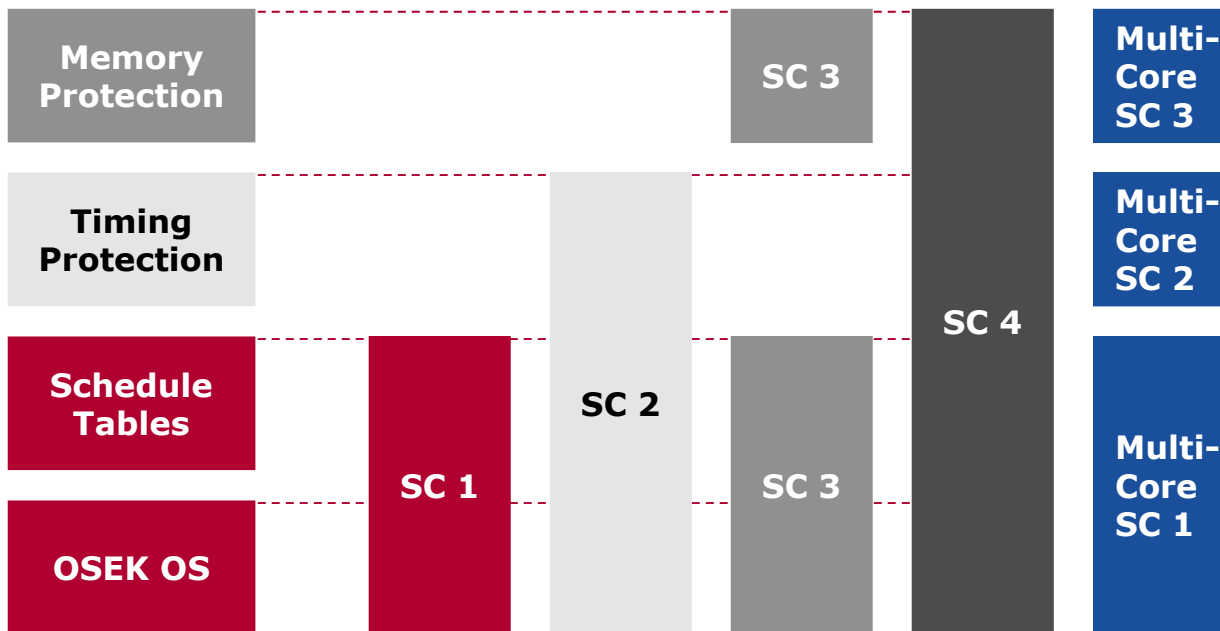


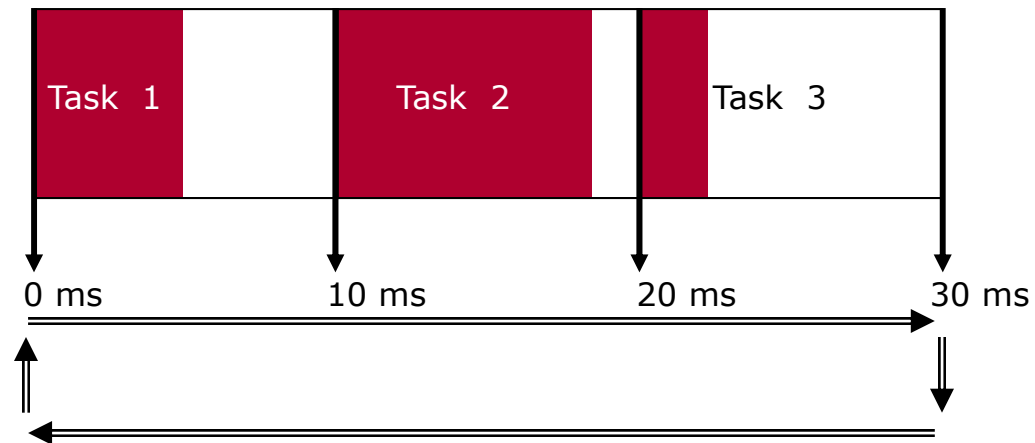
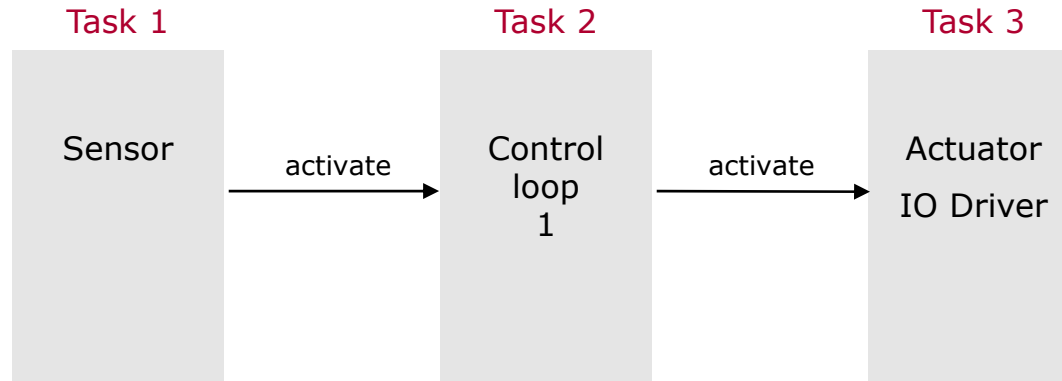


