# CONTENTS

1.	Introduction2
1.1	Document organization and format2
1.2	Technical support2
2	Software Installation3
2.1	Installation Instructions3
3	Getting started3
3.1	OSEK / VDX control main window5
4	OSEK Control Features6
4.1	OS object Information7
4.2	Task Information7
4.3	Stack Information
4.4	Vs_Event Information9
4.5	Alarm information9
4.6	Trace Information10
4.7	Task / Service graph11
4.8	Service coverage12

### 1. Introduction

The OSEK / VDX ActiveX control implements the kernel awareness debugging of OSEK applications for Seehau software using Nohau's RTOS interface. This provides the debugging facility for OSEK applications running on Nohau's Emul-12 PC (HC 12 processor family) emulator using Seehau software.

The OSEK / VDX ActiveX control analyzes the application specific information from the ORTI file and integrates with Seehau for the internal OSEK OS data to provide debugging facility for OSEK applications. The ActiveX control supports both the ORTI break and trace interface.

The ActiveX control is developed based on ORTI specification version 2.1 and Vector OS version 2.1.

In this document the OSEK / VDX ActiveX Control and OSEK Control are used as synonyms.

#### 1.1 Document organization and format

This document is organized into sections.

**Section 1 - Introduction -** Provides an overview of the manual. This section of the Introduction summarizes the contents of the remaining sections and appendices. The remainder of this section presents an overview of *OSEK Control* and provides references to related documentation and technical support.

The following sections, if read sequentially, provide a step-by-step guide for debugging OSEK applications using Seehau with OSEK Control.

**Section 2 - Software Installation -** Describes the software and hardware necessary for successful installation and operation of the OSEK Control.

**Section 3 – Getting started** - Describes the steps to configure and use the *OSEK Control* for debugging the OSEK applications.

**Section 4 – OSEK Control features -** Describes the features related to OS Objects and Parameters that are available with OSEK Control.

### 1.2 Technical support

In U.S, technical support representatives are available to answer your questions. You may also fax or e-mail your questions to us. Please include your [voice] telephone number, for prompt assistance. Non-U.S. users may choose to contact their local representatives.

Phone: (248) 488-2080 Fax: (248) 488-2082 E-mail: techsupp@dgtech.com Web site: <u>http://www.dgtech.com</u>

## 2 Software Installation

### 2.1 Installation Instructions

- Unzip the file "OSEKRelease.zip".
- Click on Install.bat that registers the "OsekVdx Control" ActiveX control.

## 3 Getting started

- Make the necessary hardware connections for Seehau.
- Launch the Seehau application.
- Select "File->Load Code..." menu option from Seehau main window to load the required OSEK application.



Screen No. 1 – Load code

• Open the RTOS window by selecting "View->Real Time OS ..." menu option from the Seehau main window.



Screen No. 2 – Select Real Time OS

 Now, select "Select-> OsekVdx Control (ORTI 2.1)" menu option from the RTOS window to bring up the OsekVdx Control window

#### **OSEK / VDX DEBUG SUPPORT SOFTWARE MANUAL**

🔛 Real Time	• OS	_ 🗆 🗡
<u>File</u> <u>Actions</u>	<u>S</u> elect <u>H</u> elp	
	1 RtosCpp Control	
	2 OsekVdx Control (ORTI 2.1)	
	<u>3</u> μC/OS-II Kernel Awareness	
	4 OsekVdx Control	
	5 RTA OsekVdx Control	
	6 RTOSDemo Control	
	7 CMXKAware Control	

#### Screen No. 3 – Select OsekVdx Control

- If a code is loaded before opening RTOS window, then "Action->Loaded" menu option from the RTOS window should be selected.
- The OSEK / VDX ActiveX object will display all the objects and corresponding attribute details that are specified in the input ORTI file. Each Object will have dedicated window for displaying its specific attributes and the information.
- The static attribute details of each object will be updated when the emulator is stopped. The dynamic attribute details of each object will be updated when the trace is stopped.
- The user will be given with the options to start and stop the trace as well as emulator.
- From the RTOS window, select "Actions->Loaded" menu option. The static information present in the OSEK application is shown in the corresponding tabs.

