

## Time Synchronization to FlexRay in OSEKtime and Autosar OS

### Schedule Tables to the FlexRay Global Time

## The OSEK OS Standard



## OSEKtime

- ❑ actual Version 1.00, 24.07.2001
- ❑ Schedule Table synchronization
- ❑ Deadline Monitoring

## Autosar OS

- ❑ actual Version 2.0.08, 30.1.2007
- ❑ Scalability Class 2

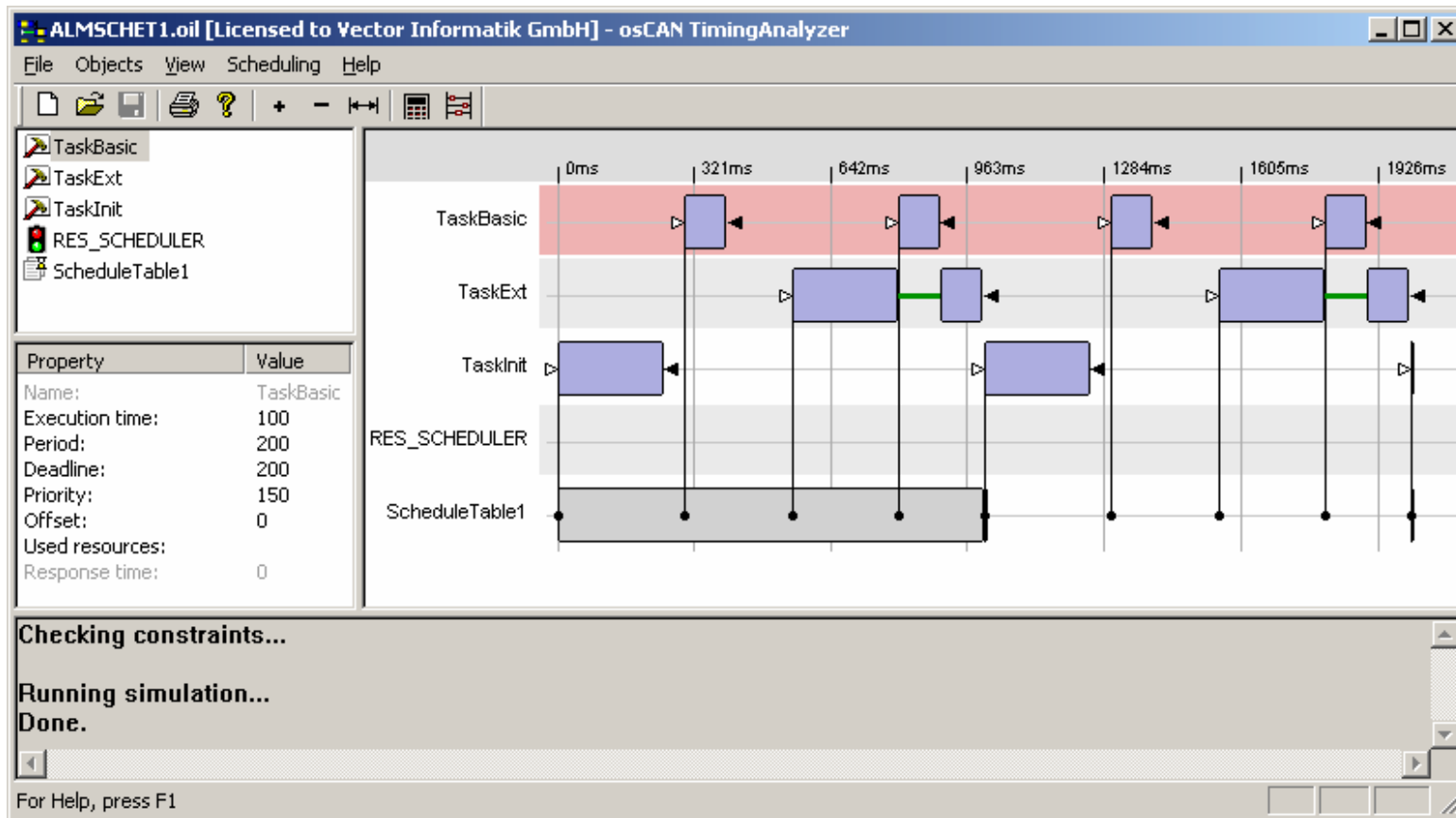
# Autosar Scalability Classes

Feature	SC1	SC2	SC3	SC4	Hardware requirements
OSEK OS	✓	✓	✓	✓	
Counter Interface	✓	✓	✓	✓	
Schedule Tables	✓	✓	✓	✓	
Stack Monitoring	✓	✓	✓	✓	
ProtectionHook		✓	✓	✓	
Timing Protection		✓		✓	timer(s) with high priority interrupt
Global Time /Synchronization Support		✓		✓	Global time source
Memory Protection			✓	✓	MPU
OS-Applications			✓	✓	
Service Protection			✓	✓	
CallTrustedFunction			✓	✓	(non-)privileged Modes

# Schedule Tables

## Attributes (statically defined)

- ❑ Executes periodically or once
- ❑ Length
- ❑ Expiry points activate tasks or set events



What can be done with Schedule Tables?