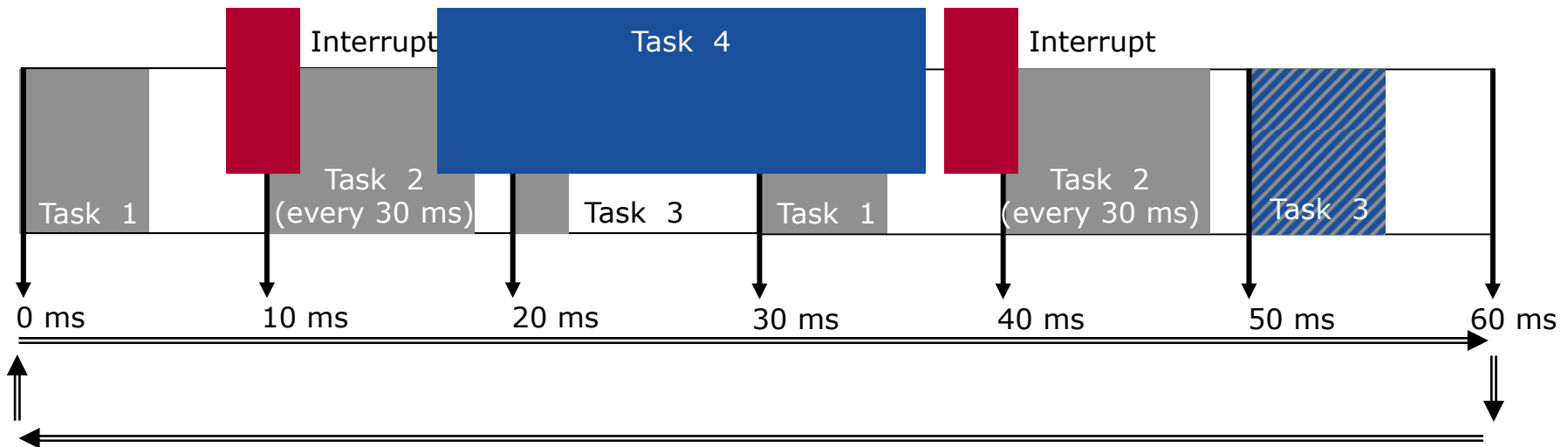
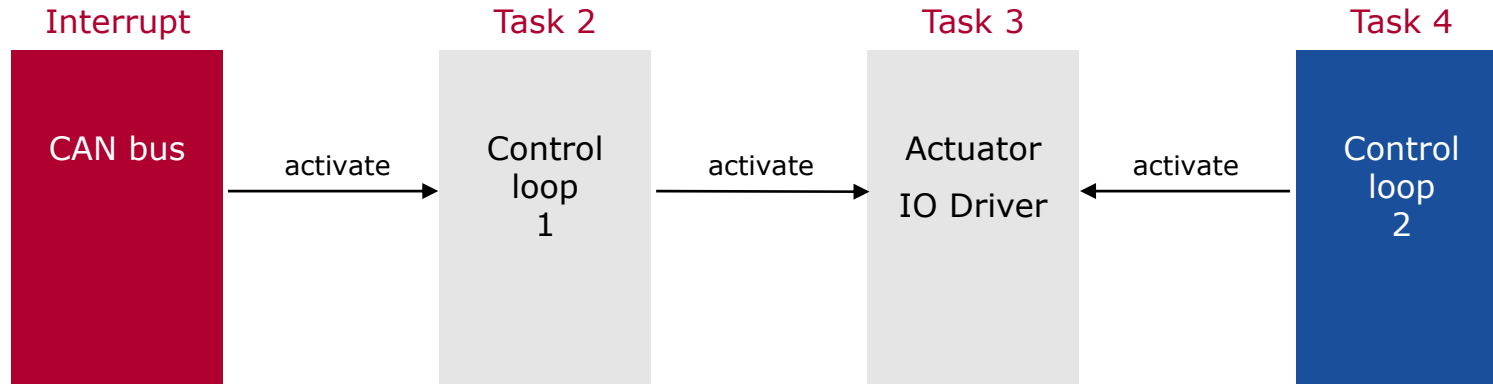
MICROSAR-OS

