시스템 최신기술
(차량 제어 시스템 Week 3)

Prof. Jong-Chan Kim
Graduate School of Automotive Engineering
Kookmin University

OSEK/VDX

- OSEK/VDX dominates the automotive control domain as the industry standard RTOS

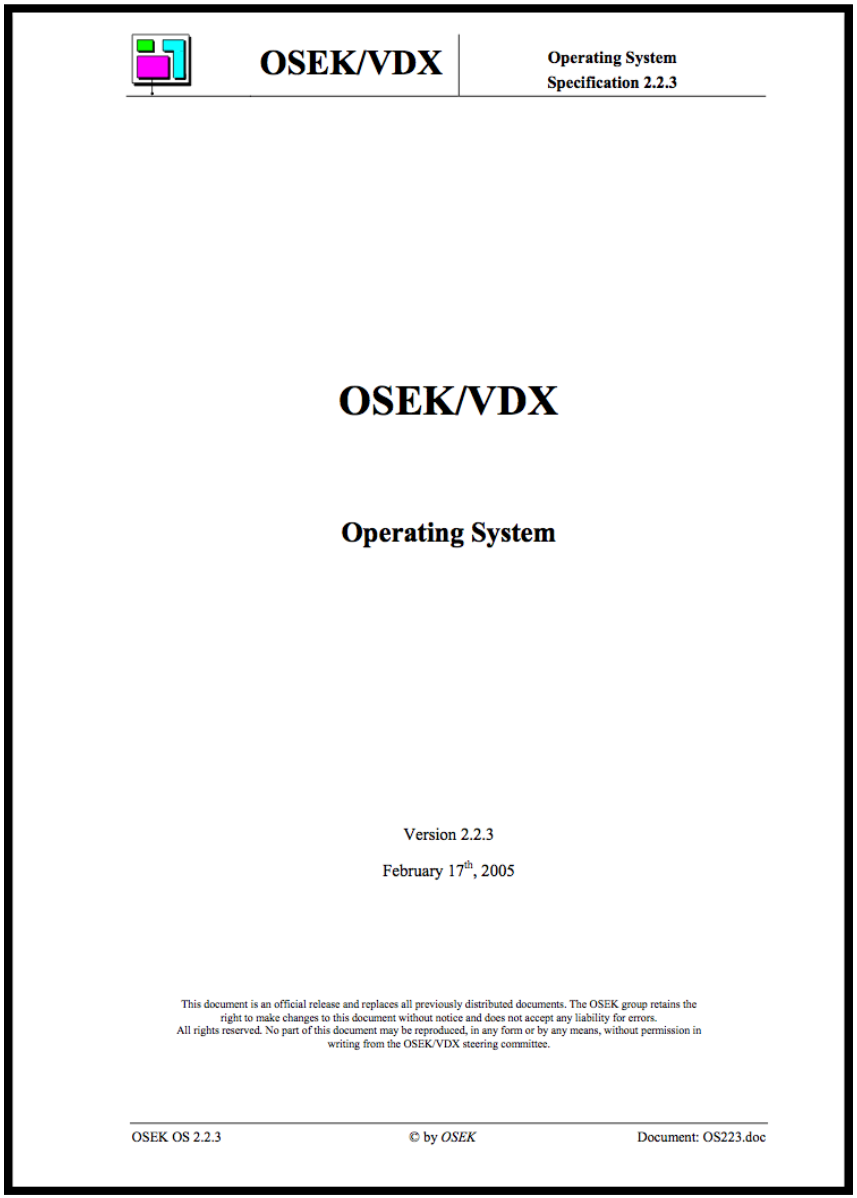
- No, it's just an **OS specification**, not a product
- We have lots of different commercial products as well as even open source OSEK/VDX implementations

You can download the specification at http://www.osek-vdx.org

OSEK/VDX

- OSEK
  - **O**ffene **S**ysteme und deren Schnittstellen für die **E**lektronik in **K**raftfahrzeugen
  - Translated as "Open Systems and their Interfaces for the Electronics in Motor Vehicles"
  - German standard
- VDX
  - **V**ehicle **D**istributed e**X**ecutive
  - French standard
- OSEK/VDX
  - A merge of OSEK and VDX

