;

#define xnsched_initq(__q)                      INIT_LIST_HEAD(__q)
#define xnsched_emptyq_p(__q)                   list_empty(__q)

#define xnsched_addq(__q, __t)                  list_add_prilf(__t, __q, cprio, rlink)
#define xnsched_addq_tail(__q, __t)             list_add_priff(__t, __q, cprio, rlink)

#define xnsched_dyna_addq(__q, __t)             list_add_prilf_bis(__t, __q, next_deadline, rlink)
#define xnsched_dyna_addq_tail(__q, __t)        list_add_priff_bis(__t, __q, next_deadline, rlink)

#define xnsched_delq(__q, __t)                  (void)(__q), list_del(&(__t)->rlink)
#define xnsched_getq(__q)                                                       \
        ({                                                                      \
                struct xnthread *__t = NULL;                                    \
                if (!list_empty(__q))                                           \
                        __t = list_get_entry(__q, struct xnthread, rlink);      \
                __t;                                                            \
        })
#define xnsched_weightq(__q)                                            \
        ({                                                              \
                struct xnthread *__t;                                   \
                __t = list_first_entry(__q, struct xnthread, rlink);    \
                __t->cprio;                                             \
        })

#endif /* !CONFIG_XENO_OPT_SCALABLE_SCHED */
```

File list.h

```c
#include <linux/list.h>

#define __list_add_pri(__new, __head, __member_pri, __member_next, __relop)
do {
        typeof(*__new) *__pos;
        if (list_empty(__head))
                list_add(&(__new)->__member_next, __head);
        else {
                list_for_each_entry_reverse(__pos, __head, __member_next) {
                        if ((__new)->__member_pri __relop __pos->__member_pri)
                                break;
                }
                list_add(&(__new)->__member_next, &__pos->__member_next);
        }
} while (0)

#define list_add_priff(__new, __head, __member_pri, __member_next)
        __list_add_pri(__new, __head, __member_pri, __member_next, <=)

#define list_add_prilf(__new, __head, __member_pri, __member_next)
        __list_add_pri(__new, __head, __member_pri, __member_next, <)

#define list_add_priff_bis(__new, __head, __member_pri, __member_next)
        __list_add_pri(__new, __head, __member_pri, __member_next, >=)

#define list_add_prilf_bis(__new, __head, __member_pri, __member_next)
        __list_add_pri(__new, __head, __member_pri, __member_next, >)
```

Thanks for your attention.

Do you have any questions ?