AN provider

DENSO HF-band RFID providers

Version 1.0.0

User's Guide

November 11, 2020

Remark

Revision history

Version	Dating	Content
1.0.0	2019-08-01	First edition
	2020-11-11	Clerical corrections

[Operation Check Model]

Model	Version	Note
AN10R-01	9000801	

[Operation Check Tag]

Tag	Standard
Rectangular tag	ISO/IEC 18000-3 mode 1 International Standard
M12 Cylinder tag	ISO/IEC 18000-3 mode 1 International Standard

Table of contents

1. Introduction

This document is a User's Guide for AN Providers, which are providers for DENSO HF-band RFID Reader/Writer.

An AN provider is a provider that outputs data from a reader/writer.

Figure 1-1 configuration diagram describes the features and implemented methods of this AN provider.

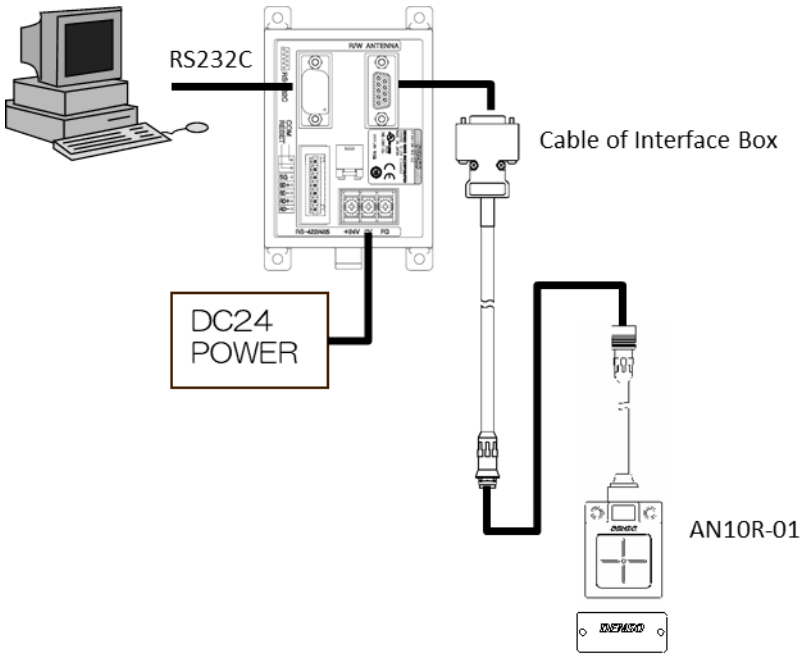

Figure 1-1 configuration diagram1

1.1. Source of reference information

All examples of programming in this book are provided in Visual Basic for Applications, but can be developed in a variety of programming languages, such as C++, Java, or.NET. For instructions, see the ORiN2 Programming Guide.

The ORiN 2 Programming Guide corresponds to the following files in the ORiN 2 SDK install folder:

- ORiN2¥CAO¥Doc¥ORiN2_ProgrammersGuide_<lang>.pdf

※Replace the <lang> part with the environment-specific language strings.

This chapter explains the basic knowledge and techniques of ORiN2,COM/DCOM required to develop applications using providers.

2. Environment Setup for Application Development

2.1. Connecting the Reader/Writer to the Client PC

For details on connecting the Reader/Writer to the Client PC, refer to the HF-band Reader/Writer User's Manual.

2.2. Setting Up the PC Development Environment

2.2.1. Automatic installation of AN providers

If the ORiN2 SDK is installed, the operating environment (Line Time) for connecting to the Reader/Writer is ready.

To set up the development environment, prepare a programming environment that supports Component Object Model (COM, Component Object Model) such as Microsoft Visual Studio, LabVIEW.

2.2.2. Manually Installing AN Providers

In order to use the AN provider, you must manually register the following registry. When registering with the registry, start the command prompt with administrator privileges and execute the regsvr32 command.

In addition, you must have one legitimate ORiN2 SDK license for each PC before the CAO engine can run. Refer to the "Adding or Removing a License" section in the ORiN2 SDK User's Guide.

