A Kernel for Energy-Neutral Real-Time Systems with Mixed Criticalities

Peter Wägemann, Tobias Distler, Heiko Janker, Phillip Raffeck, Volkmar Sieh

22nd Real-Time and Embedded Technology and Applications Symposium
RTAS 2016

April 12, 2016

partly funded by:

DFG
Application Scenario – Energy-Neutral Real-Time Systems

Energy-Neutral Real-Time Systems

- Energy-neutral prototype of the I4Copter\(^1\)
- Main modes: flying, landed/harvesting

\(^1\) Ulbrich et al. *I4Copter: An Adaptable and Modular Quadrotor Platform*, SAC'11
Application Scenario – Energy-Neutral Real-Time Systems

- Expand panels for charging
- Collapse panels during flight
- Lightweight, robust solar films

organic solar films: Belectric OPV, solarte.de
Problems – Energy-Neutral Real-Time Systems

Real-Time Systems under Hard Energy Constraints

- Guarantee **timeliness**: position/flight control
- Respect **energy constraints**: limited/replenishable storage

*Energy surpasses timeliness and vice versa*
Challenges of Energy-Neutral Real-Time Systems

1. **Scheduling**: mixed constraints (time & energy), mixed criticalities
2. **Energy monitoring**: consumed & available energy budgets
3. **Consistency**: survive blackout periods
The **EnOS** Kernel

1. **Scheduling** with mixed criticalities and mixed constraints
2. Energy-budget **monitoring**
3. Surviving **blackouts** (suspend and resume)
Outline

1. Motivation
2. Scheduling with Mixed Criticalities & Mixed Constraints
3. Energy-Budget Monitoring
4. Surviving Blackouts
5. Evaluation
6. Conclusion
Motivation – Mixed-Criticality Systems

Guarantees $\leadsto$ Pessimism $\leadsto$ Slack Time

- “Flight critical”: inertial measurement/control $\Rightarrow C(HI)$
- “Mission critical”: capturing images $\Rightarrow C(LO)$
- **Hard guarantees** for **high time criticality** tasks
- More critical task $\Rightarrow$ more pessimistic WCET estimate
- **Redistribute** resources from $LO$ to $HI$ (preempt $LO$)
Energy-Constrained Mixed-Criticality Systems

Völp et al. Has Energy Surpassed Timeliness?
Scheduling Energy-Constrained Mixed-Criticality Systems, RTAS’14

Mixed constraints: energy & time

Approach (EA-OCBP) considers
- $C(LO), \ldots, C(HI)$, and $E(LO), \ldots, E(HI)$
- Hyperperiod energy budget: $E^{HP}$

More pessimistic WCET $C \rightsquigarrow$ more pessimistic WCEC $E^{1}$

Coupling of time and energy constraints

$^{1}$worst-case energy-consumption (WCEC)
Time Criticality versus Energy Criticality

When is **timeliness** most critical?
- During **activity phases** (i.e., flight mode)
- Uncritical in suspension phase
  - No time guarantees across blackout periods

When are **energy constraints** most critical?
- Close to **inactivity phases** (i.e., depletion)
- Safe suspension to persistent memory
Time Criticality versus Energy Criticality

When is timeliness most critical?

**Time vs. Energy Criticality**

1. Energy is critical
2. Time is critical
3. ... but not necessarily at the same time

- **Decoupling** of time and energy into **distinct energy modes**
- Safe suspension to persistent memory
EnOS | Scheduling with Mixed Criticalities & Mixed Constraints – Scheduling Approach | 12
Energy-Criticality Modes & Time-Criticality Levels

**Multi-Mode Model**

- Time (criticality) *levels & energy (criticality) modes*
- Energy budget for hyperperiod: $E_{HP}^{M_i}$
- **Different functionality** (task set) per mode possible
- **Less pessimistic WCEC analyses** in less critical energy modes
- Full functionality when energy plentiful

**Decoupling** energy and time criticalities
Task model: periodic, implicit deadlines, independent tasks, mixed criticality & mixed constraints: $LOT^T$, $HIT^T$, $LOE^E$, $HIE^E$

Static priority assignment (using EA-OCBP)

### Energy-Budget Composition on Mode $M_i$

- **Execution budget** $EB$: estimate *depending on mode*
  - Stay on mode for *configurable number of hyperperiods* ($H$)
- **Grace budget** $GB$: handle budget violation in hyperperiod
  - *Ensures finishing hyperperiod*
  - Enable *safe mode switches*
How to get **safe upper bound** on Hyperperiod’s WCEC?

- WCEC of lowest energy-criticality mode is optimistic, not safe
- Trick exploits
  - **Fixed duration** of hyperperiod $HP_{M_i}$
  - **Physical upper limits** of electronic components
    - $E_{grace} = P_{max} \cdot HP_{M_i}$
    - **Very pessimistic, but safe** (required once per mode)
Scheduling – Online Part

Current Energy:

- Full battery
- Empty battery

Mode switches based on available and offline determined budgets

1. $M_1 \rightarrow M_2$
2. $M_2 \rightarrow M_{susp}$
3. $M_2 \leftarrow M_{susp}$
4. $M_1 \leftarrow M_2$

$EB_i$: Execution budget
$GB_i$: Grace budget

- Monitoring energy budgets of modes $M_i$
- Mode switches based on available and offline determined budgets
- Grace budget for safe degradation: allow finishing hyperperiod

$EB^i$: Execution budget
$GB^i$: Grace budget
Scheduling Enhancement

EA-OCBP Approach

Multi-Mode Model of EnOS

Enhancement

- Avoid pessimism when energy is plentiful
- Full decoupling of WCET and WCEC
1. Motivation

2. Scheduling with Mixed Criticalities & Mixed Constraints

3. Energy-Budget Monitoring

4. Surviving Blackouts

5. Evaluation

6. Conclusion
Deadline vs. Energy-Budget Violations

How to detect energy-budget violations?

