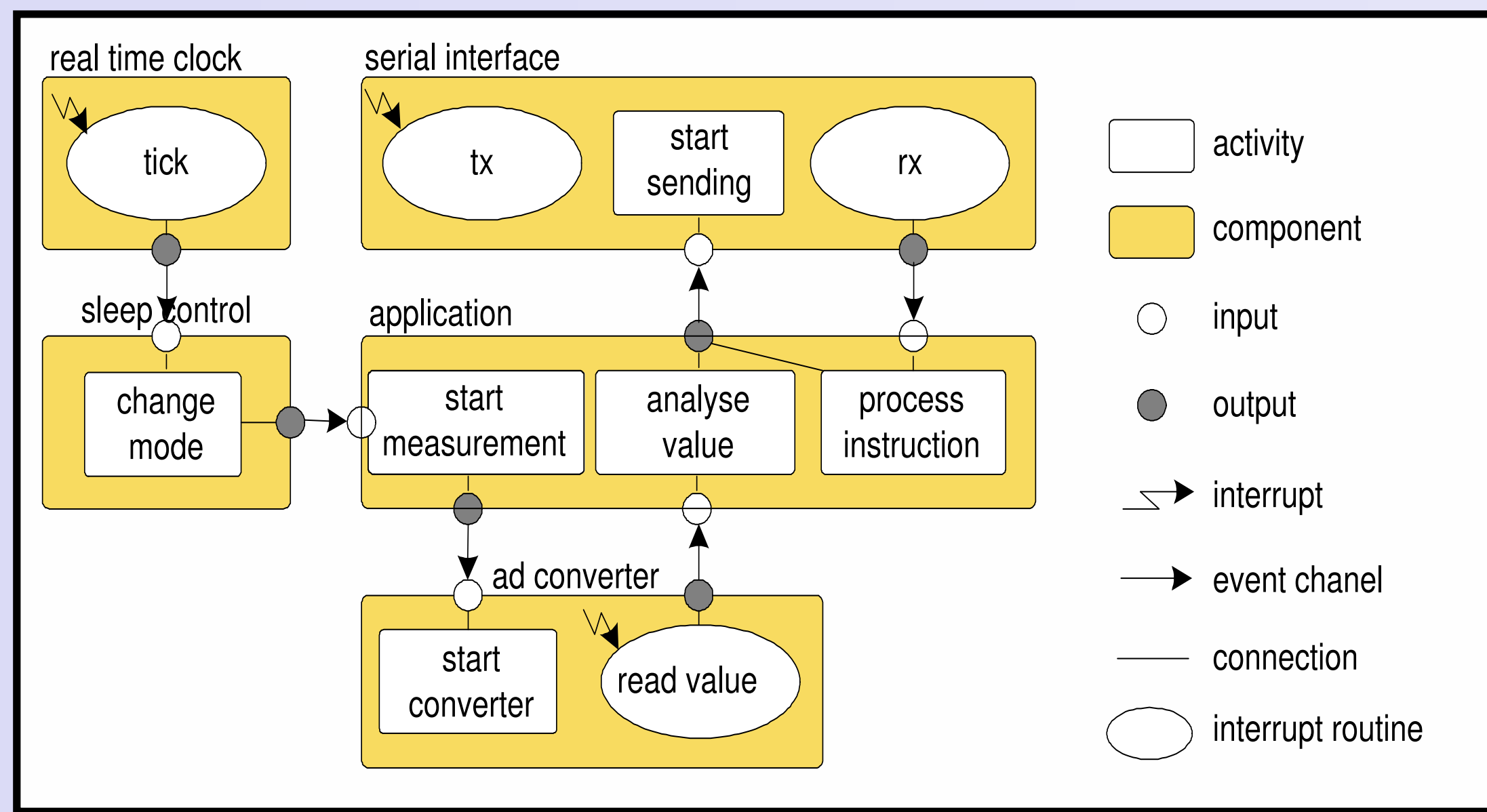


Implicit Sleep Mode Determination in Power Management of Event-driven Deeply Embedded Systems

André Sieber, Karsten Walther, Reinhardt Karnapke, Andreas Lagemann, Jörg Nolte
Distributed Systems/Operating Systems group, BTU Cottbus, Germany