Be R	eal Time (	)S				_ 🗆 ×
<u>File</u> <u>Actions</u> <u>Select</u> <u>He</u>			lp			
En	✓ <u>S</u> tay On	Тор	Nohau for EMUL12-PC			OS Analysis
	I <u>R</u> eset				S	itart Stop
	Loaded					
Emu	Emu <u>D</u> o Read Mem		88		Т	race Stopped
	Endian = $\underline{0}$			പംപ		
Endian = <u>1</u>		XI   STACK	<ul> <li>vs_Event</li> </ul>	ALAHM	RESUU	
	OS Name RUNNIN		GTASK RUNNINGISR2		SERVICETRA	
S12_DP256 basicTas		kThird NO_ISR		ActivateTask		
•						Þ

Screen No. 4 – Select Actions Loaded

 From the Seehau window select "Breakpoints->Add Breakpoint" to place the break point at the required address.

#### **OSEK / VDX DEBUG SUPPORT SOFTWARE MANUAL**

Add Breakpoint	
DEFINE: File Name stSampleFiles\sample1\samplerv.c	
Line	<u></u> K
OR: Address Ox808f	Cancel
Software     C Hardware	<u>H</u> elp

Screen No. 5 – Break point window

• Press "Go". Now, the RTOS window is updated with the trace and breakpoint interface data after it encounters breakpoint.

Enulator     Start     Stop       Go     Break     Trace Stopped						
Emulator         Nohau for EMUL12-PC         RTOS Analysis           Go         Break         Start         Stopped						
Emulator Stopped Trace Stopped						
Enitiator Stopped Trace Stopped						
Object1 Object2 Object3 Object4 Object5 TRACE						
Attribute Name Attribute1 Attribute2 Attribute3 Attribute4						

Screen No. 6 – Press GO

## 3.1 OSEK / VDX control main window

 Select "Select->OsekVdx Control" from the RTOS window to display OSEK Control window

#### **OSEK / VDX DEBUG SUPPORT SOFTWARE MANUAL**

💀 Real Time OS 📃 🗖 🗙							
<u>File Actions Selec</u>	xt <u>H</u> elp						
Emulator     Nohau for EMUL12-PC     RTOS Analysis       Go     Break     Start     Stop							
Emulator Stopped	Emulator Stopped Trace Stopped						
Object1 Object2 Object3 Object4 Object5 TRACE							
Attribute Name	Attribute1	Attribute2	Attribute3	Attribute4			
					_		
					- 11		

#### Screen No. 7 - OSEK / VDX Control window

- "Go" button is used to start the Emulation of the program code.
- "Break" button is used to stop the Emulation of the program code.
- "Start" button is used to record the selected data into trace buffer.
- "Stop" button is used to stop recording the trace data.
- Status of the emulator is displayed (started, stopped) on the RTOS window.
- The address, at which the emulation is stopped, displayed on the RTOS window.
- Status of the trace is displayed (started, stopped) on the RTOS window.

### 4 **OSEK Control Features**

Note: The following OSEK control features are described using a sample file as an example. The features described below are specific to the sample file.

## 4.1 OS object Information

🕂 Real Time OS 📃 🗖 🗙							
<u>File</u> <u>A</u> ctions	<u>File Actions Select H</u> elp						
Emulator     Nohau for EMUL12-PC     RTOS Analysis       Go     Break     Start     Stop							
Emulator Stopp	ed at: 0xC056		Trace Stopped				
OS TAS							
OS Name RUNNINGTASK RUNNIN SERVICETRACE F							
S12_DP256	basicTaskSeco	NO_ISR	ReleaseResource 2				
1	1						

#### Screen No. 7a - OSEK / VDX OS Information window

- The OS tab displays the OS object information.
- The OS window displays the various OS object attributes like OS Name, RUNNINGTASK, RUNNINGISR, SERVICETRACE etc. that are defined for the OS object in the OSEK application.