OSEK/VDX Specification



OSEK/VDX

Operating System
Specification 2.2.3

OSEK/VDX

Operating System

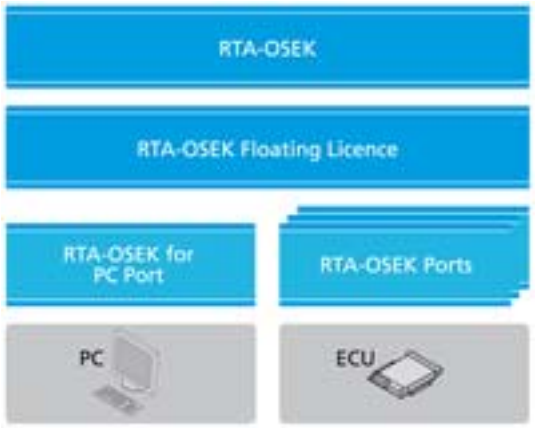
Version 2.2.3

February 17th, 2005

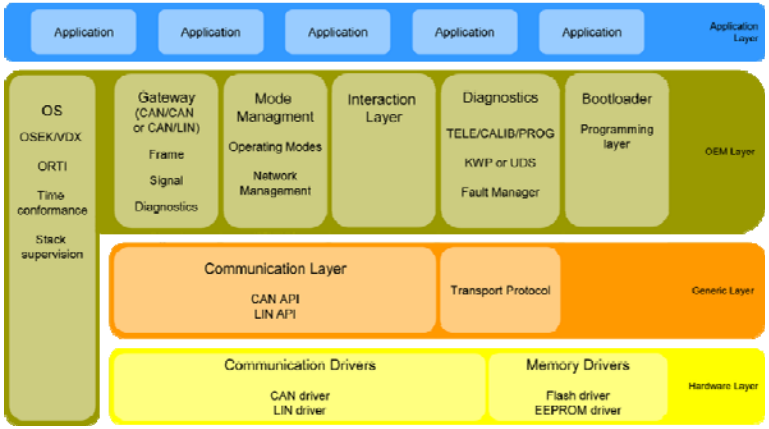
OSEK OS Specification

Version 2.2.3
Date: 2005-02-17

OSEK/VDX implementations



ETAS RTA-OESK



EB TRESOS OSEKCORE

Commercial

Opensource

osCAN

Real Time Operating System Based on the OSEK/VDX™ Standard

Vector osCAN

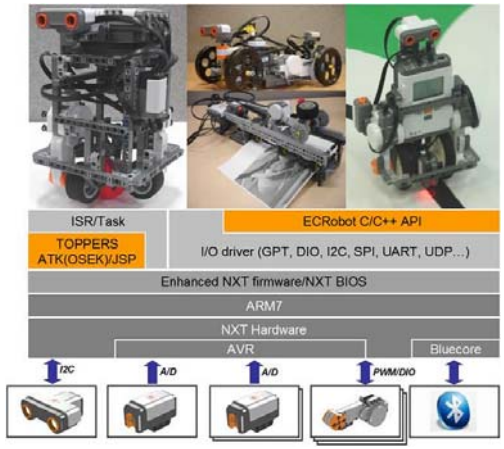
Trampoline OpenSource RTOS project

Trampoline OSEK

ERIKA ENTERPRISE

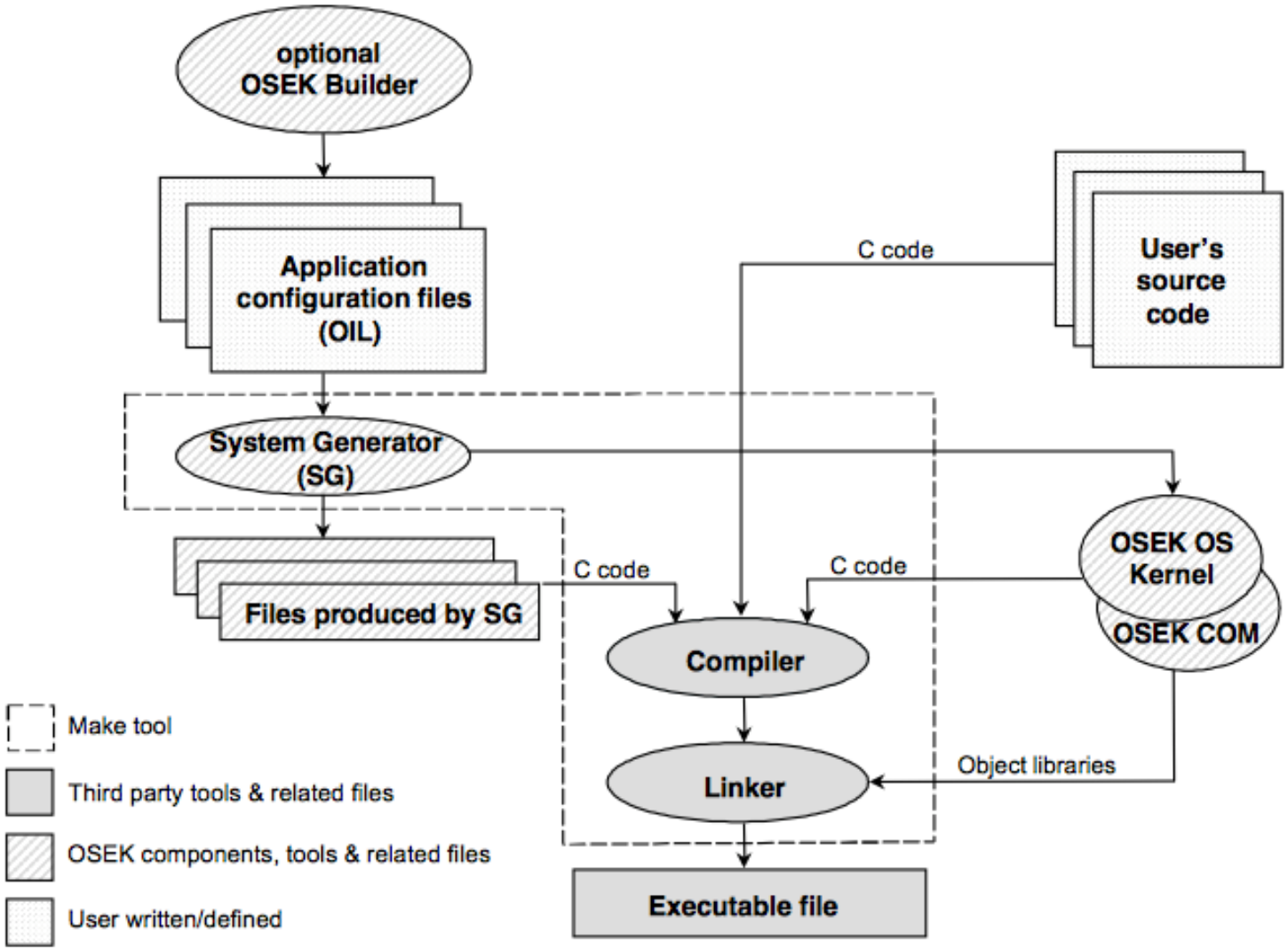
OSEK/VDX Compliant

ERIKA Enterprise



nxtOSEK

OSEK/VDX Build Process



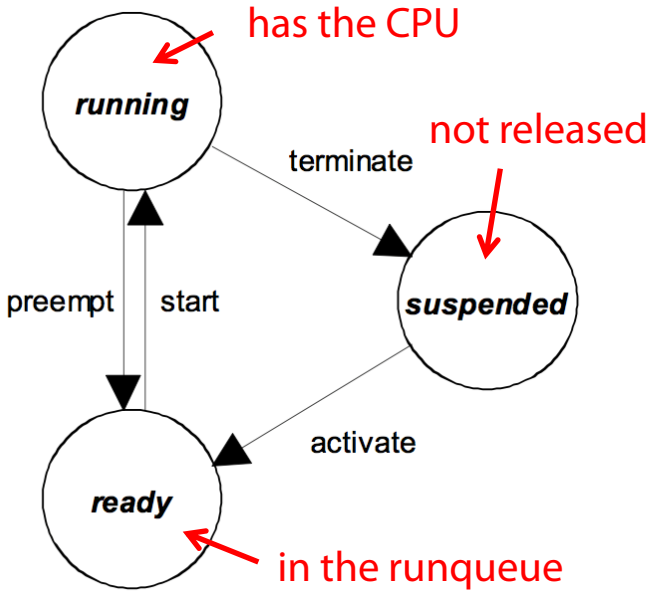
Two Task Types

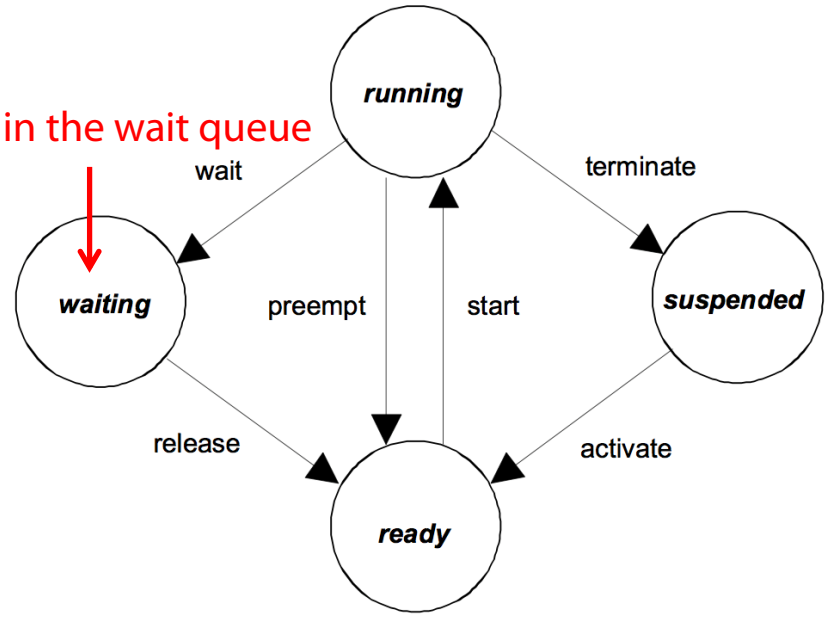
- Basic Task
  - Waiting state not allowed

```
TASK(myTask)
{
    ...
    TerminateTask();
}
```

- Extended Task
  - Waiting state allowed

```
TASK(myTask)
{
    ...
    WaitEvent(event);
    ...
    TerminateTask();
}
```

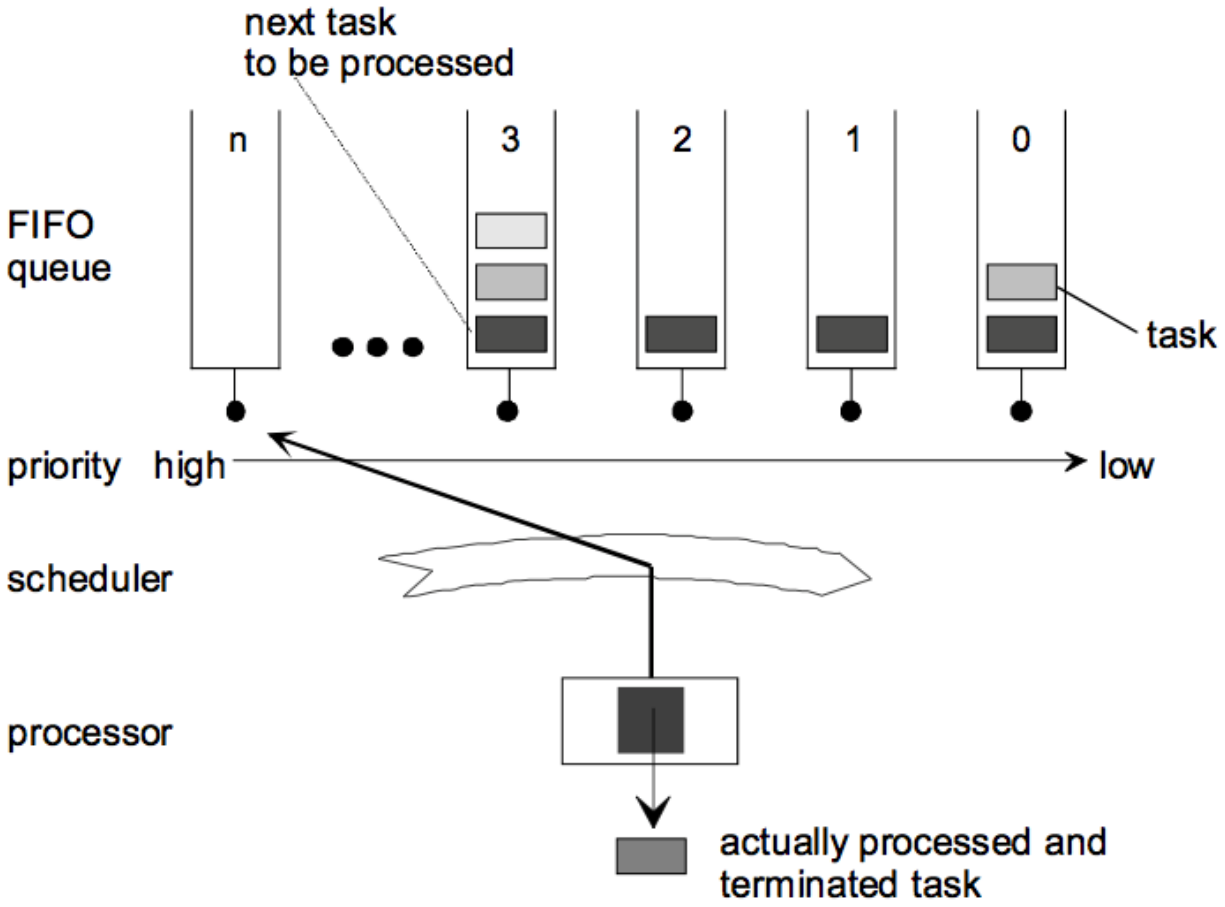




Scheduling (1/2)