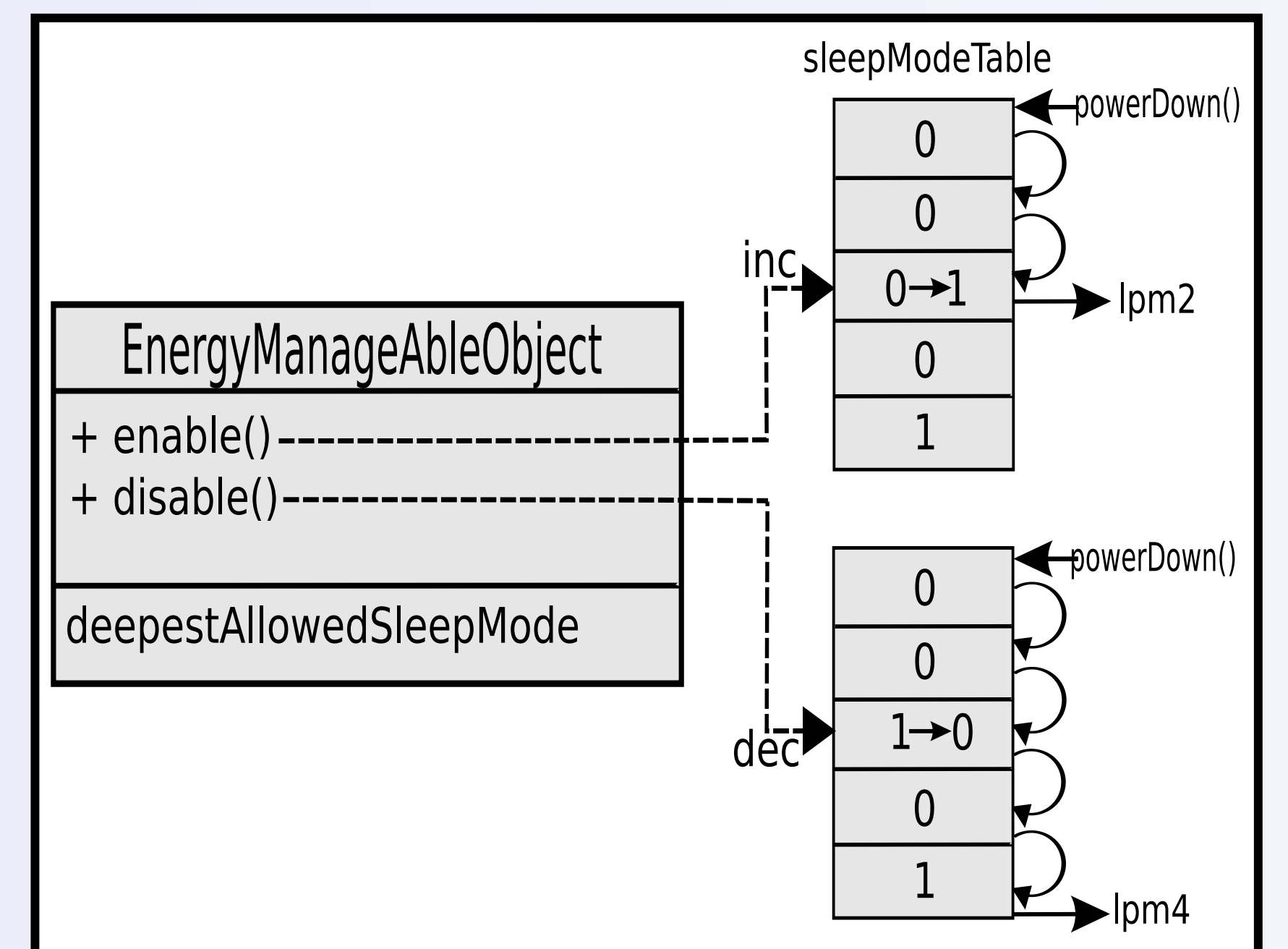
Realtime Event Flow EXECutive

- Reflex Eventflow Model
- fully event driven operation
- component based applications
- implemented in C++
- low resource consumption
- very portable
- buffering and filtering event channels
- scheduling framework (FCFS, FP, EDF, TT)
- implicit synchronization
- **implicit energy management**

