JRCS
SMS-55 providers
User's Guide

Version 1.0.0

November 12, 2021

NOTE:

[Revision History]

Version	Date	Content
1.0.0	2021-11-12	First edition

[Compatible models]

Model	Version	Notes
SMS-55		Use a serial Ethernet converter to connect to SMS-55 and only support UDP-connection.

[Operation check model]

Model	Version	Notes
SMS-55	--	MOXA's NPort5150 is used as a serial Ethernet converter.

目次

1. Introduction

This user's guide is for providers that receive channel data from JRCS's Ship Alarms, Monitoring and Control System (AMS,AMCS) SMS-55 devices. Fig. 1-1 shows the overall configuration of this provider and the device. The providers are referred to as SMS-55 providers.

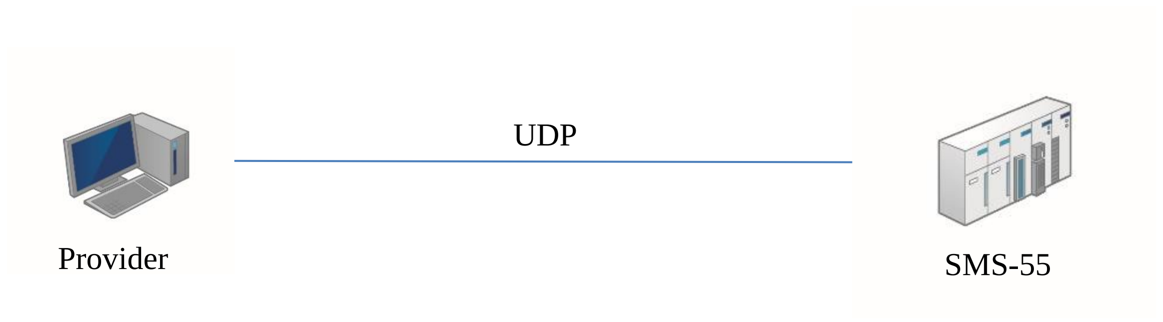

Fig. 1-1 Configuration Diagram

Fig. 1-2 shows the correspondence between this provider and each device.
(* This is an example. It does not represent everything.)

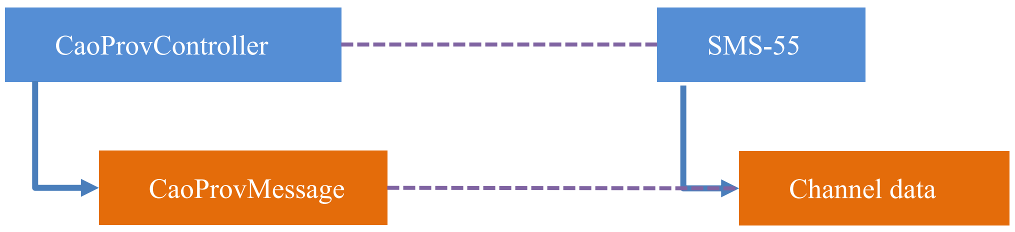

Fig. 1-2 Provider configuration and device information

1.1. Outline of SMS-55 Transmit Data

SMS-55 transmits the date and several channels of data as serial data at regular intervals. Fig. 1-3 shows an overview of the transmit data.