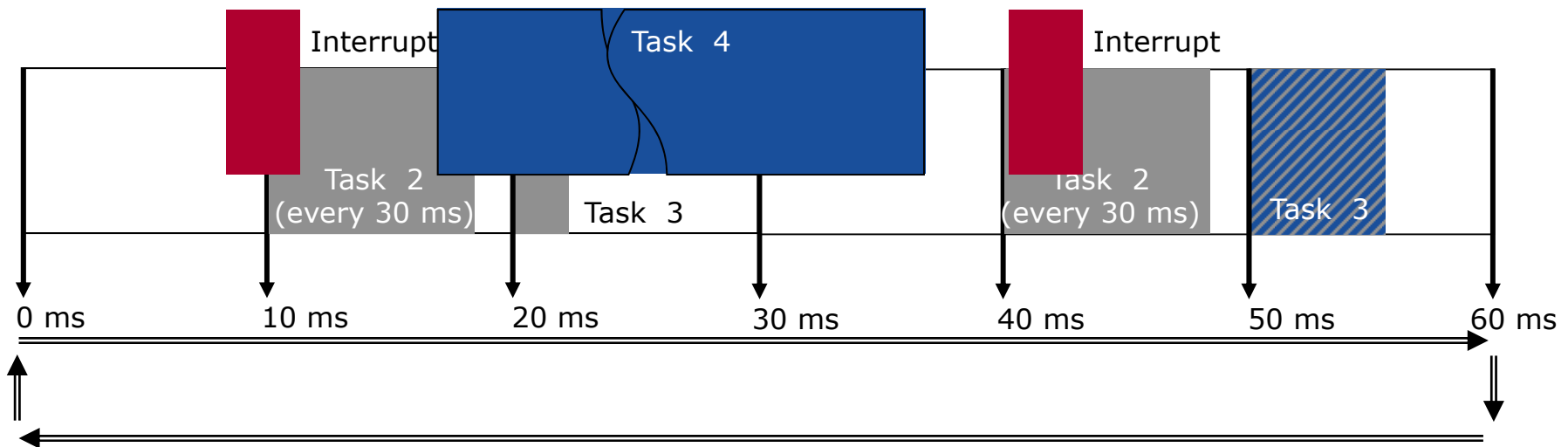
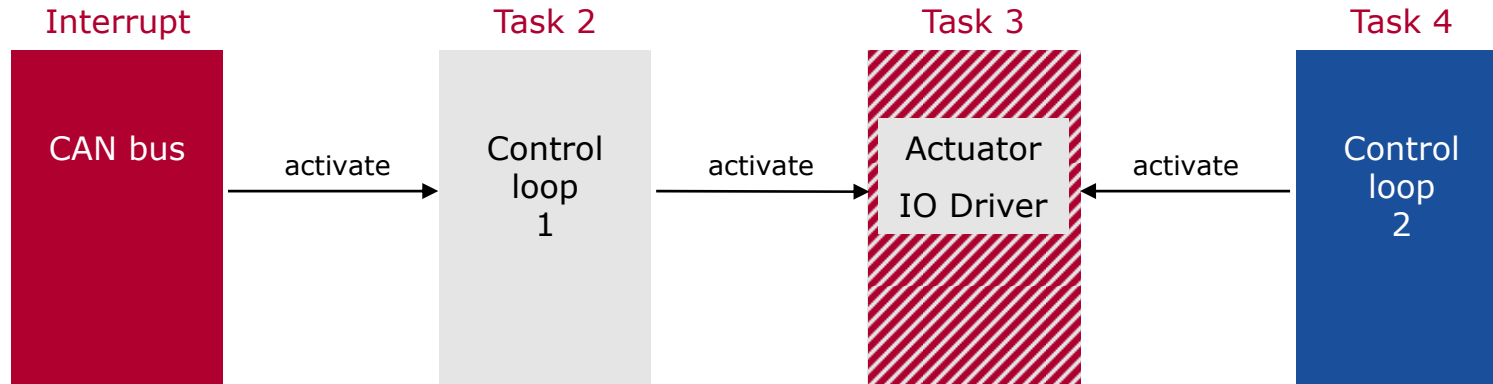
Embedded Real-time Multitasking Operating Systems

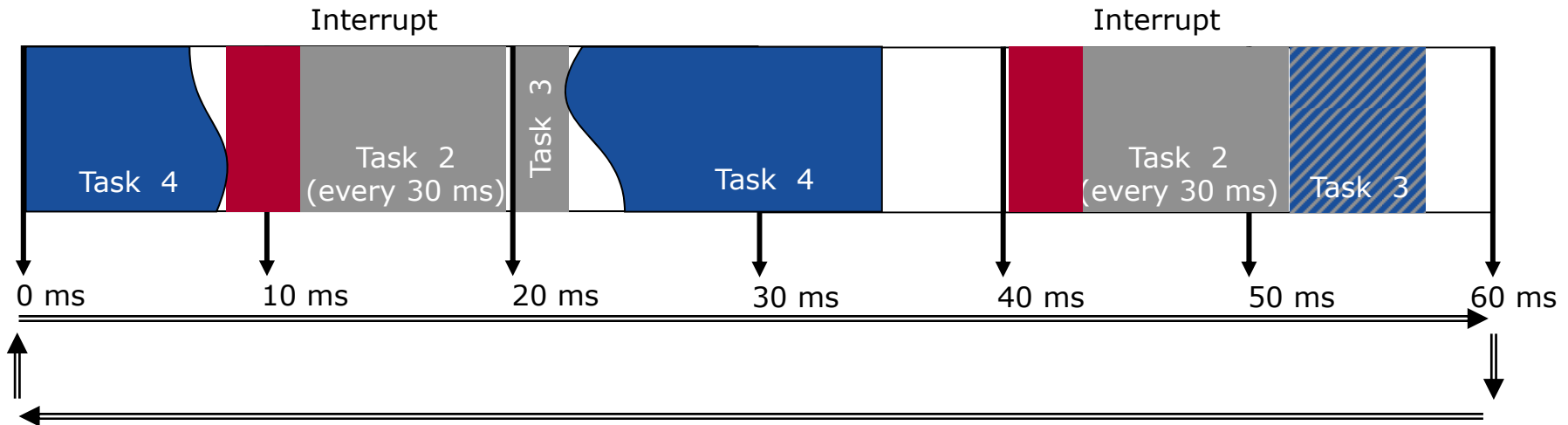
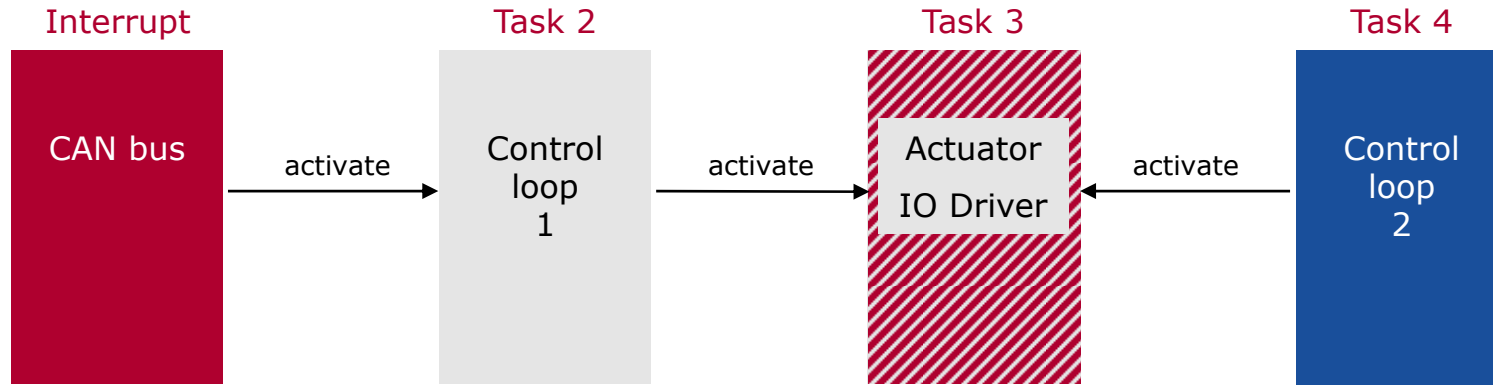
- ▶ AUTOSAR **extends** the OSEK/VDX standard of operating systems.
- ▶ AUTOSAR Add-ons are segmented into Scalability Classes (SC).