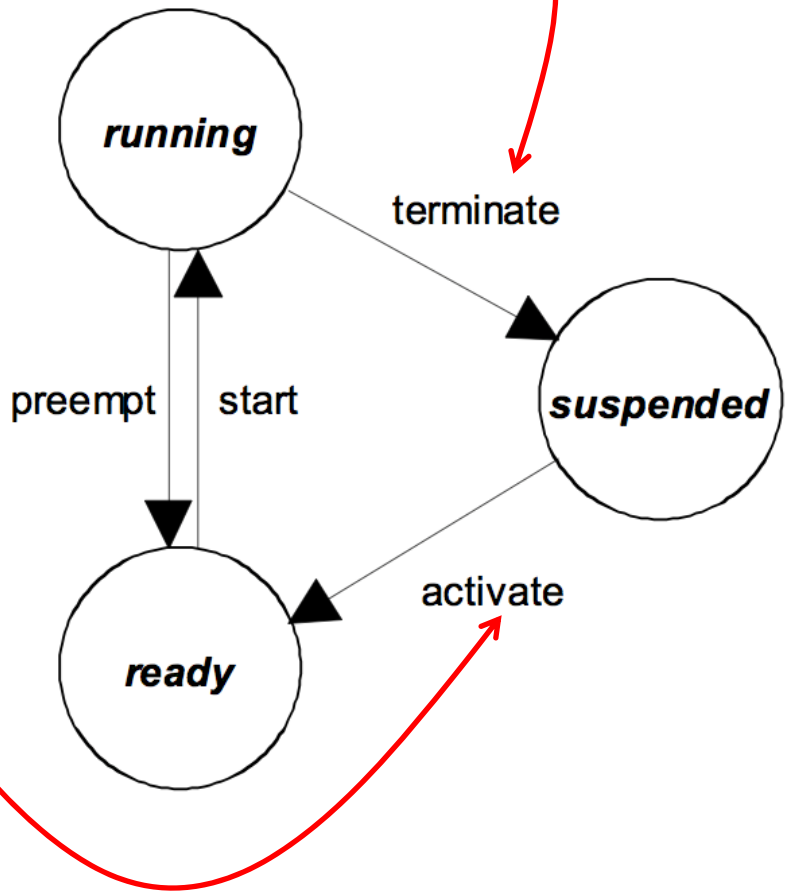
- Online fixed-priority scheduling
  - RM or not
- Static configuration
  - Scheduling parameters should be fixed offline
- Tasks can be either preemptable or not
  - It should be also configured offline
- Multiple tasks can exist with the same priority
  - FIFO policy for the same priority case

Scheduling (2/2)

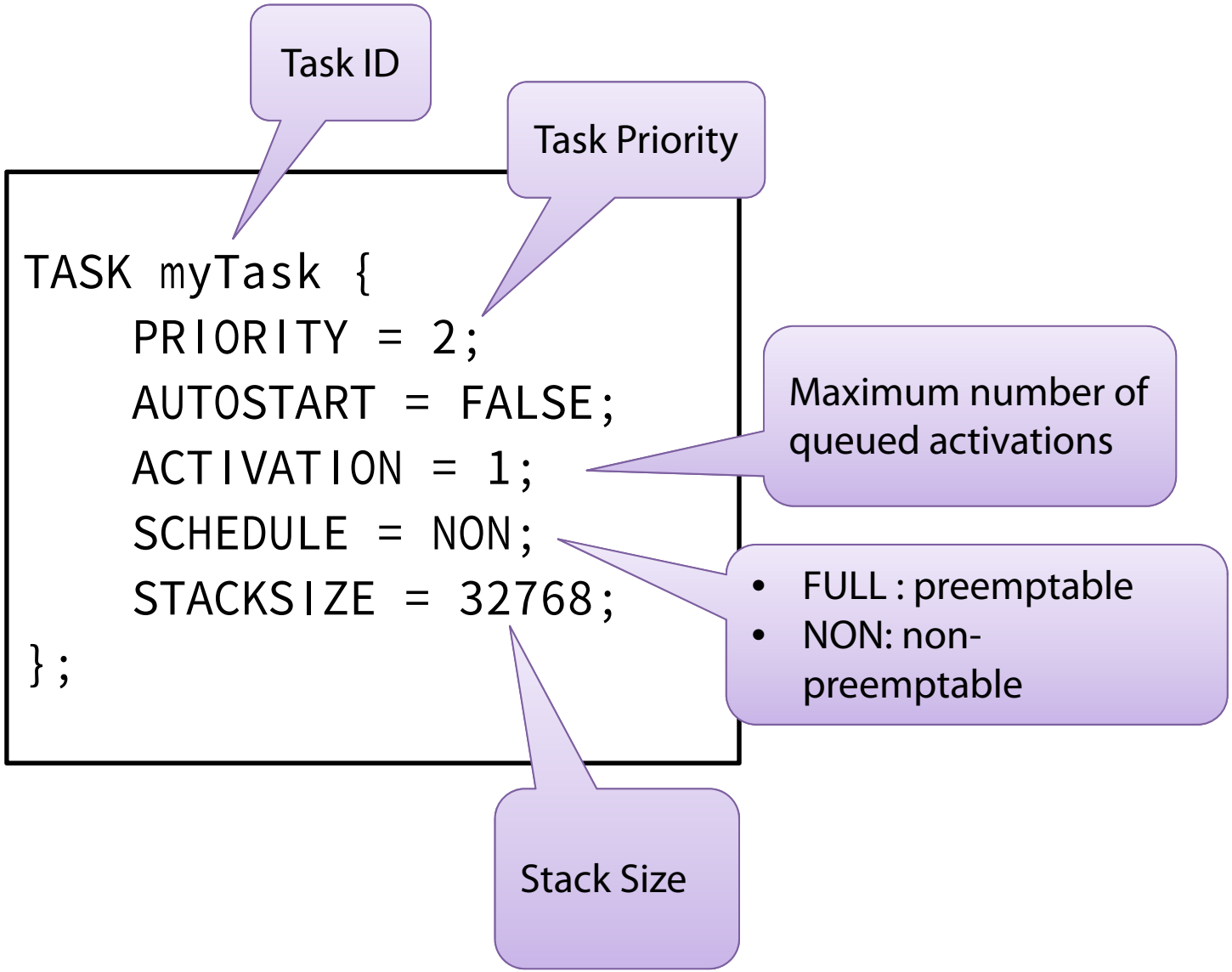


Task Management APIs

- TerminateTask
- ActivateTask
- ChainTask
  - Terminate me and activate some other task at the same time

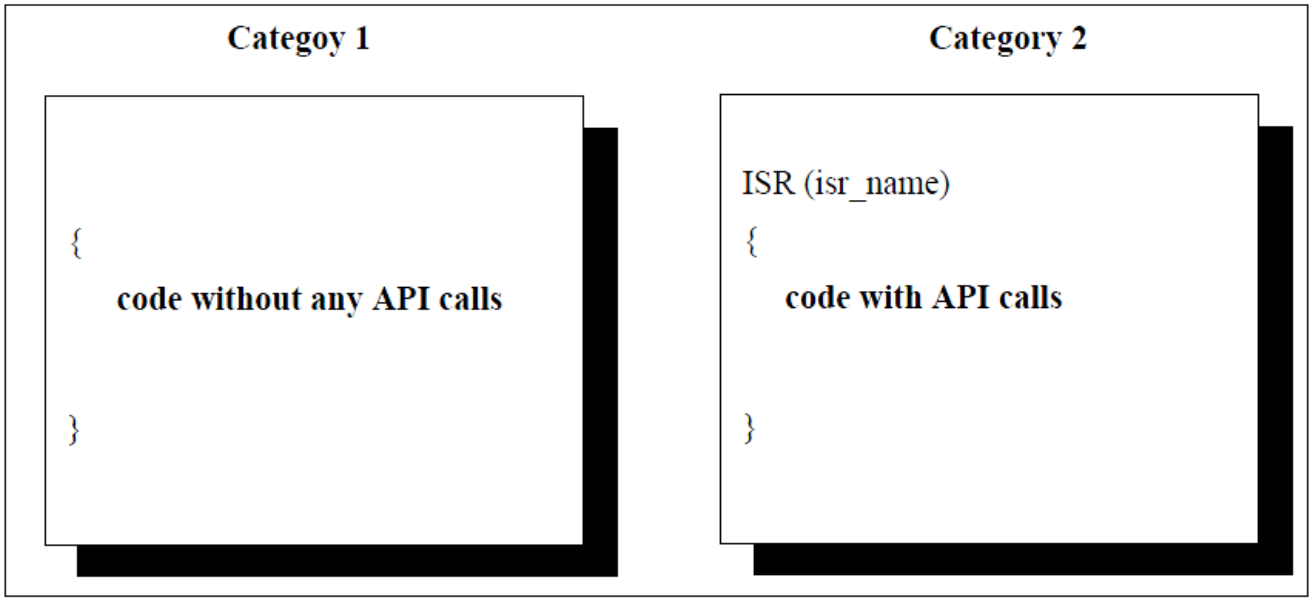


OIL Description of a Task



Interrupt Handling

- ISR (Interrupt Service Routine)
  - A function which is triggered by an interrupt
- Two ISR categories
  - ISR category 1: do not use any OS service
  - ISR category 2: may call OS services