API

StartScheduleTableRel(...)

StartScheduleTableAbs(...)

StartScheduleTableSynchron (...)

NextScheduleTable (...)

StopScheduleTable(...)

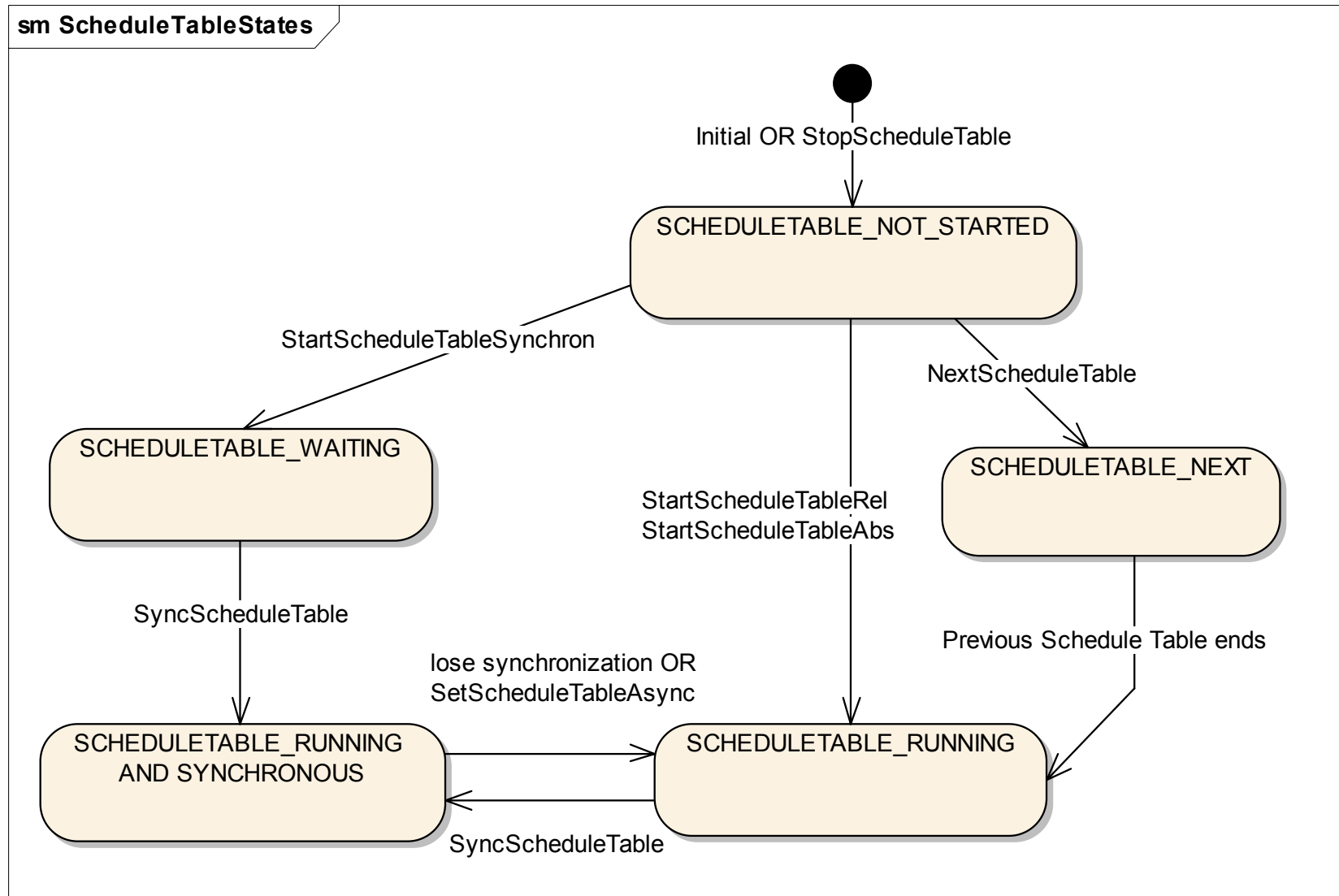
SyncScheduleTable(...)

SetScheduleTableAsync(...)

GetScheduleTableStatus(...)

...and as Autostart!