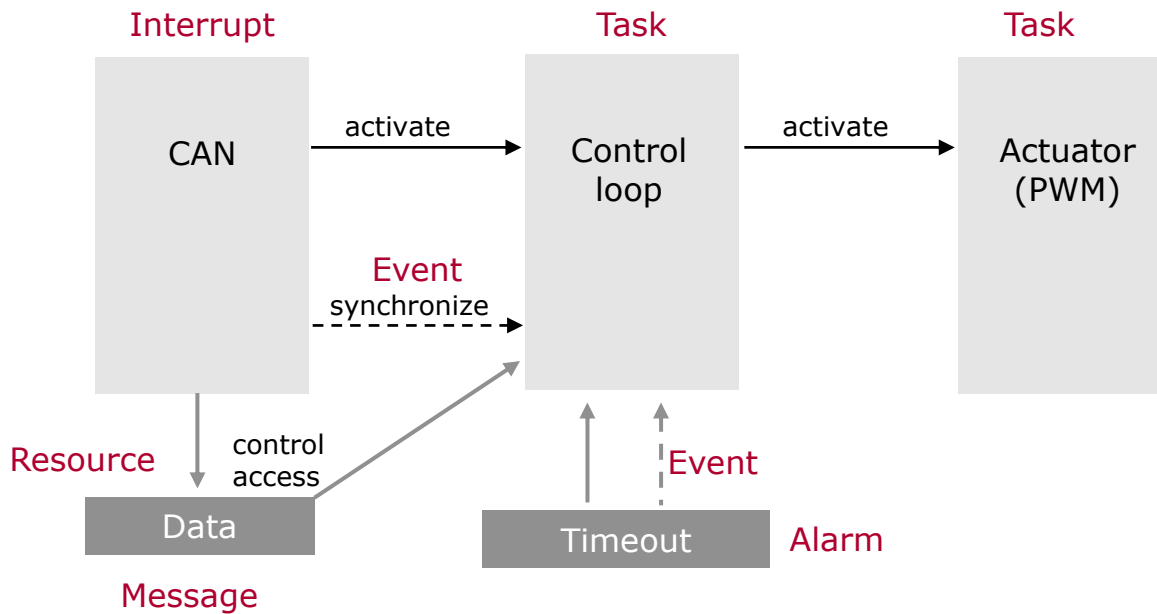
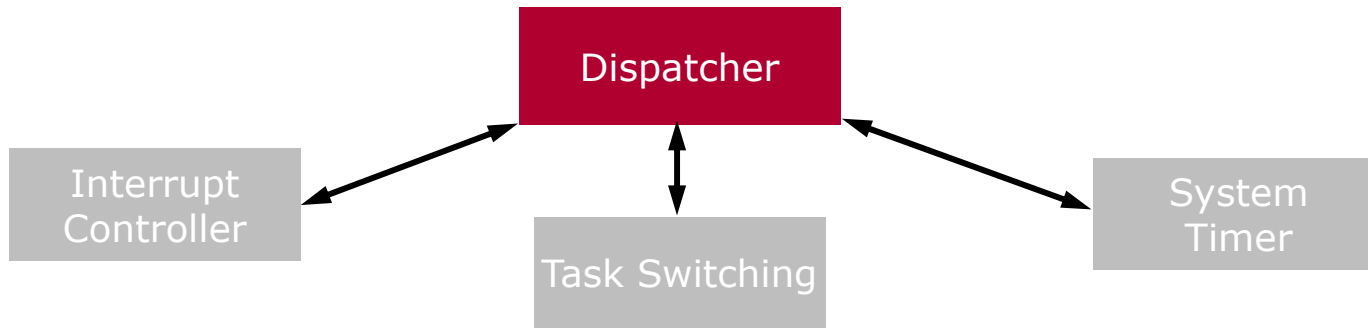












- ▶ Task & Interrupt Management
 - ▶ Multi-Tasking
 - ▶ full, non and mixed preemption
 - ▶ Interrupt disabling / enabling
 - ▶ Task activation / termination
- ▶ Time Management (Alarm Management)
 - ▶ Single-shot
 - ▶ Cyclic repetition
 - ▶ Schedule Tables
- ▶ Task Synchronization (Event Management)
- ▶ Access Synchronization
 - ▶ Resource Management
 - ▶ Message Passing

Initial Expiry Point

Expiry Point 1

Task Activation
Task A
Task B
Event Settings
Event a : Task C
Event b : Task D
Offset = 4