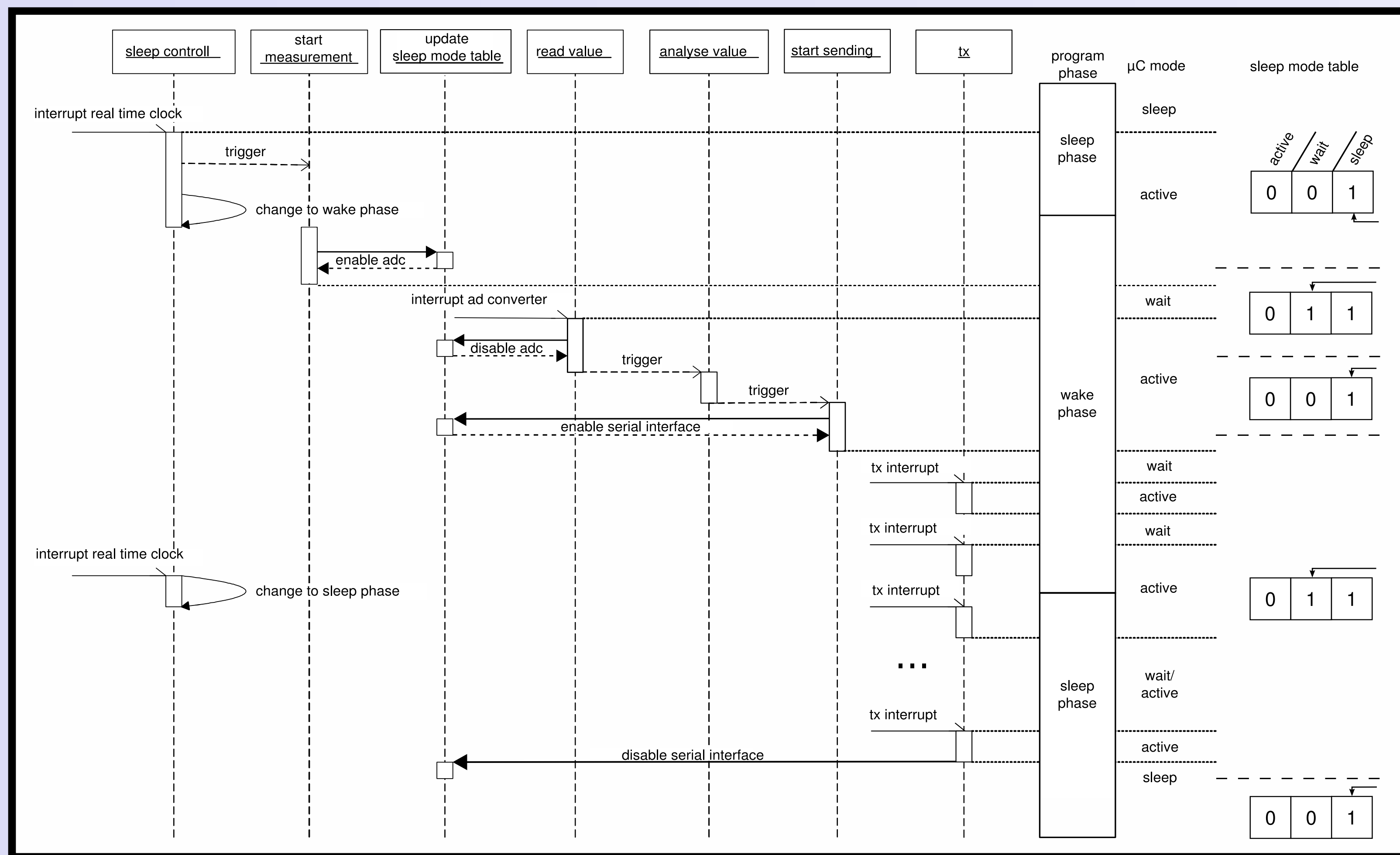
REFLEX

system view

- every instance of a component has a variable that specifies the deepest possible sleep mode that could be used when it is active
- power manager contains **sleep mode table** with counters for every available sleep mode
- component activation leads to incrementation of corresponding counter in the table
- if no event is pending the power manager iterates through the table starting at the lightest mode, first non zero value defines the deepest possible sleep mode
- not necessary to evaluate the complete machine state like in TinyOS