### 4.2 Task Information

📲 Real Time OS 📃 🗖 🗙						
<u>File Actions Selec</u>	st <u>H</u> elp					
Emulator     Nohau for EMUL12-PC     RTOS Analysis       Go     Break     Start     Stop						
Emulator Stopped at:	0xC056		Tra	ce Stopped		
OS TASK (						
TASK Name PRIORITY vs_HomePri STATE STACK						
extendedTaskF	10	10	WAITING	osTas		
extendedTaskS	8	8	SUSPEN	osTas		
basicTaskThird	12	12	SUSPEN	osTas		
basicTaskSeco	2	2	RUNNING	osTas		
basicTaskFirst	1	1	READY	osTas		

Screen No. 8 – OSEK / VDX Task Information window

- The TASK tab displays the TASK object information.
- The TASK window displays the various TASK object attributes like TASK Name, PRIORITY, vs\_HomePriority, STATE etc. that are defined for the TASK object in the OSEK application.

### 4.3 Stack Information

🐱 Real Time OS 📃 🗖 🗙						
<u>File Actions Select H</u> elp						
Emulator     Nohau for EMUL12-PC     RTOS Analysis       Go     Break     Start     Stop						
Emulator Stopped at: 0	xC056		Trace Stopped			
STACK Name	SIZE	BASEADDRESS	STACKDIRECTION			
osSystemStack	0x50	0x11FF	DOWN			
osTaskStack0	0x50	0x110F	DOWN			
osTaskStack1	0x50	0x11AF	DOWN			
osTaskStack2	0x50	0x115F	DOWN			
osTaskStack3 0x50		0x10BF	DOWN			
osTaskStack4	0x50	0x106F	DOWN			

Screen No. 9 – OSEK / VDX Stack Information window

- The STACK tab displays the STACK object information.
- The STACK window displays the various STACK object attributes like STACK Name, SIZE, BASEADDRESS, STACKDIRECTION etc. that are defined for the STACK object in the OSEK application.

## 4.4 Vs\_Event Information

Real Time OS			
<u>File</u> <u>Actions</u> <u>Select</u> <u>H</u> elp	I		
Emulator Nohau for EMUL12-PC Go Break			RTOS Analysis
Emulator Stopped at: 0xC056	Trace Stopped		
OS TASK CONTEX	OS TASK CONTEXT STACK vs_Event		
vs_Event Name	vs_Event Name vs_EventMask		
evExT1_1	evExT1_1 0x0001		
evExT2_1	evExT2_1 0x0001		II
			II

Screen No. 11 – OSEK / VDX vs\_Event information window

- The vs\_Event tab displays the vs\_Event object information.
- The vs\_Event window displays the various vs\_Event object attributes like vs\_Event Name, vs\_EventMask etc. that are defined for the vs\_Event object in the OSEK application.

### 4.5 Alarm information

🔛 Real Time OS				- 🗆 🗵
<u>File</u> <u>Actions</u> <u>S</u> elect	<u>H</u> elp			
Emulator Go Break	EMUL12-PC	RTOS Ana	lysis Stop	
Emulator Stopped at: 0xCC3F Trace Stopped				
vs_Event ALARM	RESOURCE	vs_OS_Config T	RACE	
ALARM Name	CYCLETIME	vs_CycleTime	ALARMTIME	- vs
myFirstAlarm	0x00	0x00	0x282	0x

Screen No. 12 - OSEK / VDX Alarm information window

- The ALARM tab displays the ALARM object information.
- The ALARM window displays the various ALARM object attributes like ALARM Name, CYCLETIME, ALARMTIME etc. that are defined for the ALARM object in the OSEK application.

### 4.6 Trace Information

🔛 Real Time OS				
<u>File Actions Select H</u> elp				
Emulator Nor Go Break	nau for EMUL12-PC	RTOS Analysis Start Stop		
Emulator Stopped at: 0xCC3F Trace Stopped				
vs_Event ALARM RESOL	IRCE   vs_OS_Config [ <u>I</u> ask / Service Graph <u>S</u> ervice Coverage <u>Application Analysis</u>			



This module displays the graphical representation of the "Task / Service Graph", "Service coverage" and "Application Analysis" of the OSEK / VDX application after the data has been recorded in the trace buffer.