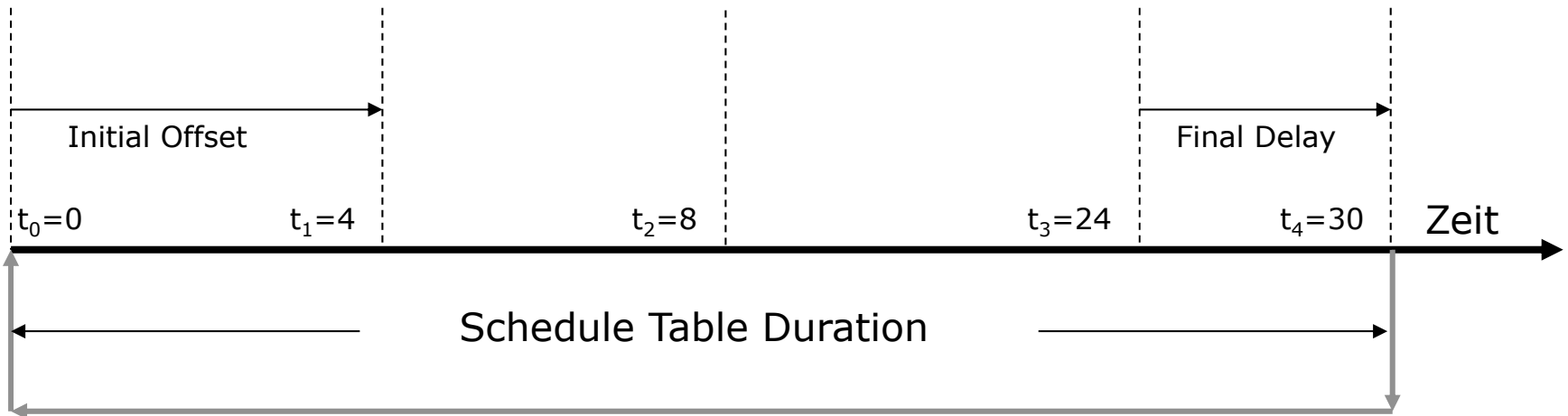
Expiry Point 2

Task Activation
<none>
Event Settings
Event c : Task A
Offset = 8

Final Expiry Point

Expiry Point 3

Task Activation
Task C
Task A
Event Settings
<none>
Offset = 24



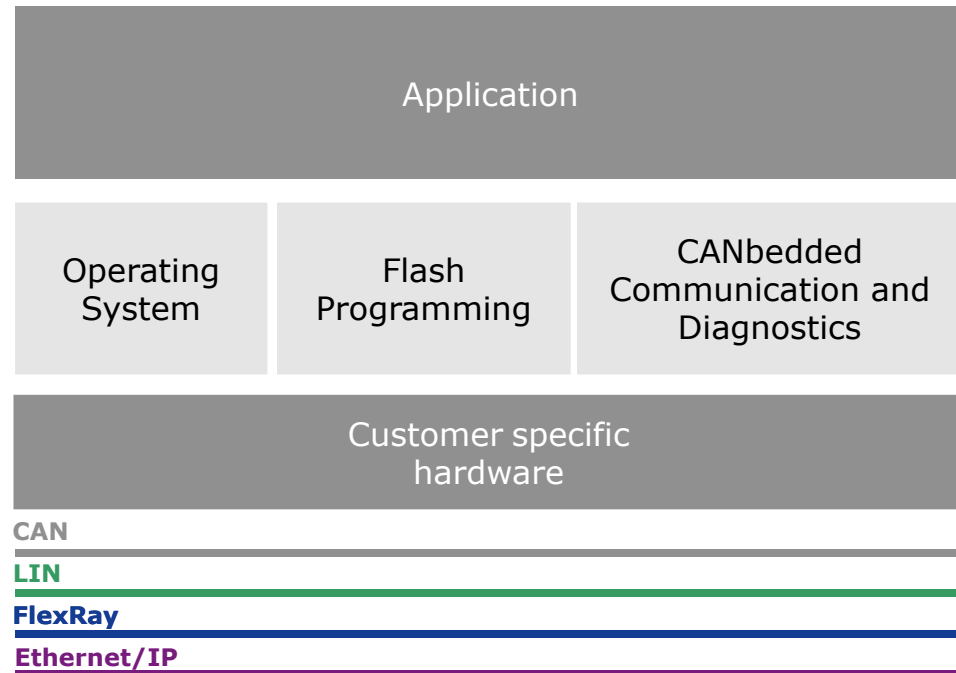
- ▶ Very small, ruggedized kernel (1 to 10 kB ROM)
- ▶ Static configuration
 - ▶ No dynamic object creation
 - ▶ No administration shell
 - ▶ Fast boot time (msec)
 - ▶ Small memory foot print
- ▶ Standardized (OSEK/VDX)
- ▶ Proven reliability;
employment in many automotive control units since 2000
- ▶ Implementations are available for many micro processor families
- ▶ Drivers and protocols available as add-ons

CANbedded - Protocols and drivers

- ❑ CAN
- ❑ LIN
- ❑ FlexRay
- ❑ Ethernet (TCP/IP)

Diagnostic - Protocols

CANflash - Flash Bootloader



Stack Check

Check for stack overflow (task, event, system)

Internal Trace

Logs API calls with configurable buffer size

Timer Initialization

System Timer Setup

Optimized Alarm Management

Minimum timer interrupt runtime

Extended error checking

e.g. consistence of OS data

API is configurable

API functions can be switched on or off

TimingAnalyzer

Tool to visualize the timing of tasks.

Configuration Report Generator

HTML based documentation of the configuration

Component Management

Allows for multiple OIL files in one project
(distributed development)