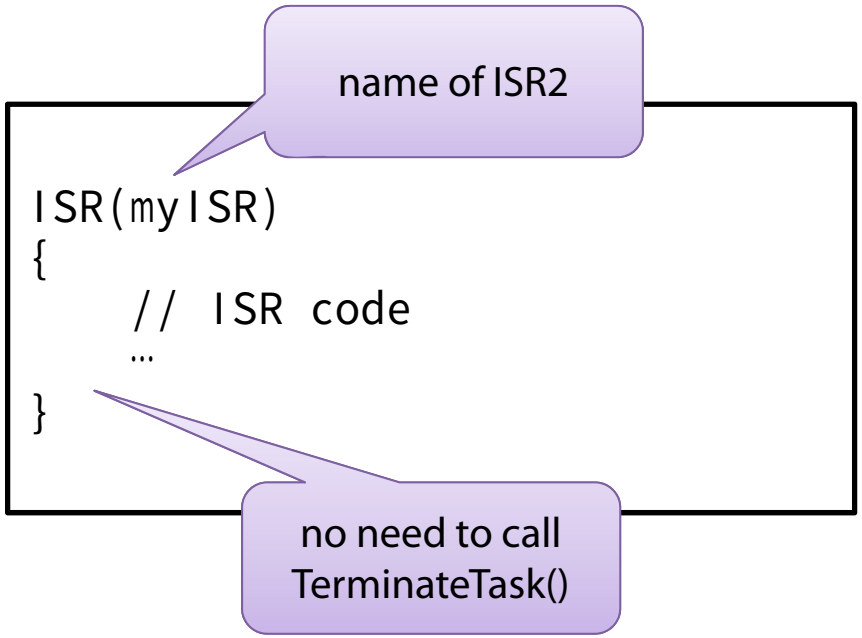


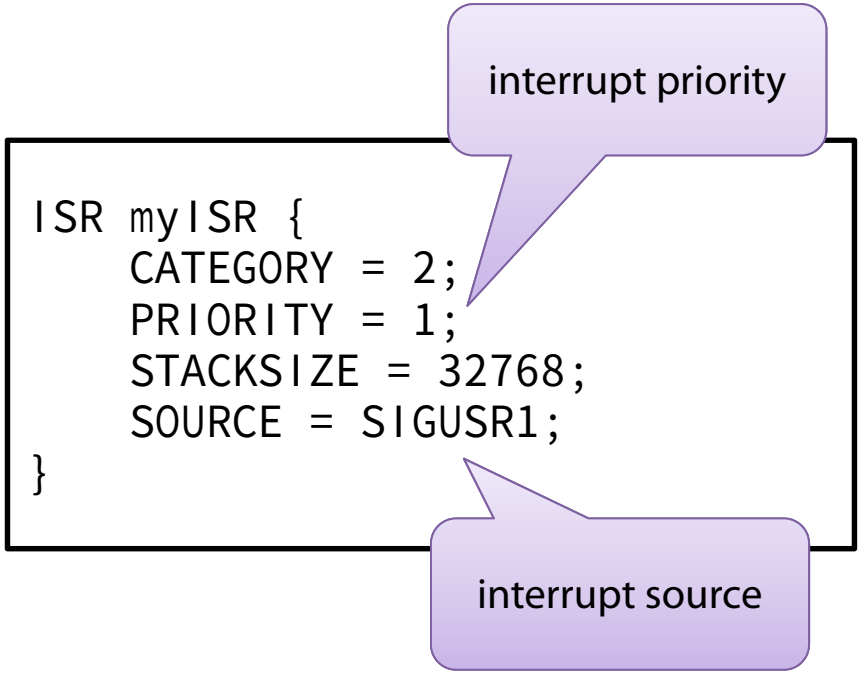
ISR1

- Low overhead / fast
- No system call allowed inside ISR1
- Ignored by the operating system
- No OIL description needed
  - Basically, ISR1 is not portable across different microprocessors

ISR2

- More overhead / slower than ISR1
- System calls allowed (ActivateTask, …)
- Operating system awares of ISR2
- OIL description needed



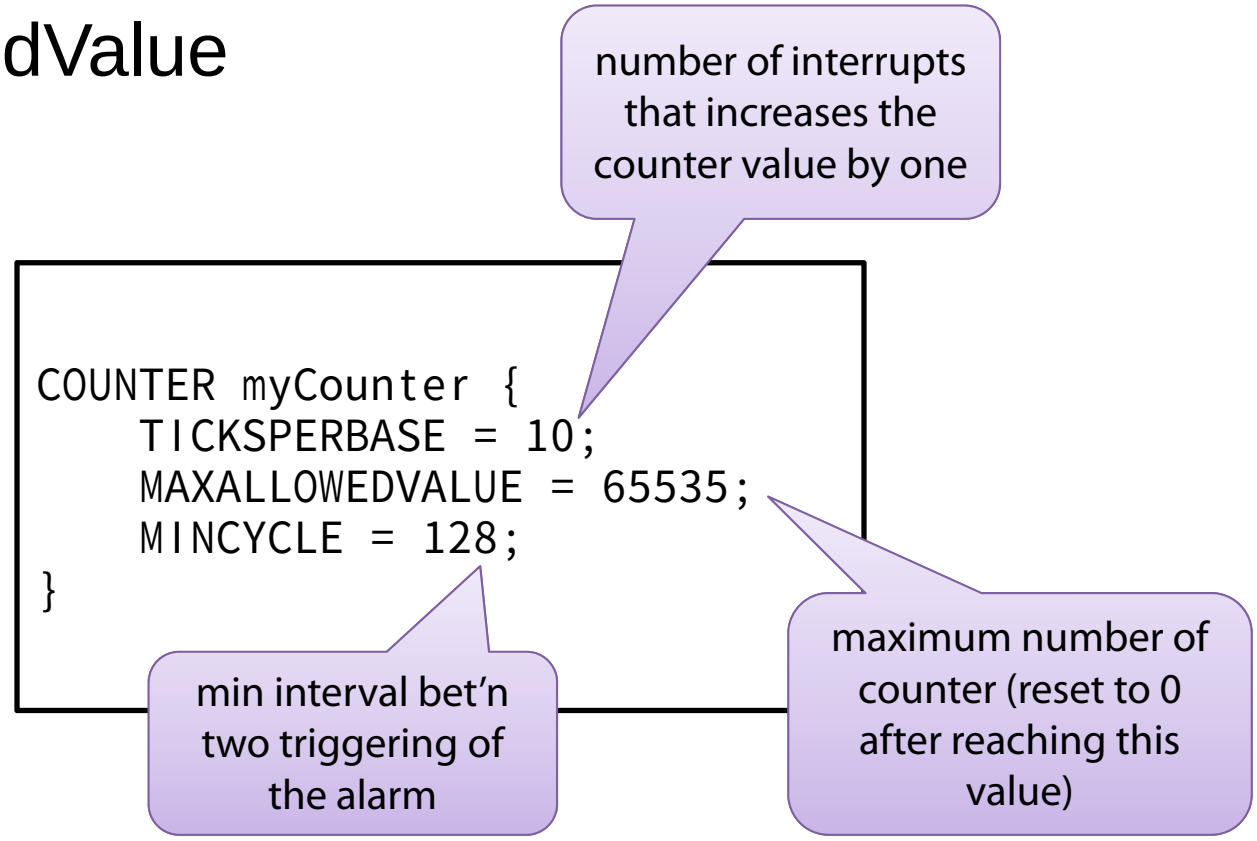


Alarm and Counter

- Perform an action after a number of clock ticks
- Actions
  - Set an event
  - Activate a task
  - Function call (a callback function)

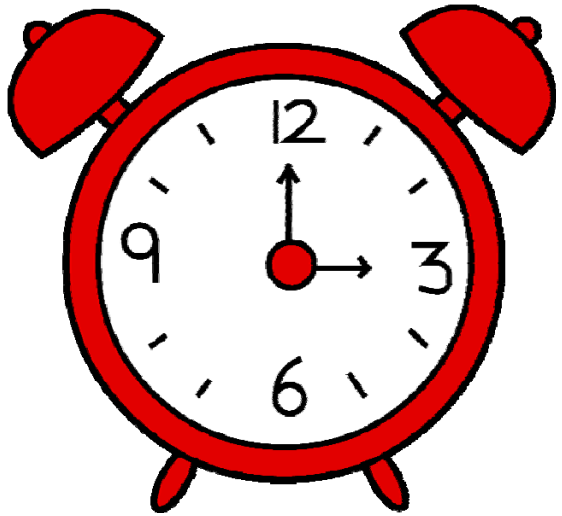
Counter

- Abstraction of hardware clock tick
- Three relavant parameters
  - TicksPerBase
  - MaxAllowedValue
  - MinCycle