Refer: Section [4.6], Screen No. 13

🔛 Task / Service Graph			
Resolution 1	Time 25.	00 use	c
	5.07602 s 5.03	607 s	5.07612 s 5.07617 s 5.07622 s 5.07627 s 5.07632 s 5.0763
TASK LIST		1	1
IDLETASK		-	
extendedTaskFirst			
extendedTaskSecond		+	
basicTaskThird			
basicTaskSecond		-	
🖸 basicTaskFirst			
SERVICE LIST			
S ActivateTask			
			F

4.7 Task / Service graph

Screen No. 14 – OSEK / VDX Task/Service graph window showing the task information

🗱 Task / Service Graph								_ 🗆 ×
Resolution	Time 25.0	Duse	c					
SERVICE LIST	5.07602 s 5.076	07 s 1111	5.07612 s	5.07617 s	5.07622 s	5.07627 s	5.07632 s	5.0763
S ActivateTask					-			
🕥 TerminateTask								
🕒 ChainTask								
GetTaskState								•
G GetResource								—
S ReleaseResource								
SetEvent								
S ClearEvent	•							

Screen No. 14a – OSEK / VDX Task/Service graph window showing the service information

The Task / Service graph window shows the graphical representation of task and service switching times. **Refer: Section [4.7], Screen Nos. 14 and 14a.** 

- All the Tasks and Services (services that have become active during the trace data capture) are sown on the left hand side of the graph. The Tasks are represented with letter "T" and the services are represented with the letter "S".
- The latest Task / Service that is ON (active) is shown with green.
- The **blue** color lines show the task switching and the **black** color lines show service switching timings
- Thick lines represent task / service ON (active) period
- Thin lines represent task OFF (inactive) period
- The cursors can be moved on the graph (by selecting and dragging with mouse) to measure the time difference. The **Time** text box displays the exact time difference between the cursors.
- The **Resolution** selection box can be used to set the resolution for time scale at x1, x10, x100 and x1000 factors to scale the display.

#### 4.8 Service coverage

🔜 Service Coverage										х			
Service	ON Time	Count	Avg ON Time	Ratio	0	10	20	30	40	50	60	70	80
ActivateTask	0.028809 ms	6	0.004801 ms	6.102541 %									
TerminateTask	0.093750 ms	6	0.015625 ms	19.858837 %									
ChainTask	0.030273 ms	2	0.015137 ms	6.412657 %									
GetTaskState	0.021484 ms	4	0.005371 ms	4.550904 %									
GetResource	0.016113 ms	4	0.004028 ms	3.413178 %									
ReleaseResource	0.073242 ms	4	0.018311 ms	15.514677 %									
SetEvent	0.067297 ms	6	0.011216 ms	14.255361 %									
ClearEvent	0.026367 ms	6	0.004395 ms	5.585258 %									
WaitEvent	0.114746 ms	6	0.019124 ms	24.306369 %									
<b> </b> •													►

Screen No. 15 – OSEK / VDX Service coverage window

The Service Coverage window shows the following statistics of the services

- The total ON time of a service.
- The count, which shows number of times each service became active.
- The average ON time of a service.
- Graphical representation of the percentage average ON time.

Refer: Section [4.8], Screen No. 15

Application Analysis		_ 🗆 ×
Task / Service	Time	<b>▲</b>
S ClearEvent	5076.021973 ms	
S GetTaskState	5076.057129 ms	
S ActivateTask	5076.075195 ms	
S WaitEvent	5076.105957 ms	
🕡 extendedTaskSecond	5076.113770 ms	
S WaitEvent	5076.148926 ms	
🚺 basicTaskFirst	5076.157227 ms	
S GetResource	5076.199219 ms	
S ActivateTask	5076.214844 ms	
S ReleaseResource	5076.251953 ms	
DesicTaskSecond	5076.259766 ms	
<b>10</b> N0_TASK	5076.278809 ms	-

## 4.9 Application Analysis

#### Screen No. 16 – OSEK / VDX Application Analysis window

The Application Analysis window shows the application snapshot in the text format. Application Analysis window includes the following parameters

- Task / Service
- Time