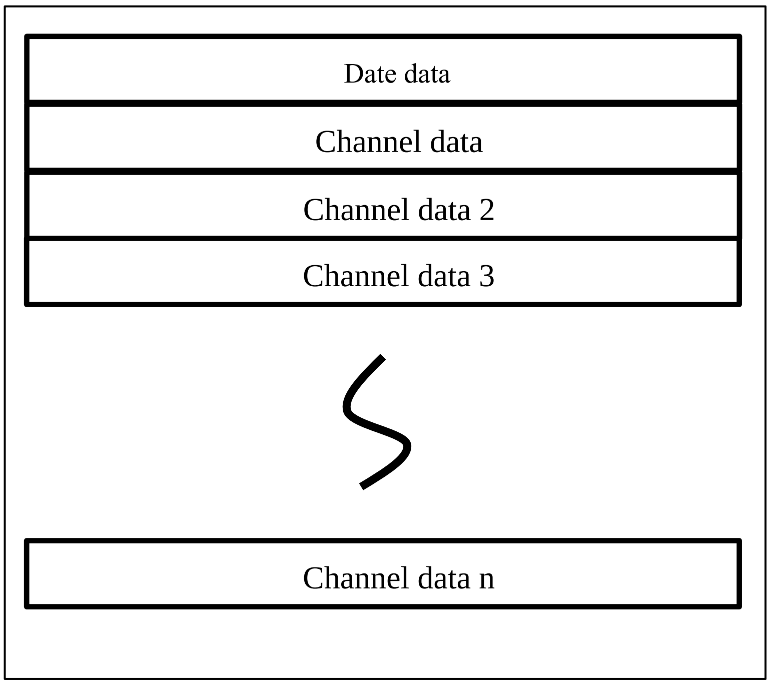

Fig. 1-3 SMS-55 Outline of Transmitted Data

1.1.1. Channel Data Overview

The channel data consists of 9 characters of data, and the first character is treated as the status and the remaining 8 characters are treated as the data. In the example in Fig. 1-4, the channel data has a status of "R" and data of "_ _ _ _ _ 9 9 9" (where "_" is a space).

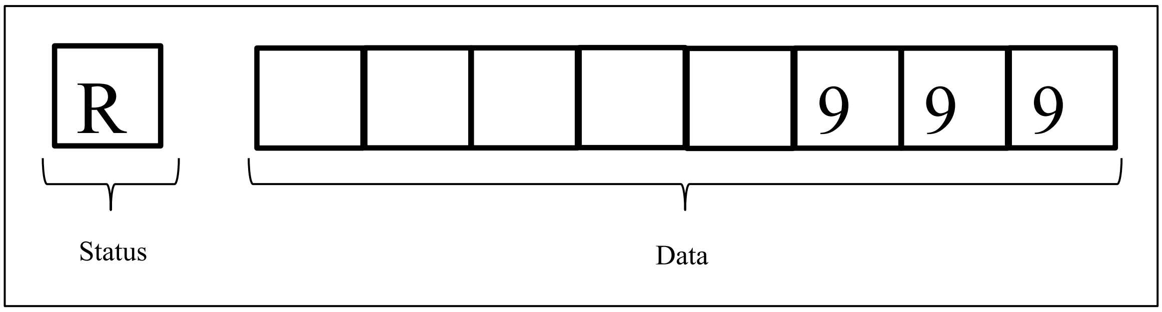

Fig. 1-4 1-channel data configuration (example)

Table 1-1 Status (Example)

Status	Description
_ (20H)	Normal data
* (2AH)	Error data
R (52H)	Stopping data

2. Setting Up Your Environment for Application Development

2.1. Connecting SMS-55 to a ClientPC

To connect to this provider from SMS-55, the serial data of RS-422 transmitted from SMS-55 must be converted to UDP. In this manual, "NPort5150" from MOXA is used as a RS-422 to UDP converter to confirm connectivity.

Because this provider operates as a UDP server, it does not judge whether the connection destination is SMS-55 or not, and it analyzes the data sent to the specified reception port.

3. Command Reference

3.1. Method/Property List

Table 3-1 List of methods and properties

Category	Methods/Properties[1]		Function	Reference
CaoWorkspace				
	AddController	M	Connected to controller	P.9
CaoController				
	Variables	P	Retrieving Variable Collections Holded by the Controller	P.10
	AddVariable	M	Adding Variable Objects	P.11
	OnMessage	E	Message reception event	P.11
CaoVariable				
	Value	P	Get/set value	P.11

3.2. Method properties

3.2.1. CaoWorkspace classes

3.2.1.1. AddController method

In CaoWorkspace, add a controller object. SMS-55 providers refer to the parameters passed when AddController method is executed and connect to SMS-55, if applicable. The following are the specifics of AddController method:

Format

```
AddController
(
"<controller name>",            // Controller name (optional)
"CaoProv.JRCS.SMS-55",          // Provider name (fixed)
"<machine name>",                      // Provider execution machine name (unused)
"<Option>"                // Option string (optional)
)
```

Option

[1] M:Indicates methods, P: properties, and E: events, respectively.

The following options are specified in the option string: The option string is a string consisting of the following options separated by a comma (,).

Option	Required	Description	Value Range
MyPort = <receiving port number>	--	Specifies the UDP-port number to receive data from SMS-55.	1-65535 Default value :1024
ReceiveWait = <Receive wait time>	--	Specifies the maximum wait time (ms) when data reception is started from SMS-55 and no data is received halfway through. ※If the specified time elapses from the last received data during data reception, a timeout error message is issued and the received data is discarded.	0-ULONG_MAX Default value :10000

Sample usage (C#)

```
// Engine
ORiN2.ManagedCAO.CCaoEngine engine = new ORiN2.ManagedCAO.CCaoEngine();
// Workspace
ORiN2.ManagedCAO.CCaoWorkspace workspace = engine.AddWorkspace("NewWrks", "");
// Controller
ORiN2.ManagedCAO.CCaoController controller= workspace.AddController("SMS-55",
                                                                    "CaoProv.JRCS.SMS-55",
                                                                    "",
                                                                    "");
```

3.2.2. CaoController classes

3.2.2.1. Variables Properties

Gets a collection of variables that the controller holds.