Table 2-1 Manually Installing AN Providers

File name	CaoProvDENSOAN.dll
ProgID	CaoProv.DENSO.AN
Registry registration[1]	Regsvr32 CaoProvDENSOAN.dll
Unregistering the registry	Regsvr32 /u CaoProvDENSOAN.dll

[1] You do not need to manually register/delete the ORiN SDK installations.

3. AN provider programming

The AN provider can connect the client PC to the reader/writer in the following procedure.

- Creating a CaoEngine
- Creating a CaoWorkspace
- Creating a CaoController

After connecting to the Reader/Writer, you can use the Execute method of the CaoController to access the information of the Reader/Writer itself and the information of the RFID tags read by the Reader/Writer.

3.1. Sample Programming to Get RFID Tags

The following example shows a sample program that acquires RFID tags read by the reader/writer. Table 3-1 describes the requirements of the sample program, Figure 3-1 Sample program flow describes the flow of the sample program, and 3.1.1 describes the actual program.

Table 3-1 Sample Program Requirements

Requirement	Description
Destination	Connect by RS232C
Processing content	Gets the memory data of the RFID tag using the TAG_DATA variables.

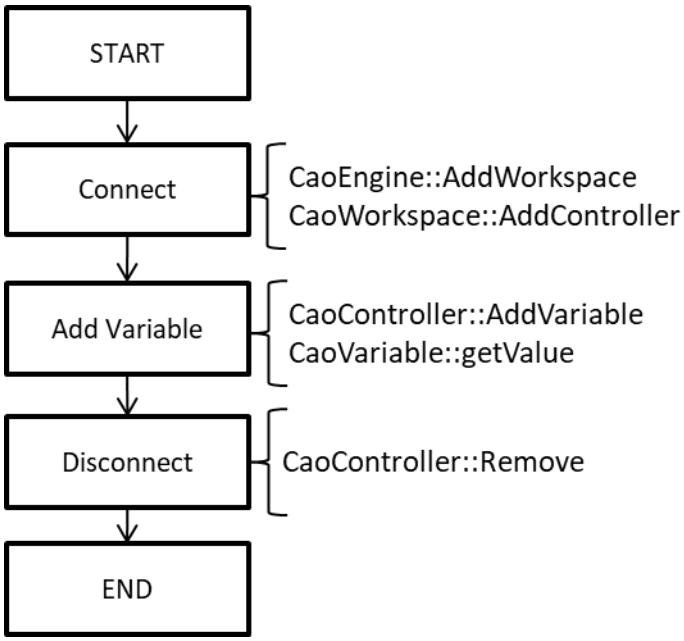

Figure 3-1 Sample program flow

3.1.1. Sample program

The following is an overview of the sample program.

Sample	TAG_ID Sample.vb

```
    Imports CAOLib

    Public Class Form1
        Dim engine As CaoEngine
        Dim workspace As CaoWorkspace
        Dim controller As CaoController
        Dim variable As CaoVariable

        Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
            ' Create CaoEngine
            engine = New CaoEngine
            ' Create CaoWorkspace
            workspace = engine.Workspaces.Item(0)
            ' Create CaoController
            controller = workspace.AddController("Sample",_
                                                 "CaoProv.DENSO.AN",_
                                                 "",_
                                                 "Conn=com:1")
            variable = controller.AddVariable("TAG_DATA",_
                                               "machineNo = 1, Addr = 0, Length = 10")

            TextBox1.Text = variable.Value

            ' Delete CaoVariable from CaoController
            Call controller.Variables.Remove(variable.Index)
            variable = Nothing

            ' Delete CaoController from CaoWorkspace
            Call workspace.Controllers.Remove(controller.Index)
            controller = Nothing

            ' Delete CaoWorkspace from CaoEngine
            Call engine.Workspaces.Remove(workspace.Index)
            workspace = Nothing

            ' Delete CaoEngine
            If engine IsNot Nothing Then
                engine = Nothing
            End If
        End Sub
    End Class
```

3.1.1.1. Connection

To connect to the Reader/Writer, follow the procedure below.