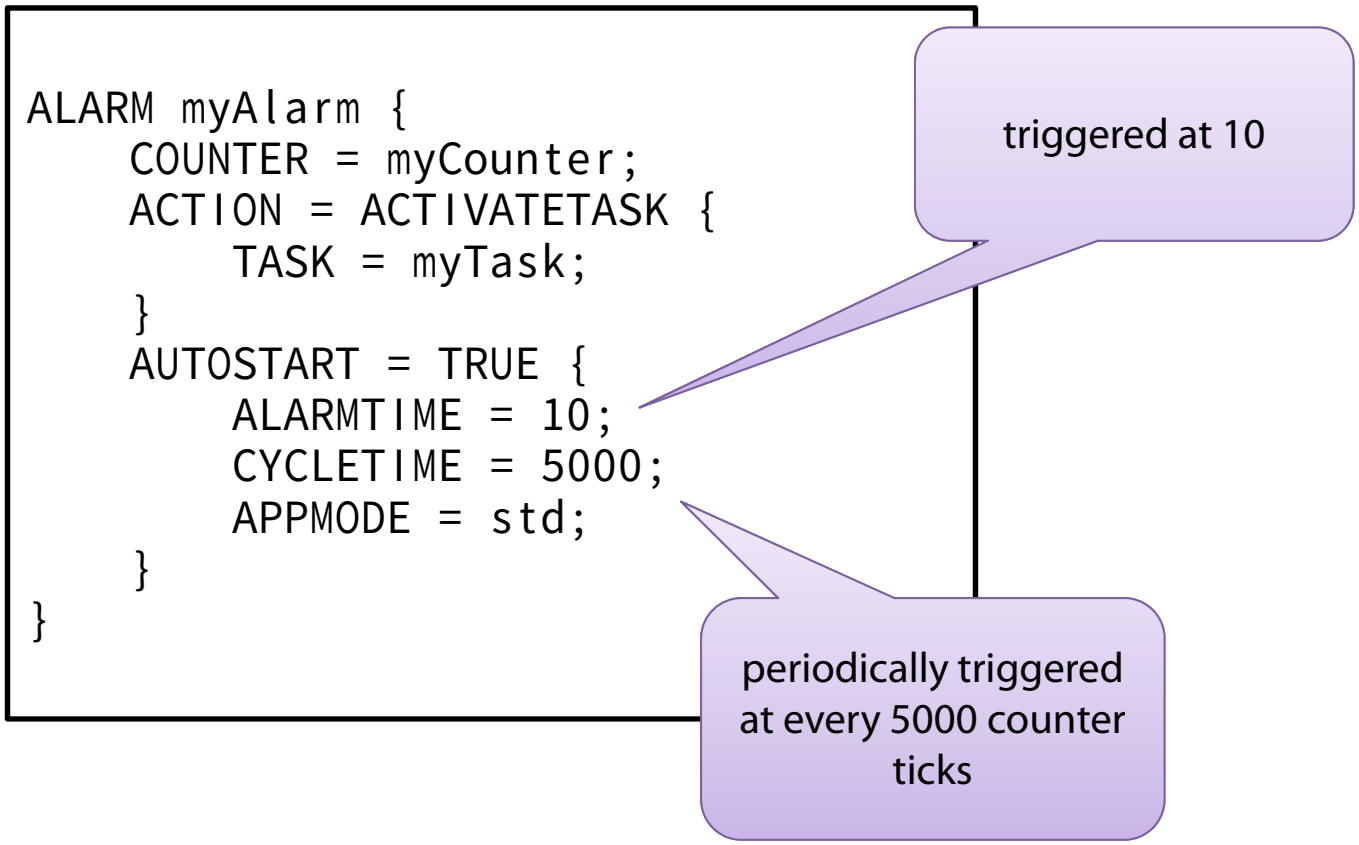


Alarm

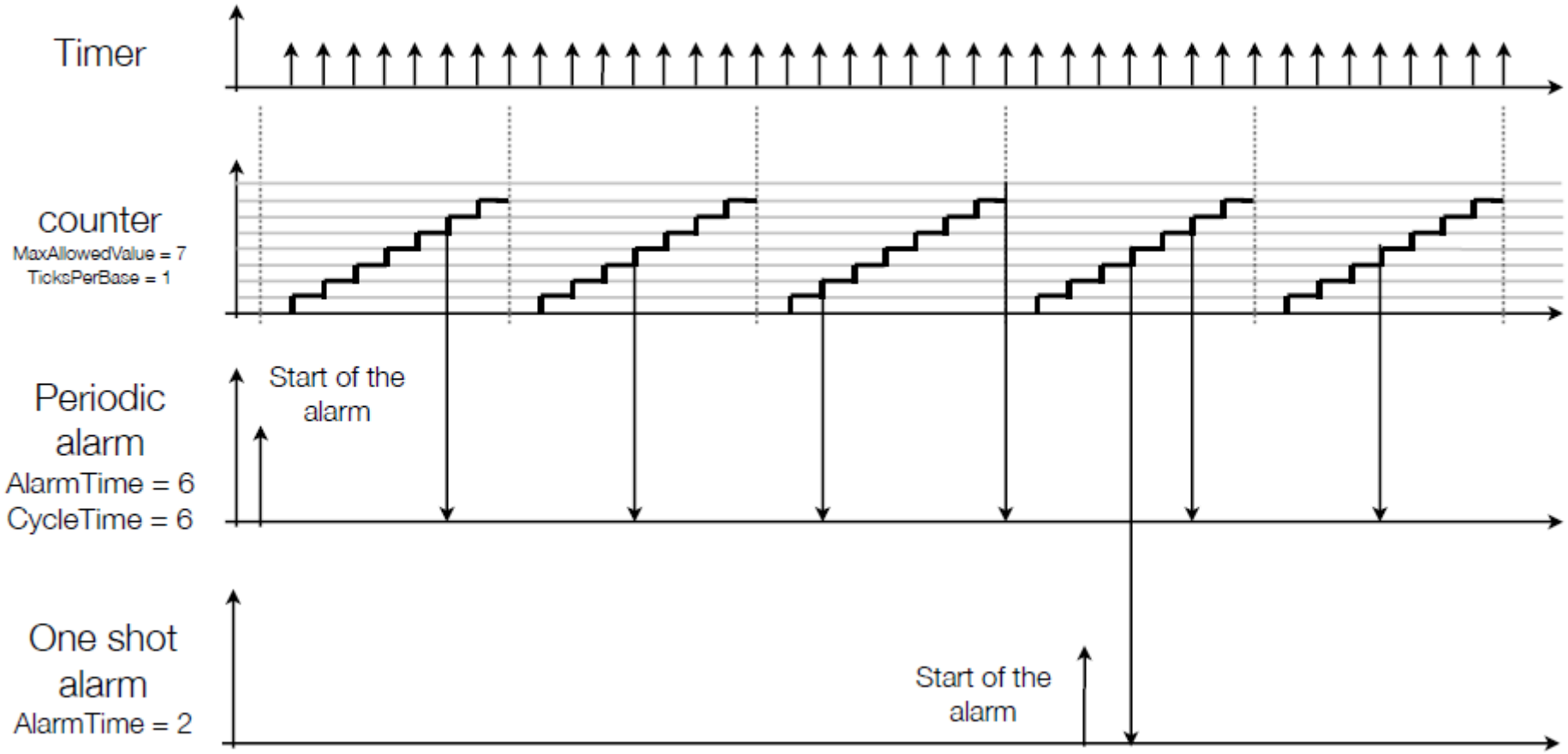
- Alarm is connected to a counter to perform an action
- When a counter reaches a value of the alarm, the pre-defined actions are performed
  - Set an event
  - Activate a task
  - Function call



OIL Description of Alarm



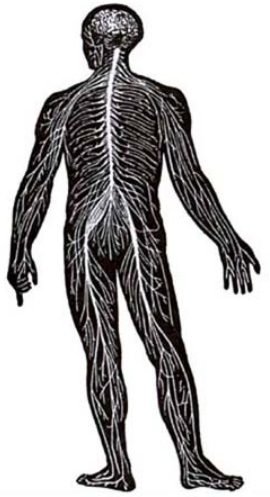
Counter and Alarm



In-Vehicle Networks

Like a human central nervous system that connects a brain, sensory organs, and locomotive organs

In-vehicle network



Central nervous system

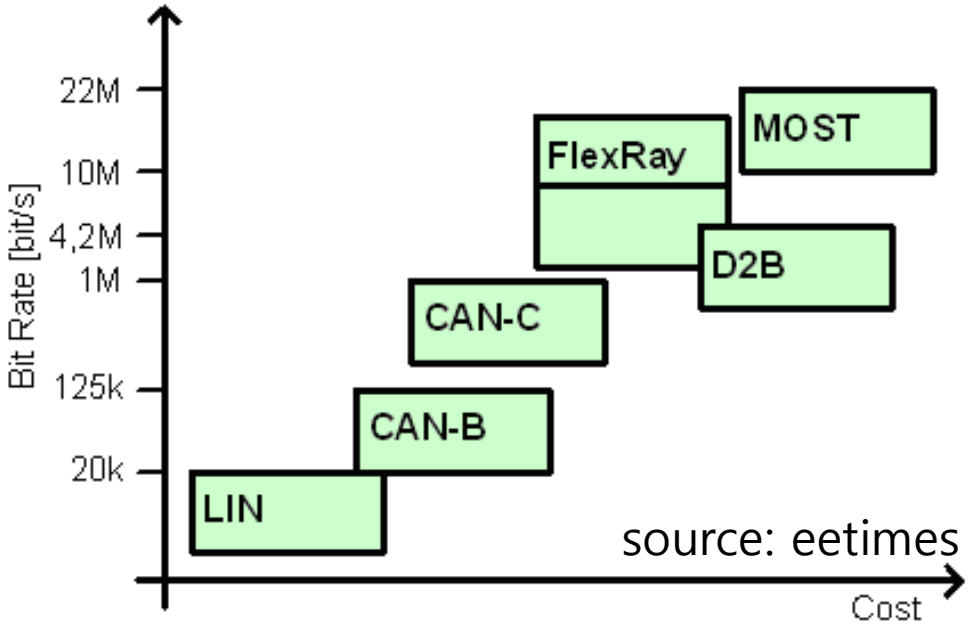
Delivers signals (e.g., engine RPM, vehicle speed) throughout ECUs, sensors, and actuators in a vehicle

In-vehicle network standards : CAN, LIN, FlexRay, MOST, ...