Template Generator

Generated template main.c file

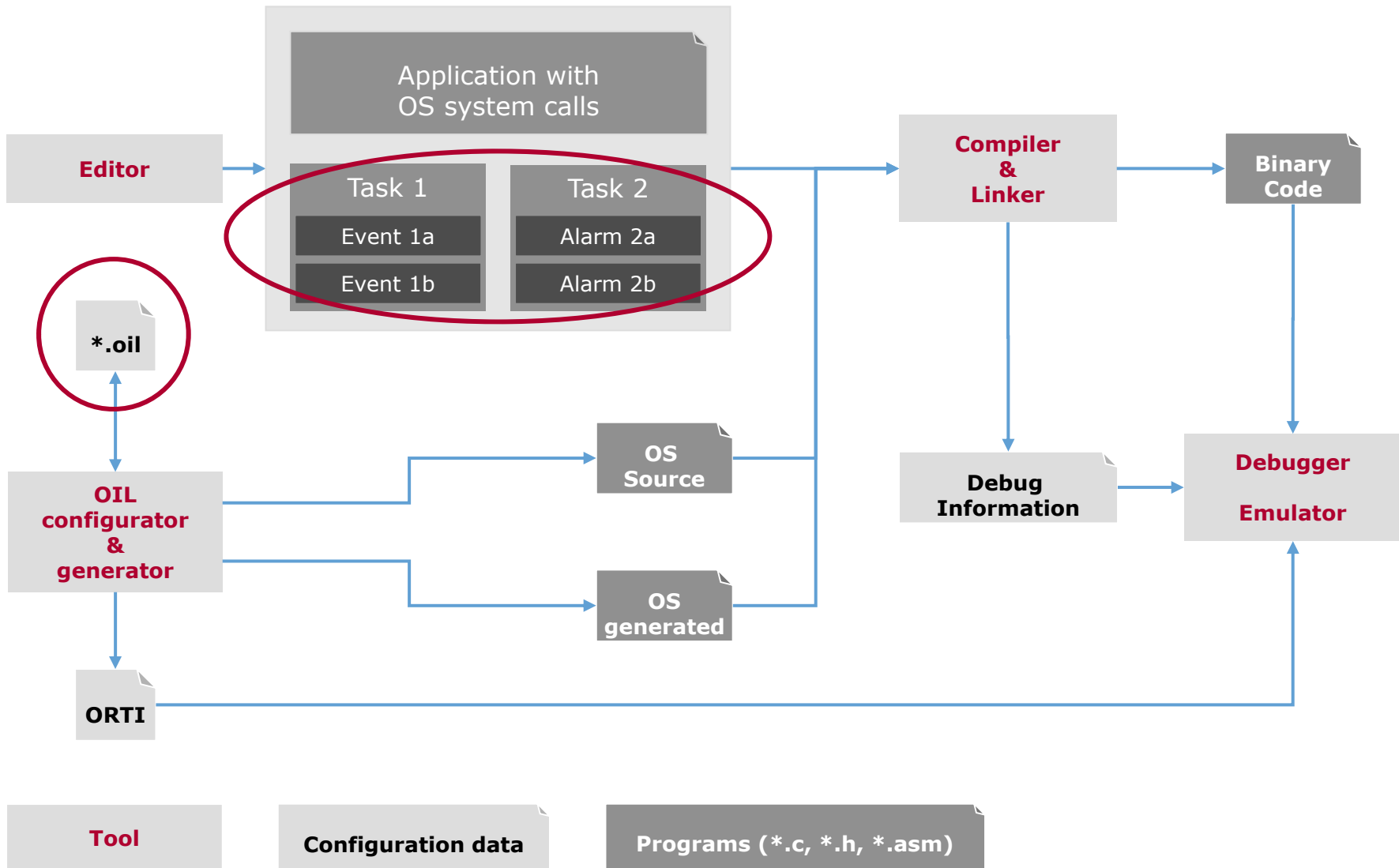
Delivered as Source Code

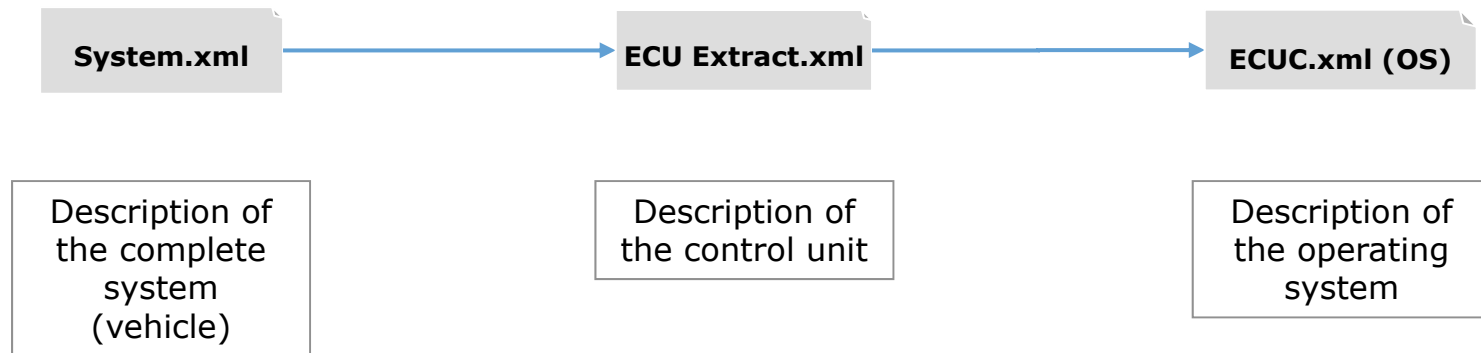
Generator runs also via batch file

Many microcontrollers supported

MICROSAR-OS

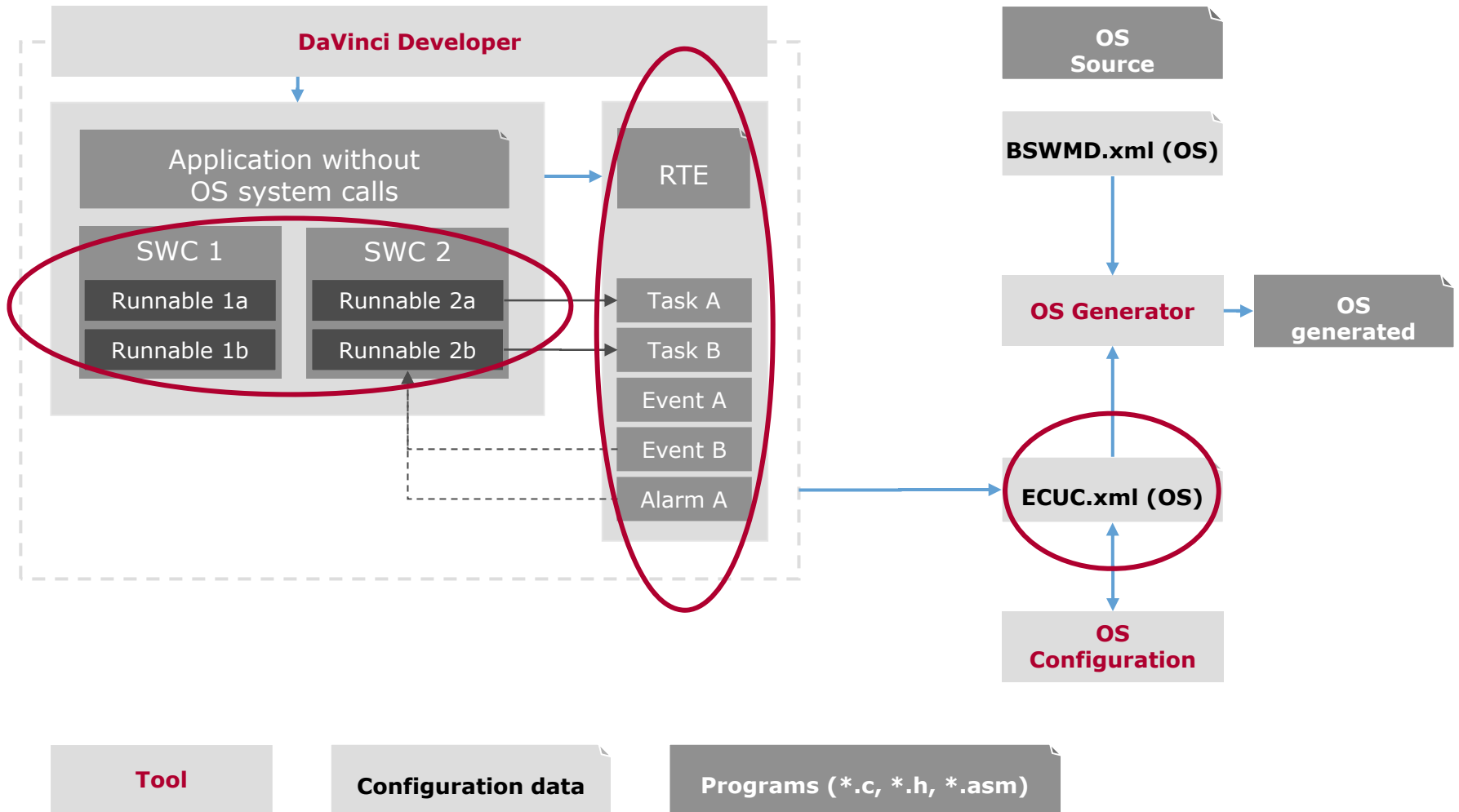
Development Process – OSEK/VDX





MICROSAR-OS

Development Process– AUTOSAR



Vector offers a complete range of services to support the usage of the operating systems

Training

Workshops to explain the usage of OSEK-OS and the design of applications.

Updates

Offering software updates for

- ❑ specification changes
- ❑ support of a new derivative or compiler version

Support

Helping a customer with the handling of osCAN in a project, e.g. help with the installation (set up) or configuration.

Consulting

Assistance with the design and the architecture of your software

Custom-specific solutions

Implementation of an application as a customer's subcontractor

"Hidden" Values

More capacity

By utilizing Vector's employees

Less risk

Since Vector does advanced development

Lower costs

Because of distribution to multiple users

Proven software

Due to multiple, widespread use

Better delivery time

Because of specialized production process

Maintenance

Assumed by Vector

Warranty

Assumed by Vector

- ▶ The webinar series about operating systems
 - ▶ 2011-10-25 MICROSAR OS - a pre-emptive realtime multitasking operating system