(1) Provides variables for holding objects. The objects required for connecting controllers are the CaoEngine object, the CaoWorkspace object, and the CaoController object. The CaoWorkpace object does not need to have variables when retrieving the CaoController object from the CaoWorkspaces. You also need CaoVariable objects for accessing variables. The following is an example of code in VB.Net.

```
Dim engine As CaoEngine                 ' Variables for CaoEngine Objects
Dim workspace As CaoWorkspace           ' Variables for CaoWorkspace Objects
Dim controller As CaoController         ' Variables for CaoController Objects
```

(2) Generates CaoEngine objects. CaoEngine objects are generated using the new keywords.

```
' Creating CaoEngine Objects
Engine = New CaoEngine
```

(3) Gets or creates a CaoWorkspace object. When you create a CaoEngine object, it defaults to creating one CaoWorkspaces object and one CaoWorkspace object. Here are some examples of newly generated CaoWorkspace objects and a CaoWorkspace of defaults:

```
' Creating CaoWorkspace Objects
Workspace = engine.AddWorkspace("NewWrks", "")
```

(4) Create a CaoController object. To create a CaoController object, set the provider name to use and the parameters to use. For AN providers, specify the connectivity options. Here is an example of the code:

```
' Creating CaoController Objects
Controller = workspace.AddController("SampleController", _
                                     "CaoProv.DENSO.AN", _
                                     "", _
                                     "conn=com:1")
```

3.1.1.2. Get RFID Tags

To acquire the information of RFID tags read by the reader/writer, TAG_ID generation values, which are CaoController user variables, are acquired. For more information, see the TAG_ID variable.

3.1.1.3. Disconnection

When disconnecting from the controller, delete the created object and delete the object to be deleted from the collection class that manages the object. Here is an example of the code:

```
' Remove CaoController from CaoWorkspace
Call workspace.Controllers.Remove(controller.Index)
' Clear CaoController
Controller = Nothing
' Remove CaoWorkspace from CaoEngine
Call engine.Workspaces.Remove(workspace.Index)
' Clear CaoWorkspace
Workspace = Nothing
' Clear CaoEngine
Engine = Nothing
```

4. Command reference

4.1. Method/Property List

Table 4-1 Method/Property List

Category	Methods/Properties[2]		Facility	Reference
CaoWorkspace				
	Addcontroller	M	Connect to Controller	P.11
CaoController				
	VariableNames	P	Get list of variable names that can be connected	P.13
	Variables	P	Get the collection of variables held by the controller	P.13
	AddVariable	M	Adding Variable Objects	P.14
	Execute	M	Executing Extended Commands	P.14
	OnMessage	E	Message reception event	P.17
CaoVariable				
	Value	P	Get/Set Values	P.17

4.2. Method Properties

4.2.1. CaoWorkspace classes

4.2.1.1. AddController method

Add controller objects to the CaoWorkspace. The following are the specifications of the AddController method.

Format

CaoController AddController

```
(
        Controller name                 // Controller name (optional)
        "CaoProv.DENSO.AN",             // Provider name (fixed)
        Machine name                    // Provider Running Machine Name (Unused)
        Option                          // Option string (optional)
)
```

Option

[2] M: Methods, P: Properties, and E: Events are shown.

Here are the options that you specify in the option string: The option string is a string of the options shown below, concatenated with a comma (,).

Option	Required	Description	Values range	Default value
CONN = <Communication parameter>	✓	Specify the host information.	COM	--
Timeout Communications timeout	--	Specifies the communication timeout in ms.	0 or more	500
@EventDisable	--	Configure the occurrences of OnMessage events(4.2.2.5).	True/False	False

Examples of uses

```
Dim engine As CaoEngine 'Engine objects
Dim workspace As CaoWorkspace 'WorkSpace objects
Dim controller As CaoController 'Controlle objects

Engine = New CaoEngine
Workspace = engine.Workspaces.Item(0)
Controller = workspace.AddController("SampleController", _
                                     "CaoProv.DENSO.AN", _
                                     "", _
                                     "conn=com:1")
```

4.2.1.1.1. CONN options

The following are the connection-parameter strings for the Conn options: Here, square brackets ("[]") are optional, and the underscore in the description of each parameter indicates the default value when no option is specified.