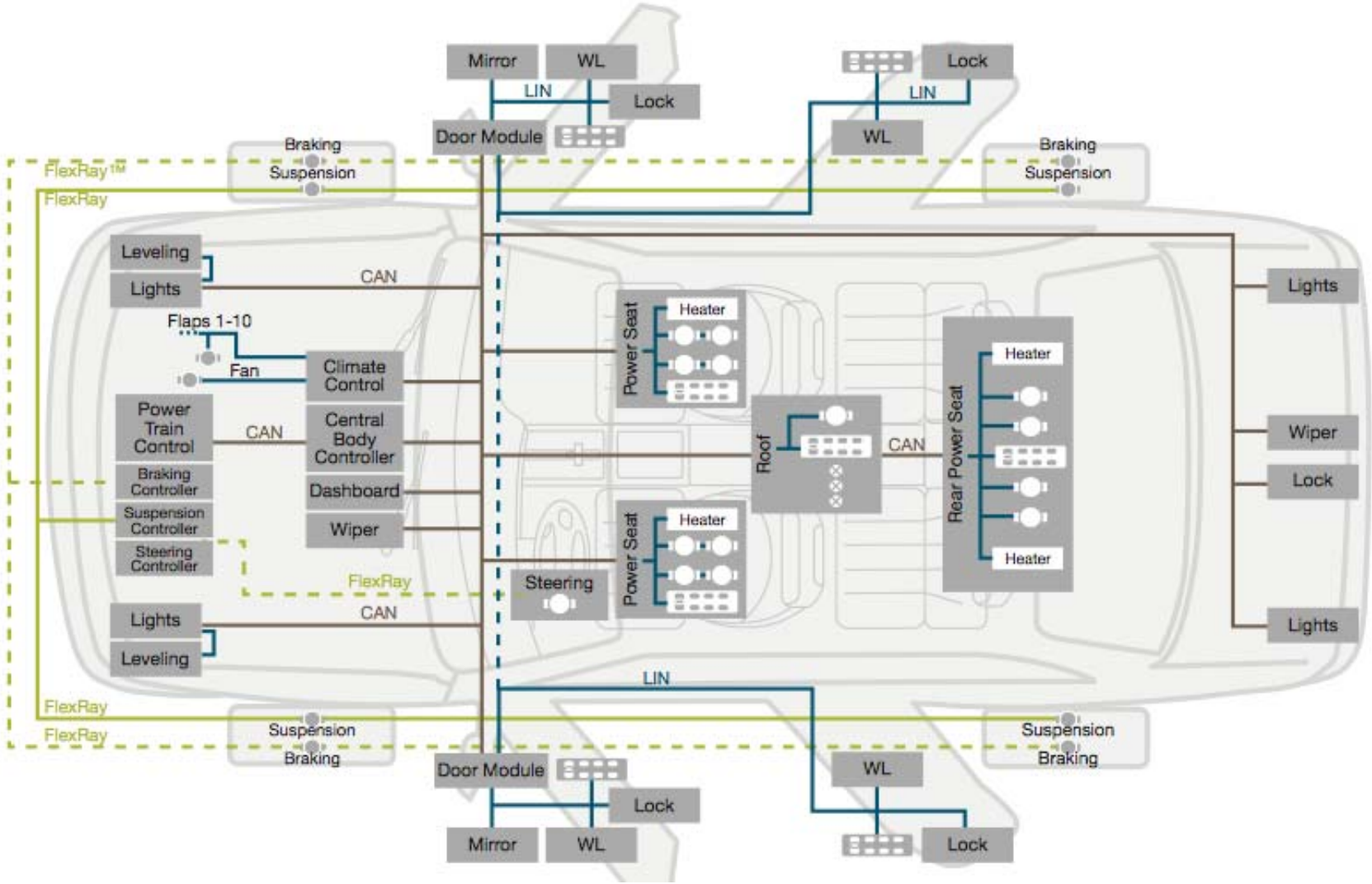
Today's most widely used in-vehicle network standard

In-vehicle networks

- Various in-vehicle networks
  - LIN : slow, low cost
  - CAN : most-widely used
  - FlexRay : time-triggered
  - MOST : for multimedia
  - …



In-vehicle network example



Source: Freescale

Controller Area Network (CAN)

- The standard serial data bus in the automotive industry developed by Bosch in the 1980s
  - http://esd.cs.ucr.edu/webres/can20.pdf
  - No master node
  - Bit rate depends on bus length
    - Up-to 1 Mbit/s
  - Strong to noise
  - Message ID based arbitration
    - No node address required
    - Message ID := Priority
  - Event-driven
  - Non-preemptive priority-driven scheduling

Bit rate	Bus length
1 Mbit/s	25 m
800 kbit/s	50 m
500 kbit/s	**100 m**
250 kbit/s	250 m
125 kbit/s	500 m
50 kbit/s	1000 m
20 kbit/s	2500 m
10 kbit/s	5000 m

CAN Physical Layer

- Two-wire differential bus

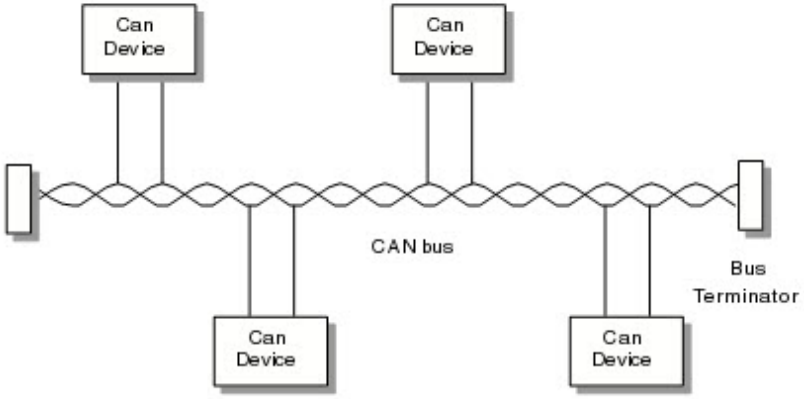
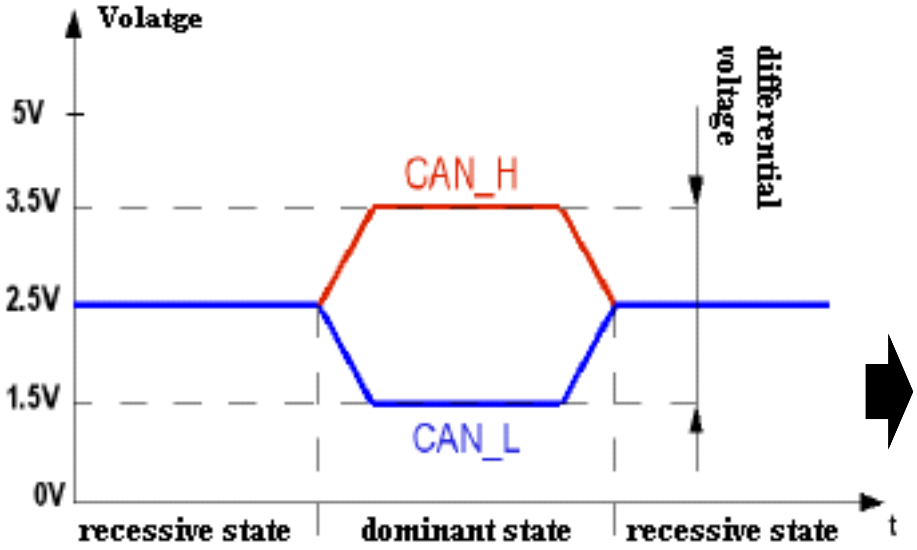
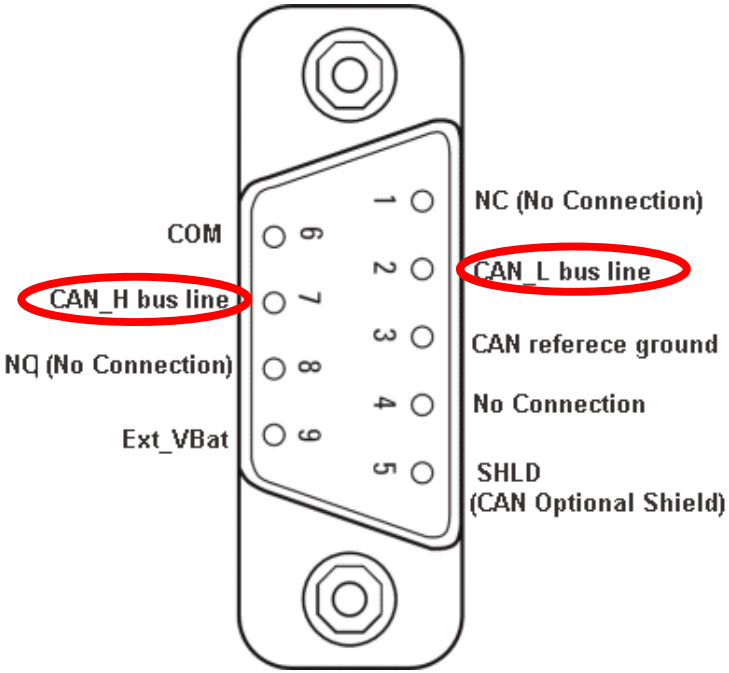