example



user view

modes

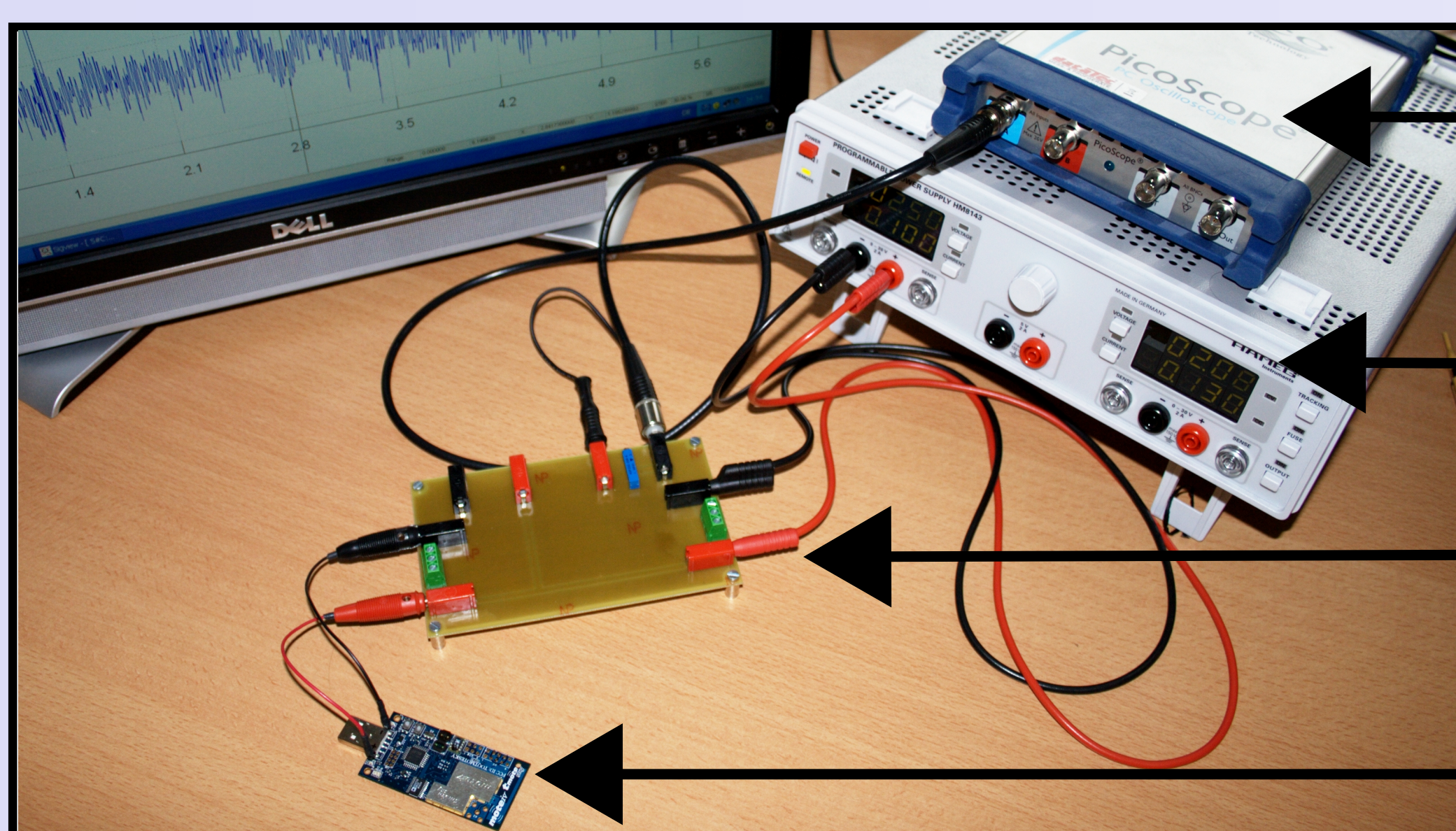
- modes utilize different parts of hard and software
- allow dividing the execution of an application into different phases
- programmer is responsible for defining and changing modes, e.g. timer driven

groups

- each manageable object is part of at least one group
- groups independently activated and deactivated
- objects in multiple groups are only deactivated when all of their groups are deactivated

experimental setup & results

Tmote sky @ 1Mhz, 2.2V - running TinyOS 2.2 or REFLEX
simple time triggered send application



- oscilloscope
- power supply unit
- shunt board
- TMoteSky