RS232C

"Conn=COM:<COM Port>[<:BaudRate>]"

<COM Port> : COM port number . '1' -COM1, '2' - COM2, …

<BaudRate> : Communications speed 9600, 19200, 38400, 115200

4.2.2. CaoController classes

4.2.2.1. VariableNames properties

Gets the list of variable names that can be connected. Variable names obtained with this property can be used as the first arguments of the AddVariable method described later.AddVariable method

Examples of uses

```
Dim engine As CaoEngine 'Engine objects
Dim workspace As CaoWorkspace 'WorkSpace objects
Dim controller As CaoController 'Controlle objects

Engine = New CaoEngine
Workspace = engine.Workspaces.Item(0)
Controller = workspace.AddController("SampleController", _
                                      "CaoProv.DENSO.AN", _
                                      "", _
                                      "conn=com:1")

' Acquisition of file name list
Dim variables as Variant
Variables = controller.VariableNames
```

4.2.2.2. Variables properties

Gets the collection of variables held by the controller.

Examples of uses

```
Dim engine As CaoEngine 'Engine objects
Dim workspace As CaoWorkspace 'WorkSpace objects
Dim controller As CaoController 'Controlle objects

Engine = New CaoEngine
Workspace = engine.Workspaces.Item(0)
Controller = workspace.AddController("SampleController", _
                                      "CaoProv.DENSO.AN", _
                                      "", _
                                        "conn=com:1")

' Retrieve Variable Collection
Dim variables as CaoVariables
Variables = controller.Variables

' Get Variable
Dim variable as CaoVariable
Variable = variables.Item(0)
```

4.2.2.3. AddVariable method

Add variable objects to the CaoController. Variable names can only be those shown in the 4.3Variable List.

The AddVariable specifications are as follows:

Format

CaoVariable AddVariable

(

Variable name // Variable name

Option // Option string (optional)

)

4.2.2.4. Execute method

Execute the ConController extended commands. The Execute specifications are as follows:

Format

Variant Execute

(

"<extended command name>", // Extended command name

"<Option string>" // Option string (optional)

)

The following is a list of extended commands that can be specified by Execute. The examples are described in the extension command detail.

Table 4-2 Execute Commands

Commanded	Description	Reference
RAW	Sends a message and returns a response.	P.15
ReadTag	Reads data from tags.	P.15
WriteTag	Writes data to the tag.	P.16
ReadID	You can scan up to four UIDs.	P.16
ConnectState	Gets the connection status.	P.17

4.2.2.4.1. Raw command

The entered message is sent to the reader/writer.

Item	Type Description	
Argument	VT_BSTR	String in message format. Enter a hexadecimal string with a length that is a multiple of 2.
Returned value	VT_BSTR	Response message

Refer to the HF band reader/writer manual for the message format.

In this manual, the following formats are assumed for messages.

Table 4-3 Message Format

Item	STF	ADDR	CTL	LEN	DATA					ETF	FCS
					CMD	Para1	Para2	Para3	Data		
Length	2	1	1	2	1	1	1	1	N	2	2

Examples of uses

```
' Send command message
Dim command As string
Command = controller.Execute("RAW","AA55010000047100000055AA0274")

End If
```

4.2.2.4.2. ReadTag commands

Reads data from tags.

Item	Type Description		
Argument	VT_UI2|VT_ARRAY		
	1	VT_UI2	Equipment No. Range: 1-31
	2	VT_UI2	The read start address. Range: 0-65535
	3	VT_UI2	Read data length. Range: 1-65535
Returned value	VT_BSTR		Scanned data

Examples of uses

```
' Send command message
Dim param As Array
Dim data As string
Param = {1,0,10}
```

```
String = controller.Execute("ReadTag", param)

End If
```

4.2.2.4.3. WriteTag commands

Writes data to the tag.