Figure 12-1: A CAN bus

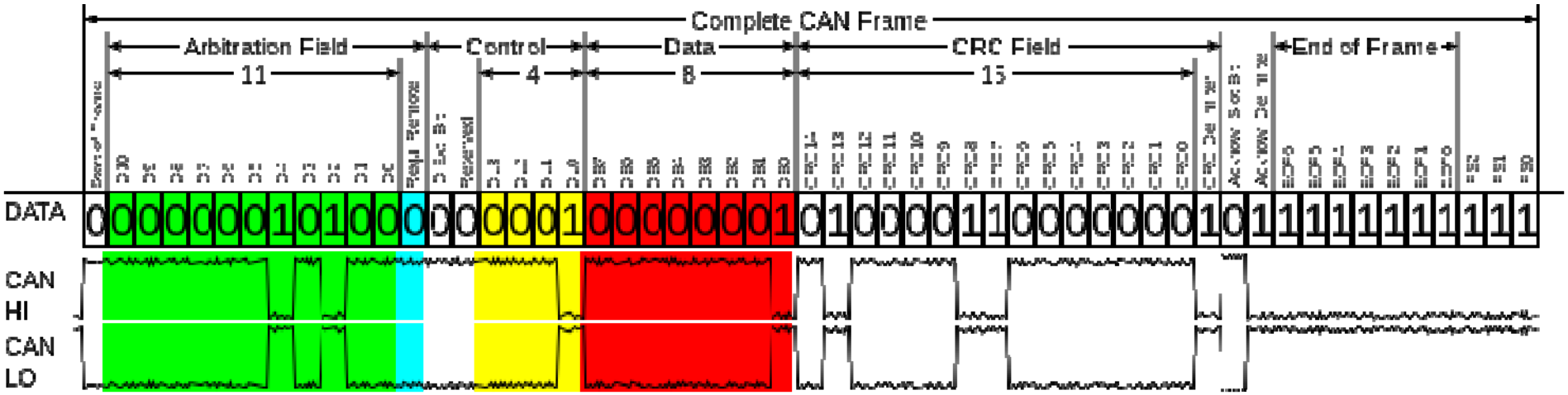




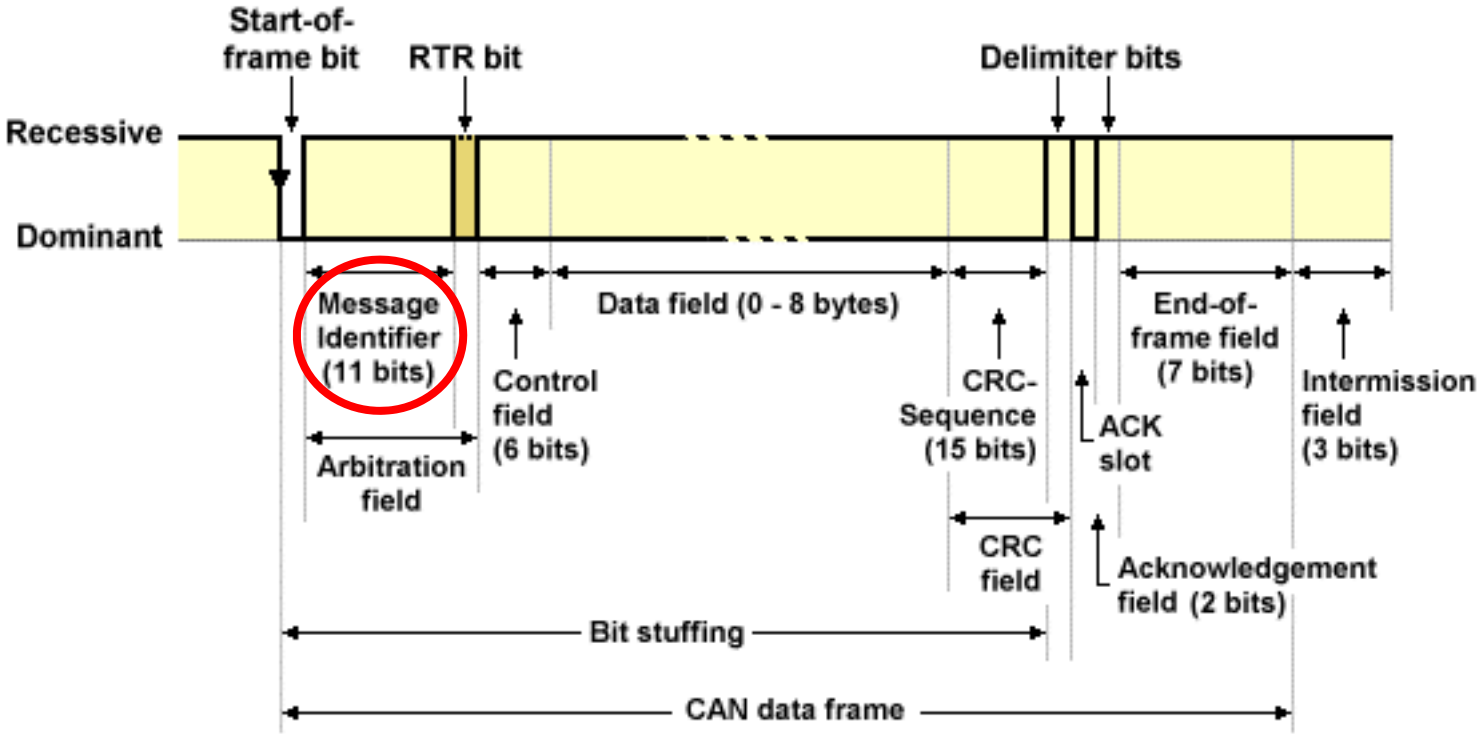
Strong to external noise

Strong to external noise

CAN Transmission Example



CAN Data Frame Format

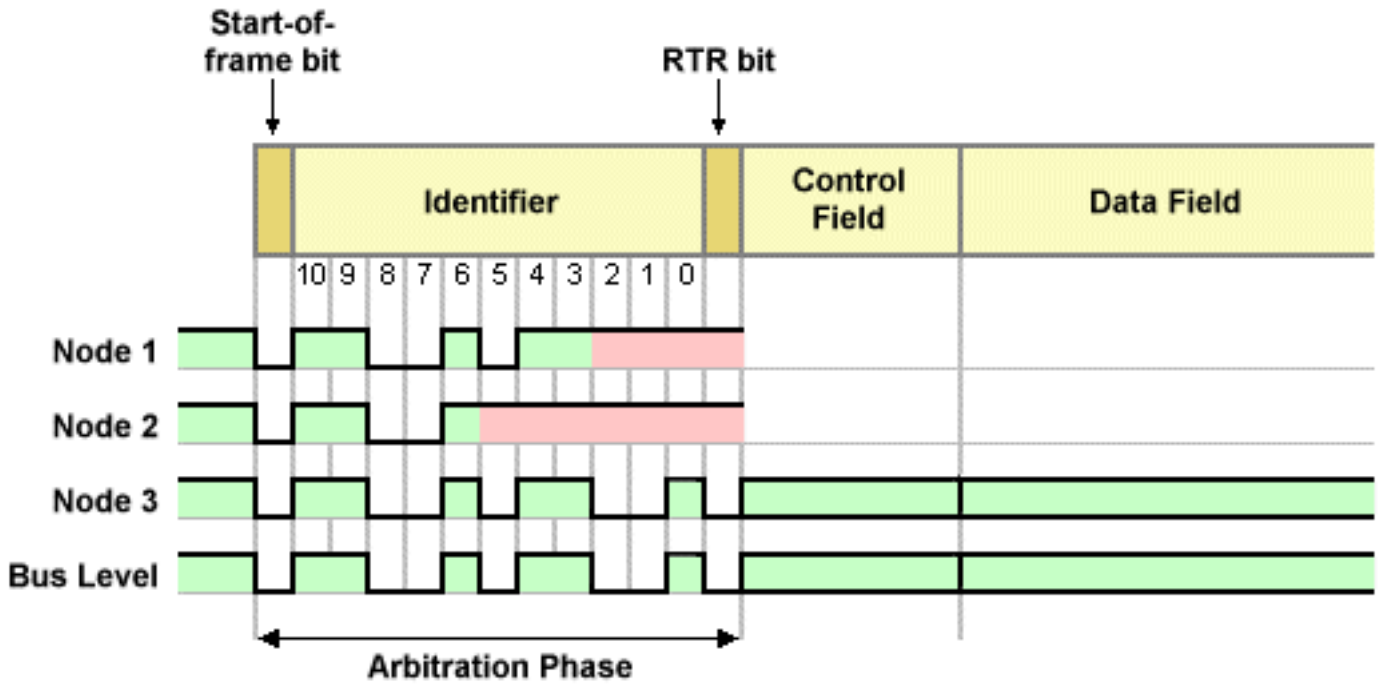


Message ID plays as the message's priority. How?

http://www.technologyuk.net/telecommunications/industrial_networks/can.shtml

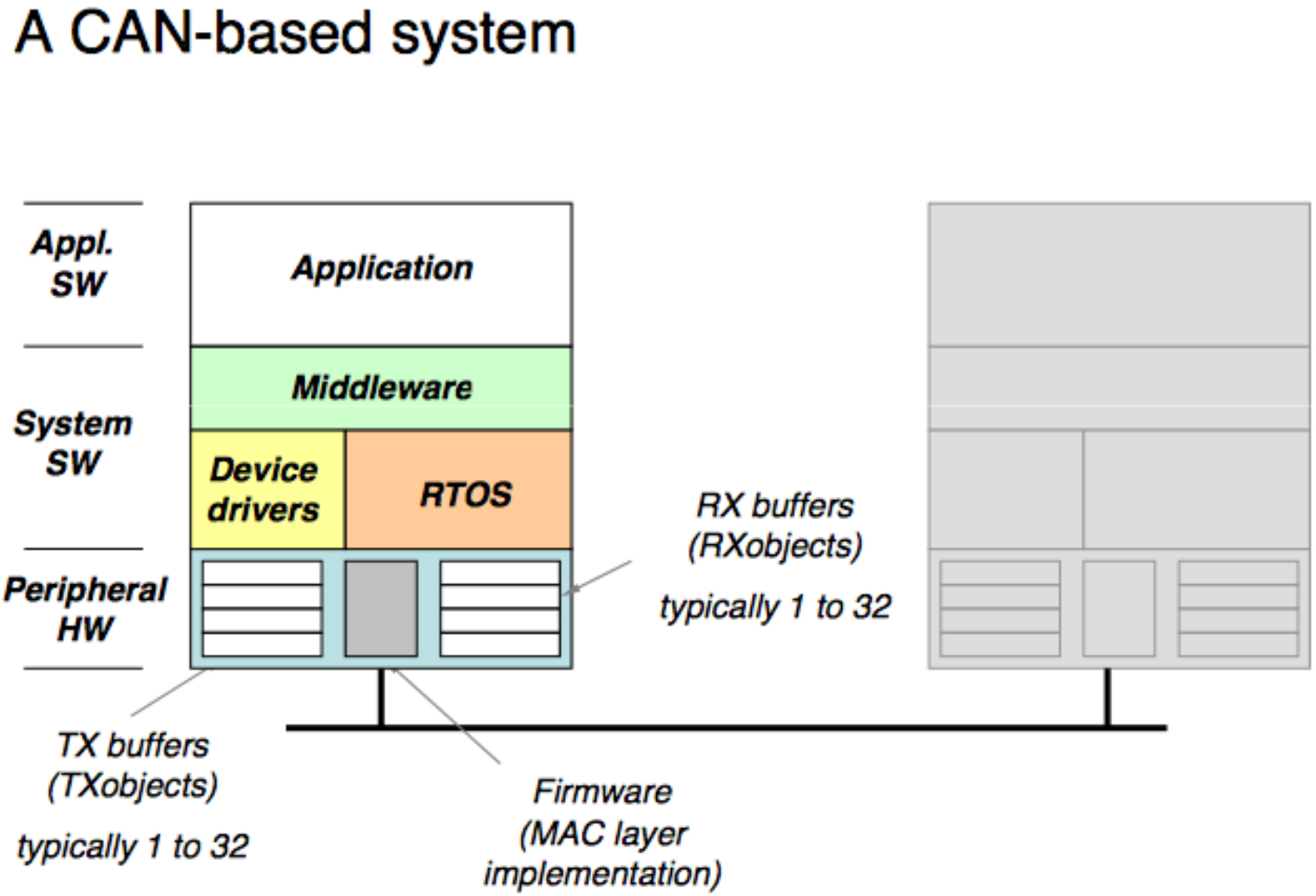
CAN Bus Arbitration

- Writes 1, but reads 0, then stop transmitting
- Lower message ID = higher priority