 - ▶ 2011-11-08 TimingAnalyzer – schedulability analysis of task runtime
 - ▶ 2011-11-22 Memory and runtime protection of the MICROSAR OS operating system
 - ▶ 2011-11-29 Introduction into the multi-core operating system from Vector

- ▶ Registration to the upcoming Webinars and the list of recorded Webinars:
http://www.vector.com/vi_webinars_en.html

- ▶ The overview of Vector's training services:
http://www.vector.com/vi_training_en.html

- ▶ We stay online for some more minutes to answer your questions. Please ask your questions in the Q&A window.

- ▶ Contact data for additional questions, product information or presentation :
 - ▶ helmut.brock@vector.com
 - ▶ +49 (0) 711 80670 385
 - ▶ embedded@de.vector.com

► A Sample Application

```
main - Visual C++ 2005 Express Edition
File Edit View Debug Tools Window Community Help
beforestart.c
main.c
/* File: main.c, automatically generated by GENTRPL_E2E, Version: 3.00 */
/* Exec: D:\LOCAL_P8\REDIR\B\USER\B\01 */
/* vs: 1 Win_Msc 16 1419146 2009 */

#include <csb.h>

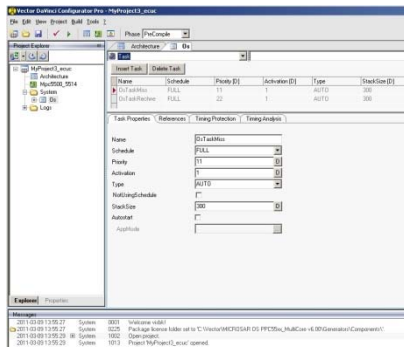
TASK(suppleTask) /* Priority: 10 Schedule: NOW Type: EXTENDED */
{
    EventBackType ev;

    for (;;)
    {
        WaitEvent (0);
        GetEvent (suppleTask, ev);
        CleanEvent (ev);
    }
} /* END OF suppleTask */

TASK(Task2) /* Priority: 20 Schedule: FULL Type: EXTENDED */
{
    EventBackType ev;

    for (;;)
    {
        WaitEvent (0);
    }
}
```