Item	Type	Description	
Argument	VT_VARIANT|VT_ARRAY		
	1	VT_UI2	Equipment No. Range: 1-31
	2	VT_UI2	Write start address. Range: 0-65535
	3	VT_BSTR	Written data Supports only input in hexadecimal notation and with a length that is a multiple of 2.
Returned value	VT_BOOL		Writing result

Examples of uses

```
' Write data to Tag
Dim result As bool
Dim param As Array
Param = {1,0,"AABBCCDDEE"}

Result = controller.Execute("WriteTag", param)

End If
```

4.2.2.4.4. ReadID commands

You can read UIDs from up to four tags.

Item	Type	Description	
Argument	VT_UI2		Equipment No. Range: 1-31
Returned value	VT_ARRAY | VT_VARIANT		
	1	VT_UI2	Number of UID data read
	2	VT_BSTR	First scanned UID data* Empty character if no data
	3	VT_BSTR	Second scanned UID data * Empty character if no data
	4	VT_BSTR	Third scanned UID data * Empty character if no data
	5	VT_BSTR	4th UID data scanned* Empty character if no data

Examples of uses

```
'Read UID
Dim result As Array
Result = controller.Execute("ReadID", 1)
TextBox1.Text = result(0).ToString()
                              + result(1).ToString()
                              + result(2).ToString()
                              + result(3).ToString()
                              + result(4).ToString()
End If
```

4.2.2.4.5. ConnectState commands

Writes data to the tag.

Item	Type	Description
Argument	VT_UI2	Equipment No. Range: 1-31
Returned value	VT_BSTR	Connectable: "OK" Connectable: "NG"

Examples of uses

```
' Get connection status
Dim state As string
State = controller.Execute("ConnectState", 1)

End If
```

4.2.2.5. OnMessage events

When an error is returned from a device when the device executes a command, the provider passes detailed error details to the client as OnMessage events of CaoController classes. The error details are the same as those that can be retrieved with the @LAST_ERROR variable.

Table 4-4 OnMessage content

Number properties	Value properties	Data type
1	Error details	VT_BSTR

4.3. Variable List

4.3.1. Controller class

Defines a list of variables that are available for each class. Variables refer to objects of CaoVariable classes.

Table 4-5 List of System Variables

Variable name	Description	Attribute	
		Get	Put
@MAKER_NAME	Get the manufacturer name.	✓	-
@VERSION	Get the provider version.	✓	-
@LAST_ERROR	Returns the last error encountered.	✓	-

4.3.1.1. @MAKER_NAME

Get the manufacturer name.

Data type

Item	Type	Description
Returned value	VT_BSTR	Get the manufacturer name.

Examples of uses

```
' Add Variable
Dim variable As CaoVariable
Variable = controller.AddVariable("@MAKER_NAME")
' Value acquisition
Dim strVal As String
StrVal = variable.value
```

4.3.1.2. @VERSION

Get the provider version.

Data type

Item	Type	Description
Returned value	VT_BSTR	Get the provider version.

Examples of uses

```
' Add Variable
Dim variable As CaoVariable
Variable = controller.AddVariable("@VERSION ")
' Value acquisition
Dim strVal As String
StrVal = variable.value
```

4.3.1.3. @LAST_ERROR

Gets the last error that occurred.

Data type

Item	Type	Description
Returned value	VT_BSTR	The hexadecimal string of the last error that occurred.

Refer to the reader/writer's manual for details of the error.

Examples of uses

```
' Add Variable
Dim variable As CaoVariable
Variable = controller.AddVariable("@LAST_ERROR")
' Value acquisition
Dim strVal As String
StrVal = variable.value
```

Table 4-6 List of User Variables

Variable name	Description	Attribute	
		Get	Put
DEVICE_VERSION	Get the firmware version of the device.	✓	-
TAG_DATA	Reads and writes RFID tags.	✓	✓
TAG_ID	Reads RFID UIIs.	✓	-
CONNECT_STATE	Gets the connection status.	✓	-

4.3.1.4. DEVICE_VERSION

Get the version of the device.

Option	Required	Description	Range of values	Default
MachineNo	✓	Equipment No.	1 - 31	-

Data type

Item	Type	Description
Returned value	VT_BSTR	Device version

Examples of uses

```
' Add Variable
Dim variable As CaoVariable
Variable = controller.AddVariable("DEVICE_VERSION", "MachineNo = 1")
' Value acquisition
Dim strVal As String
StrVal = variable.value
```

4.3.1.5. TAG_DATA

Reads data from and writes data to RFID tags.