## States of a Schedule Table



## Possibilities

1. Direct driven from synchronized hardware  
e.g. by Flexray Interface

```
FrIf_SetAbsoluteTimer(FrIf_AbsTimerIdx, Cycle, MacrotickOffset )  
FrIf_SetRelativeTimer(FrIf_RelTimerIdx, MacrotickOffse)  
FrIf_EnableRelativeTimerIRQ(FrIf_RelTimerIdx)
```

Remark: No execution (no interrupts) if no communication

2. By using OS functionality

```
StatusType SyncScheduleTable(ScheduleTableID, GlobalTime)
```

## Possibilities (other view)

### 1. Hard synchronization

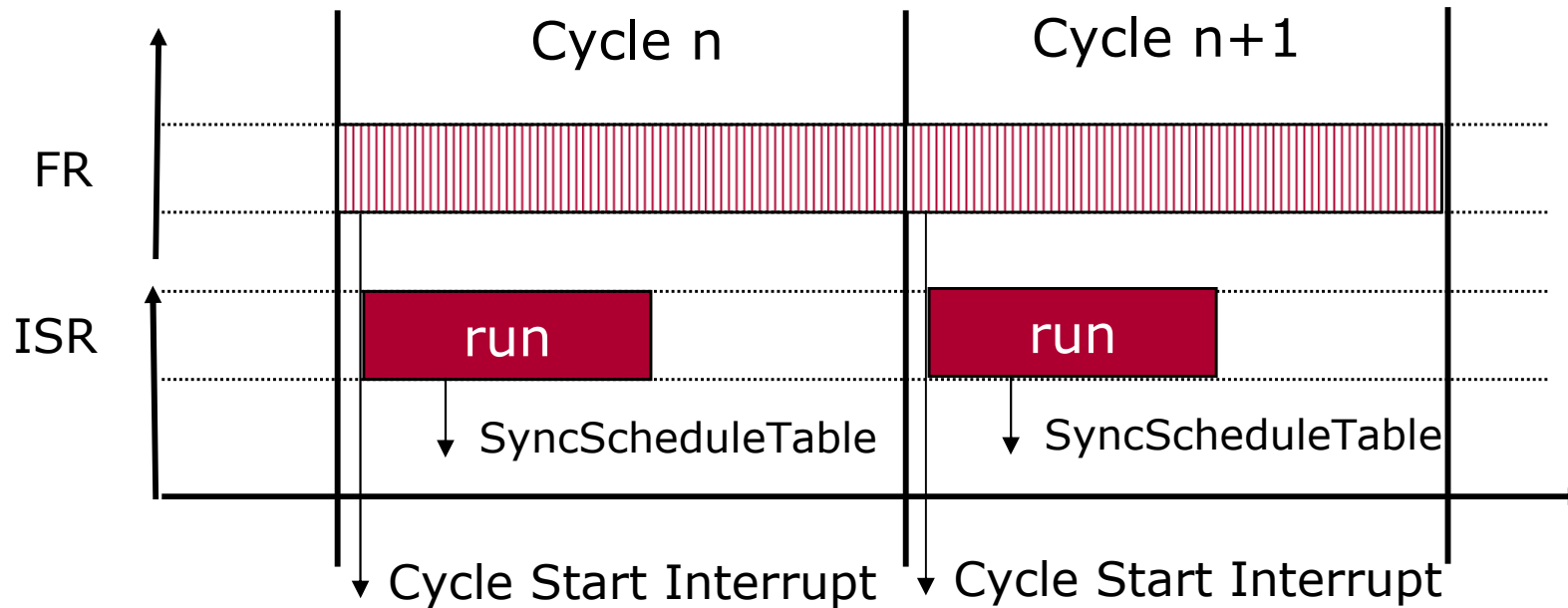
- ❑ Schedule Table can start immediately
- ❑ Schedule Table can wait for synchronization
- ❑ Synchronization done at the end of ST by jump

### 2. Smooth synchronization

- ❑ Schedule Table can start immediately
- ❑ Schedule Table can wait for synchronization
- ❑ Synchronization done at **each** expiry point by steps
- ❑ If (time deviation < PRECISION) then change state

# Synchronization to Global Time

## Synchronization to Flexray



Macrotick counter: 16 bit

Cycle counter: 8 bit

## Synchronization to Flexray

### Length of Schedule Table:

- exactly equal to one cycle (3~5ms)
- exactly equal to one communication cycle (1..64 -> 3..320ms)

### Counter for Schedule Table

- MAXALLOWDVALUE = Macroticks
- MAXALLOWDVALUE = Macroticks \*Cycles

## Synchronization to Flexray

### Cyclestart Interrupt

```
ISR(CycleStart)
{
    FrIf_GetGlobalTime(FrIf_CtrlIdx, &Cycle, &MacroTick);

    SyncScheduleTable(Tbl, (Cycle << 16) + MacroTick);
}
```

**Remark:** If MacroTicks (per cycle) is smaller than 16 bit adjust!

Thank you for your attention.

For detailed information about Vector  
and our products please have a look at:

[www.vector-informatik.com](http://www.vector-informatik.com)

**Author:**

**Winfried Janz**

**Vector Informatik GmbH**

**Ingersheimer Str. 24**

**70499 Stuttgart**