► The Configuration Tool



Sample Application

Task StateMachine

2 States

- Wiper on
- Wiper off

Set up activities of respective state

Task ContactCheck

Checks state of

- Wiper left contact
- Wiper right contact
- Switch „Wiper on/off“

Operates cyclically

Task Wiper

Controls the stepper motor

MICROSAR-OS

Configuration

The screenshot displays the Vector DaVinci Configurator Pro interface for configuring the OS. The main window is titled "Vector DaVinci Configurator Pro - Webinar_OS_Basic_ecuc". The "Project Explorer" on the left shows a tree structure for "Webinar_OS_Basic_ecuc" with sub-items like "Architecture", "Mpc5500_5514", "System", "Os", "Task", "Counter", "Alarm", "Resource", "Event", "Isr", "Message", "Com", "AppMode", "Application", "ScheduleTable", "Implementation Info", and "Logs".

The "Task" table in the center lists tasks with the following columns: Name, Schedule, Priority [D], Activation [D], Type, and StackSize [D].

Name	Schedule	Priority [D]	Activation [D]	Type	StackSize [D]
eTaskStateMachi...	FULL	10	1	EXTENDED	240
eTaskWiper	FULL	30	1	AUTO	120

The "Task Properties" panel on the right shows the configuration for the selected task "eTaskStateMachine":

- Name: eTaskStateMachine
- Schedule: FULL
- Priority: 10
- Activation: 1
- Type: EXTENDED
- NotUsingSchedule:
- StackSize: 240
- Autostart:
- AppMode: /Os/OSDEFAULTAPPMODE

The "Messages" panel at the bottom shows a log of system events:

Time	System	ID	Message
2011-10-26 16:09:19	System	0001	Welcome visbk!
2011-10-26 16:09:19	System	0225	Package license folder set to 'C:\Vector\MICROSAR OS PPC55xx v5.15\Generators\Components\.'
2011-10-26 16:09:23	System	1002	Open project.
2011-10-26 16:09:23	System	1013	Project 'Webinar_OS_Basic_ecuc' opened.

```
26
27     #include <Os.h>
28
29     /** tasks **/
30
31
32     void main(void)
33     {
34         HwSpecBeforeStartOs();
35
36         StartOS(OSDEFAULTAPPMODE);
37     }
38
39
40     TASK(eTaskStateMachine)
41     {
42         int iCurrentState;
43         EventType ev;
44
45         SetRelAlarm(almContactCheck, msec(10), msec(10));
46
47         for(;;)
48         {
49             WaitEvent(evWiperOn | evWiperOff);
50             GetEvent(&ev)
51             ClearEvent(ev)
52
53             if( ev & evWiperOn) {
54                 ActivateTask (eTaskWiper);
55             }
56
57             if( ev & evWiperOff) {
58                 SetEvent (eTaskWiper, evWiperStop);
59             }
60         }
61     } /* END OF eTaskStateMachine */
62
```

```
62 TASK(eTaskWiper)
63 {
64     int iDirection = Left;
65
66     SetRelAlarm(almWiperMotorTrigger, msec(1), msec(1))
67
68     for(;;)
69     {
70         WaitEvent(evMotorTrigger | evLeftContact | evRightContact | evWiperStop);
71         GetEvent(&ev)
72         ClearEvent(evMotorTrigger | evLeftContact | evRightContact)
73
74         if (ev & evMotorTrigger) {
75             TriggerMotor(iDirection)
76         }
77         if(ev & evLeftContact) {
78             iDirection = Right;
79         }
80         if(ev & evRightContact) {
81             iDirection = Left;
82         }
83         if(ev & evWiperStop & evRightContact) {
84             CancelAlarm(almWiperMotorTrigger)
85             TerminateTask();
86         }
87     }
88 } /* END OF eTaskWiper */
89
90 TASK(bTaskContact)
91 {
92     /* test for left contact */
93     if (WiperContact & 0x01) {
94         SetEvent(eTaskWiper, evLeftContact)
95     }
96     /* test for right contact */
97     if (WiperContact & 0x02) {
98         SetEvent(eTaskWiper, evRightContact)
99     }
100
101     /* test for handle port - wiper on */
102     if (HandlePort & 0x01) {
103         SetEvent(eTaskStateMachine, evWiperOn)
104     }
105     /* test for handle port - wiper off */
106     if (HandlePort & 0x02) {
107         SetEvent(eTaskStateMachine, evWiperOff)
108     }
109
110     TerminateTask();
111 } /* END OF bTaskContact */
112
113
114
```

```
114
115  /** hook routines **/
116
117  volatile StatusType gErrorCode;
118  volatile uint16 guiError;
119
120  void ErrorHook(StatusType ErrorCode)
121  {
122      gErrorCode = ErrorCode;
123      guiError = OSErrorGetosCANError();
124
125      /* this function can be filled with user specific code in userspec.c */
126      usrErrorHook();
127  } /* END OF ErrorHook */
128
129
130
131  void ShutdownHook(StatusType ErrorCode)
132  {
133      /*implement userspecific shutdown reaction here...*/
134      ErrorCode = ErrorCode;
135  } /* END OF ShutdownHook */
136
137
138  /** main **/
139
140
141
142  /* END OF MODULE main.c */
143
```

Thank you for your attention.

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

www.vector.com

Author:

Helmut Brock

Vector Informatik GmbH

Ingersheimer Str. 24

70499 Stuttgart