- Detecting deadline violations
  - Hardware operates timing-oriented
  - General-purpose timers
- Detect energy-budget violations

Detecting deadline violations

Detect energy-budget violations

EnOS Energy-Budget Monitoring
Energy-Budget Monitoring – Concept

Lightweight Signalling Mechanism for Energy-Related Events

- Monitor budget violations
- **Polling** energy storage
  - **Overhead**
  - Not feasible as *wakeup mechanism* (from deep-sleep modes)

*Notifications for energy-related events: “energy interrupts”*
Exploitation of Commercial Off-the-Shelf Hardware

- Setup notifications
  - Digital-analog converter: set threshold
  - Comparator: issue energy interrupt

- Supercaps
  - Linear relationship between voltage and energy
  - No additional charging circuit for charging

- Used for both violations and wakeup for resumption
Outline

1. Motivation

2. Scheduling with Mixed Criticalities & Mixed Constraints

3. Energy-Budget Monitoring

4. Surviving Blackouts

5. Evaluation

6. Conclusion
Surviving Blackouts: Problem

When is enough energy harvested?

✔ Suspend early enough
  ✢ Exploit a priori knowledge of WCEC
  - Avoid waking up and immediately suspending
  - Forward-looking scheduling required

✗ When to wake up safely for useful operation?
Suspension & Resumption

Safe Resumption

- Minimum of hyperperiods executions
- Configurable energy mode for resumption \((M_1, M_2, \ldots, M_n)\)
- Allow developer to state *useful work*
  
  Wakeup only when *enough energy* to do *useful work*
1. Motivation

2. Scheduling with Mixed Criticalities & Mixed Constraints

3. Energy-Budget Monitoring

4. Surviving Blackouts

5. Evaluation

6. Conclusion
Hardware Components

1. Solar cells
2. Buck-Boost-Converter
3. Supercapacitors
4. Switching regulator
5. Freescale KL46Z: Cortex-M0+
6. Non-volatile FRAM
Mode-Dependent Energy Budgets

Baseline: coupling of WCET and WCEC (using EA-OCBP)

Synthetically generated task sets (using UUniFast)

Criticality factor between $E(HI)$ and $E(LO)$: 0.7x

脿 Up to 17.5% smaller energy budget
Setup *energy interrupt* and monitor offset

**Worst-case resolution**: 198 mJ (mean: 94 mJ, few seconds)

Useless as basis for single scheduling decisions of tasks

Feasible to monitor modes
**Precision of Execution Budgets**

<table>
<thead>
<tr>
<th>Measurement &amp; modes</th>
<th>$E_{HI}^{HP}$</th>
<th>$E_{MID}^{HP}$</th>
<th>$E_{LO}^{HP}$</th>
<th>$E_{measured}^{HP}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 2 3 4 5 6 7</td>
<td>6.83</td>
<td>6.04</td>
<td>5.39</td>
<td>5.17</td>
</tr>
</tbody>
</table>

Execution budgets for a hyperperiod [in mJ]

- **WCEC analysis** [1] of task set (memory- and CPU-bound tasks)
- Pessimistic, but safe $E_{HI}^{HP}$
- Precise computation of $E_{LO}^{HP}$ (respecting overheads & idle times)
- **Smaller budgets** (21%, 11%)
  - Enable switch back earlier when energy is less critical

---

Evaluating Energy-Mode Switches

Energy and mode trace on hardware

Configured hyperperiods: $M_1$: $1000 \times (100 \text{ s})$, $M_2$: $2000 \times (200 \text{ s})$

Requested number of HPs on mode fulfilled

Low offsets of mode switches between 0.3% and 3%

Less than grace budget

<table>
<thead>
<tr>
<th>Mode Switch</th>
<th>Time</th>
<th>Static $E_{\text{switch}}(i)$</th>
<th>Observed $E_{\text{switch}}(i)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_1 \rightarrow M_2$</td>
<td>336 s</td>
<td>12,252 mJ</td>
<td>12,218 mJ</td>
</tr>
<tr>
<td>$M_2 \rightarrow M_{\text{susp}}$</td>
<td>579 s</td>
<td>87 mJ</td>
<td>65 mJ</td>
</tr>
<tr>
<td>$M_{\text{susp}} \rightarrow M_1$</td>
<td>636 s</td>
<td>17,640 mJ</td>
<td>18,122 mJ</td>
</tr>
<tr>
<td>$M_1 \rightarrow M_2$</td>
<td>742 s</td>
<td>12,252 mJ</td>
<td>12,218 mJ</td>
</tr>
</tbody>
</table>
Outline

1. Motivation

2. Scheduling with Mixed Criticalities & Mixed Constraints

3. Energy-Budget Monitoring

4. Surviving Blackouts

5. Evaluation

6. Conclusion
Conclusion

Application Scenario:
Energy-Neutral Real-Time Systems

Scheduling with Mixed Criticalities & Mixed Constraints

✓ Energy-Criticality Modes & Time-Criticality Levels
✓ Multi-mode model: decoupling time and energy

Energy-Budget Monitoring

✓ Exploit existing peripherals (i.e., CMP, DAC)
✓ Mechanism to wakeup & switch modes: energy interrupts

Surviving Blackouts

✓ Safe degradation of services through modes until suspension
✓ Resumption when energy for modes guaranteed
Questions

Get EnOS...

**gitlab.cs.fau.de/enos**

Lots of thanks to...

- Our anonymous shepherd(s)
- Andreas Distler
- Peter Ulbrich

Questions?

Thank you for your attention!