# Abstract

This paper presents the FGPARTS approach for resolving discontinuous power supplies in real time embedded systems. FGPARTS allows the hardware designer to use smaller and less expensive passive components while ensuring that critical tasks complete as required, even when faced with unreliable power sources. The FGPARTS approach is aware of both power supply and consumption at a fine grained level. Each peripheral or group of peripherals is submetered and task dispatch decisions are influenced by their usage heuristics as well as deadline requirements.

# Categories

Power Aware Scheduling

# Key Words

Power-aware, Scheduling, Embedded Systems, Real-Time

# Introduction

Real Time tasks are characterized as those whose results are worthless when not completed by a specified deadline. FGPARTS records the energy used by a task as well as the energy supplied by the system from dispatch time through release. By using these data points and calculating the rate of change in energy, the scheduler can adjust the priority of tasks to match the current system status.

In particular, after the system has run the task set long enough to generate sufficient heuristics, the scheduler should never schedule a task that will not release normally due to power constraints. Additionally, the scheduler can acknowledge and prevent lower priority tasks with low energy consumption from starving those with high priority and high energy consumption in an energy limited environment.

## Power Aware Scheduling

Talk about Mosse’s paper with checkpointing and then running at full speed in order to meet deadlines after recovery.

## Checkpointing

Discuss checkpointing as introduced by Mosse if we are going to be using it here. I would prefer to use a concept of registered memory rather than official checkpointing. In that case we can discuss that here instead, possibly mention its relationship to checkpointing.

## Producers and Consumers

In an energy-limited embedded system there may be energy producers (Solar, Piezo, Magnetic, RF Harvesting, etc) and there are energy consumers consisting of the CPU and any peripherals. With FGPARTS, we track the energy contribution of all producers in aggregate. We track the consumption of energy by all peripherals individually. This allows us to characterize a task by which peripherals it uses and for how long.

# Related Work

## Feedback controlled scheduling

# A motivating Example

Similar to the Gantt chart used in the Mars rover example.

# experimental Setup

## Hardware

Boards are placed in the following configuration

1. A supervisor board is capable of taking inputs from each submetered section of each DUT
2. The DUT with FGPARTS scheduler enabled
3. A control DUT using rate monotonic scheduling and checkpointing
4. Other control DUTs? I would like to use 4 total.

The supervisor board has a constant power supply and monitors the functionality of the worker boards while the power supply is fluctuating into the system.

All boards have an identical and typical distribution of peripheral devices: EEPROM, FLASH, RAM, RS232, RS485, Zigbee, ADC, DAC, LCD and USB.

## Firmware

The FGPARTS approach depends on collecting accurate power consumption from each branch of peripherals in the system.