http://www.technologyuk.net/telecommunications/industrial_networks/can.shtml

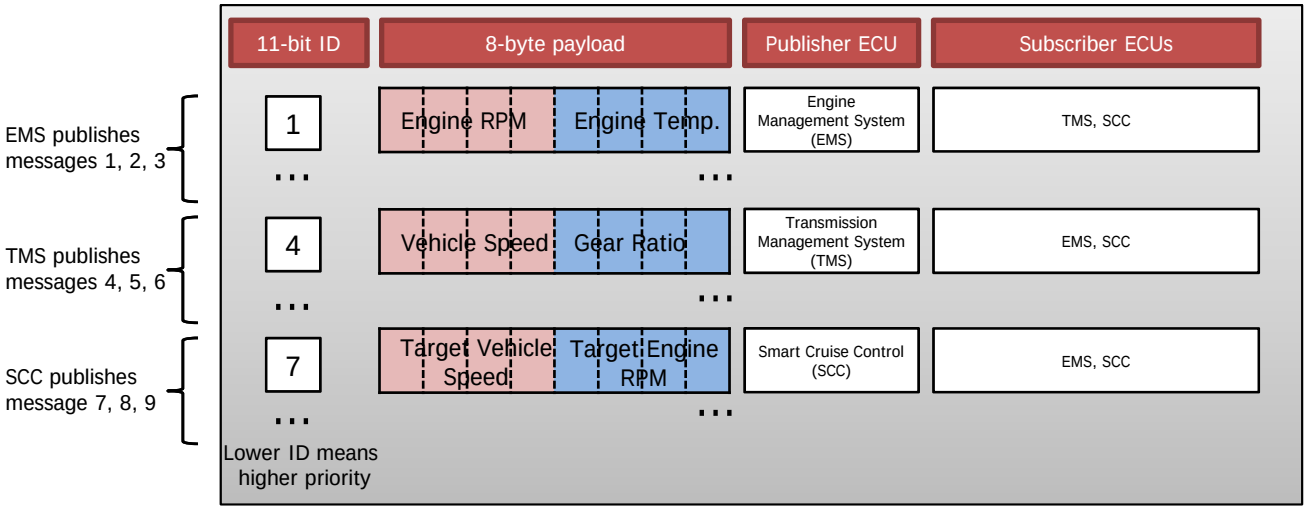
Typical CAN-based System

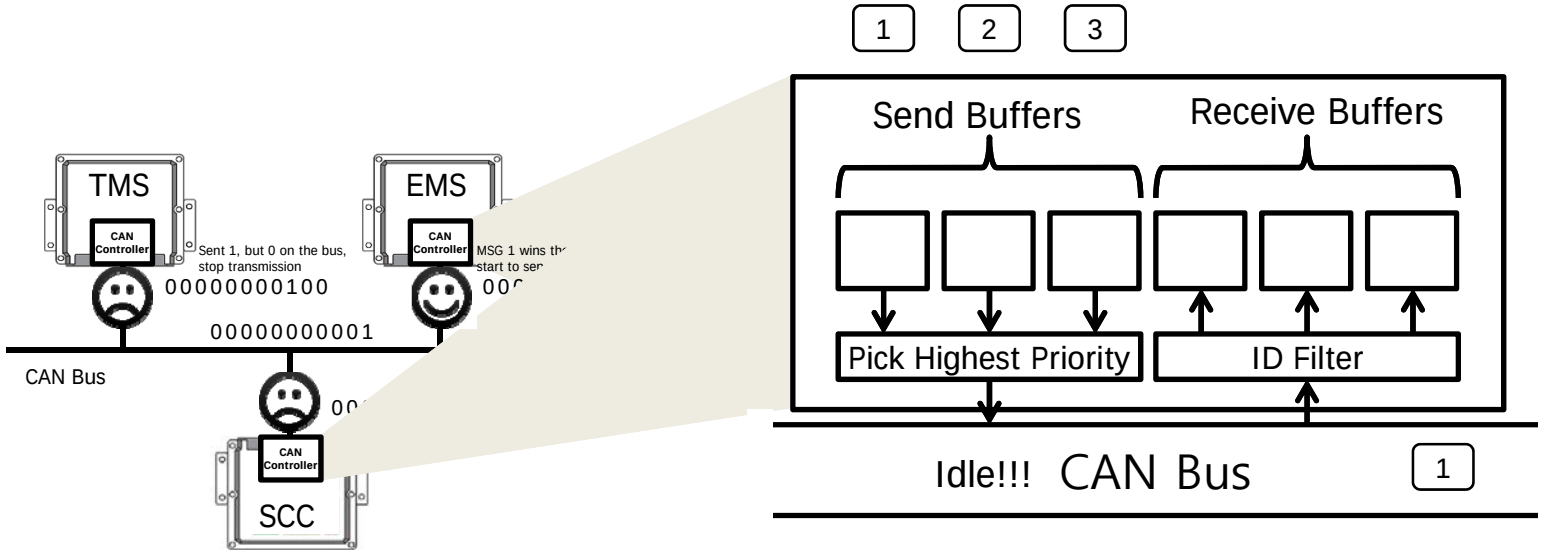


http://inst.eecs.berkeley.edu/~ee249/fa07/CANBus.pdf

CAN Arbitration Example

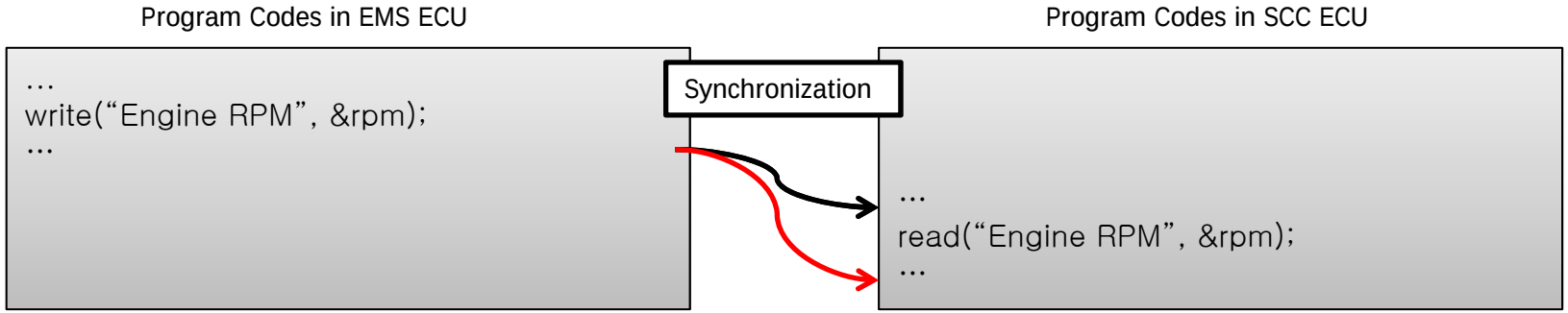
- A simple CAN bus message design



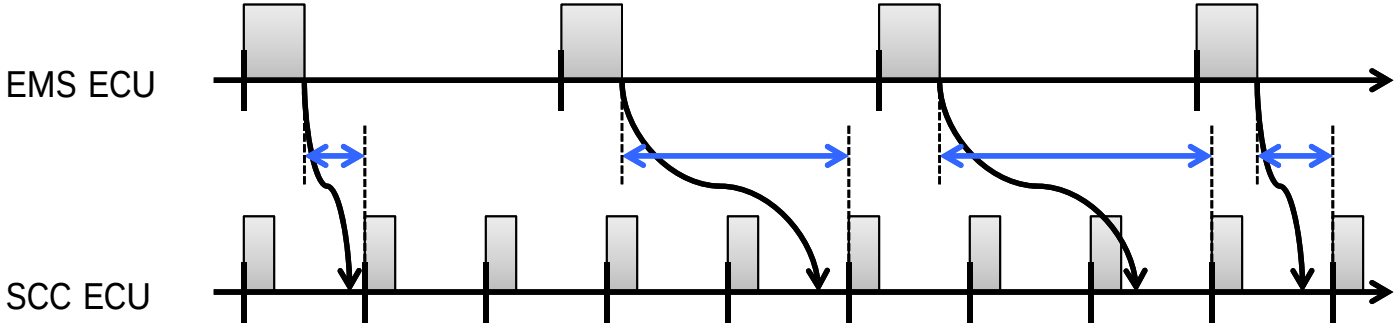


Programmer's Perspective

- Distributed key-value DB with synchronization
- The synchronization can be delayed due to the priority-based bus arbitration



- The effective synchronization delay varies depending on receiver's release pattern as well as CAN bus arbitration



Questions