Option	Required	Description	Range of values	Default
MachineNo	✓	Equipment No.	1 - 31	-
Addr	✓	Read/write start position	0 – 65535	-
Length	✓	Read data length ※Not used at the time of writing	1 – 65535	-

Data type

Item	Type	Description
Returned value	VT_BSTR	Read/write data. Writes can be entered in hexadecimal notation and with a length that is a multiple of 2.regardless of the length specified by the Length options.

- During reading

Examples of uses

```
' Add Variable
Dim variable As CaoVariable
Set variable = controller.AddVariable("TAG_DATA","MachineNo=1,Addr=0,Length=100")
' Value acquisition
Dim strVal As String
StrVal = variable.value
```

<Supplementation>

Reading takes about 600 ms every 400 bytes.

- At the time of writing

Examples of uses

```
' Add Variable
Dim variable As CaoVariable
Variable = controller.AddVariable("TAG_DATA","MachineNo=1,Addr=0,Length=100")
' Value acquisition
Dim strVal As String
Variable.value = "00112233445566778899AABBCCDDEEFF"
```

4.3.1.6. TAG_ID

Reads and writes UIDs for RFID tags.

Option	Required	Description	Range of values	Default
MachineNo	✓	Equipment No.	1 - 31	-

Data type

Item	Type Description		
Returned value	VT_ARRAY | VT_VARIANT		
	1	VT_UI2	Number of UID data read
	2	VT_BSTR	First scanned UID data* Empty character if no data
	3	VT_BSTR	Second scanned UID data * Empty character if no data
	4	VT_BSTR	Third scanned UID data * Empty character if no data
	5	VT_BSTR	4th UID data scanned* Empty character if no data

Examples of uses

```
' Add Variable
Dim variable As CaoVariable
Variable = controller.AddVariable("TAG_ID","MachineNo=1")
' Value acquisition
Dim arryUID As Array
ArryUID = variable.value
```

4.3.1.7. CONNECT_STATE

Returns whether the device can be connected.

Option	Required	Description	Range of values	Default
MachineNo	○	Equipment No.	1 - 31	-

Data type

Item	Type	Description
Returned value	VT_BSTR	Connectable: "OK" Connectable: "NG"

Examples of uses

```
' Add Variable
Dim variable As CaoVariable
Variable = controller.AddVariable("CONNECT_STATE","MachineNo=1")
' Value acquisition
Dim state As string
State = variable.value
```

4.4. Error-code

Table 4-7 shows the unique error codes defined by the AN provider.

Table 4-7 Error Codes

Error name	Error Number	Description
Equipment No. error	0x80100001	Out-of-range MachineNo options are specified. Set in 1-31.
Data address setting error	0x80100002	Out-of-range Addr options are specified. Set it from 0-65535.
Data length setting error	0x80100003	Out-of-range Length options are specified. Set it from 1-65535.
Destination device No error	0x80100004	The received device number differs from the device number specified as an option. Check the device or option.
Write data length error	0x80100005	Write length is not in 2byte units. Set for each 2byte.

Write data error	0x80100006	Write data contains non-hexadecimal numbers. Specify a hexadecimal string between 00 and FF.
Abnormal packet	0x80100007	FCS mismatches. Contact the equipment manufacturer.
Error from the device	0x8010FF01 0x8010FF02 0x8010FF03 0x8010FF65 0x8010FF71	An error was returned from the device when the command was executed. The last two digits indicate the command sent from the provider. Refer to Table 4-8 for the correspondence between the commands used by the provider and the device commands.

Table 4-8 Command Correspondence Table

Provider-side processing			Device-side commands
Variable	TAG_DATA	Get	0x01
		Put	0x02
	TAG_ID	Get	0x03
	CONNECT_STATE	Get	0x65
	DEVICE_VERSION	Get	0x71
Execute Method	ReadTag		0x01
	WriteTag		0x02
	ConnectState		0x65

Refer to the Error Codes section of the ORiN2 Programming Guide (Link) for additional information about ORiN2 common errors.