Sample usage (C#)

```
// Variable Collection Retrieval
ORiN2.ManagedCAO.CCaoVariables variables = controller.Variables;
// Variable acquisition
ORiN2.ManagedCAO.CCaoVariable variable = variables[0];
```

3.2.2.2. AddVariable method

Adds a variable object to CaoController. Only the variable names shown in 3.3.2 can be used.

AddVariable is specified as follows.

Format

```
AddVariable
(
    "<variable name>",          // Variable Name
    "<Option>"                  // Option string (optional)
)
```

3.2.2.3. OnMessage event

You can receive controller error notifications and status changes as OnMessage events. See 3.4 for the events that can be received.

3.2.3. CaoVariable classes

3.2.3.1. Value Properties

The behavior depends on the variable name. For details, refer to section 3.3, Variable List.

3.3. Variable list

Defines a list of variables that can be used in each class. Variables refer to objects of CaoVariable classes.

3.3.1. System Variables

Only system variables exist in this provider.

System Variables

A variable that accesses only the information in the object that holds the variable. All system variables of this provider are static data. System variables are preceded by "@".

3.3.2. CaoController class-variable

Variable Name	Description	Value		Reference
		Get	Put	
@MAKER_NAME	Obtain the manufacturer's name.	✓	-	P.12
@VERSION	Get the DLL version.	✓	-	P.12

3.3.2.1. @MAKER_NAME

Obtain the manufacturer's name.

Data type

Type Description	
VT_BSTR	Obtain the manufacturer's name.

Sample usage (C#)

```
// Add Variable
ORiN2.ManagedCAO.CCaoVariable var = controller.AddVariable("@MAKER_NAME","");
// Acquisition of Values
String value = var.Value as string;
```

3.3.2.2. @VERSION

Gets the DLL version.

Data type

Type Description	
VT_BSTR	Get the DLL version. *.*.*

Sample usage (C#)

```
// Add Variable
ORiN2.ManagedCAO.CCaoVariable var = controller.AddVariable("@VERSION","");
// Acquisition of Values
String value = var.Value as string;
```

3.4. Event list

You can receive controller error notifications and status changes as OnMessage events.

No.	Description
0	Received data Data received normally from SMS-55 is displayed.
101	Receiving timeout message
102	Receive data error An error message occurs when the received data is unexpected.
103	Date data error message An error occurs when the received date data cannot be converted.

No.	Description
104	Channel data error message An error occurs when the received 1-channel data is not composed of 9-character data.

3.4.1.1. Received data message

Issue the received packet data as a message with the date, channel data status array, and channel data array.

Value Property Data Types

Type Description			
VT_ARRAY | VT_VARIANT			
0	VT_DATE		Date
1	VT_ARRAY | VT_BSTR		Status section of the received data
	0	VT_BSTR	Status of the 0th channel data
	~	...	
	N	VT_BSTR	Status of the nth channel data
2	VT_ARRAY | VT_BSTR		Data portion of the received data
	0	VT_BSTR	0th channel data
	~	...	
	N	VT_BSTR	Nth channel data

Sample usage (C#)

```
///<summary>
/// Example of Processing Methods for Reception
///</summary>
///<param name="chData><receive-data</param>
Private void OnReceivedData(object[] chData)
{
    CDateTime date = Convert.ToDateTime(chData[0]) ;//date data
    String[] status = chData[1] as String[];//status array
    String[] data = chData[2] as String[];//data array
}
```

4. Programming by SMS-55 providers

With SMS-55 providers, you can connect SMS-55 to the client computer as follows:

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

After you connect to SMS-55, the providers issue channel data as messages that come from SMS-55. The issued message can be retrieved by the event of CaoProvMessage.

4.1. Sample programming to receive channel data

This section shows a sample program that reads and writes the value of the global OM data register for a message issued as an example. Table 4-1 describes the requirements of the sample program, and Fig. 4-1 describes the flow of the sample program.

Table 4-1 Sample program requirements

Requirements	Description
Host	Connect via UDP
Connection source	The receiving port number is 4001.
Process Description	Receive data from device
	Received data is stored for each element.

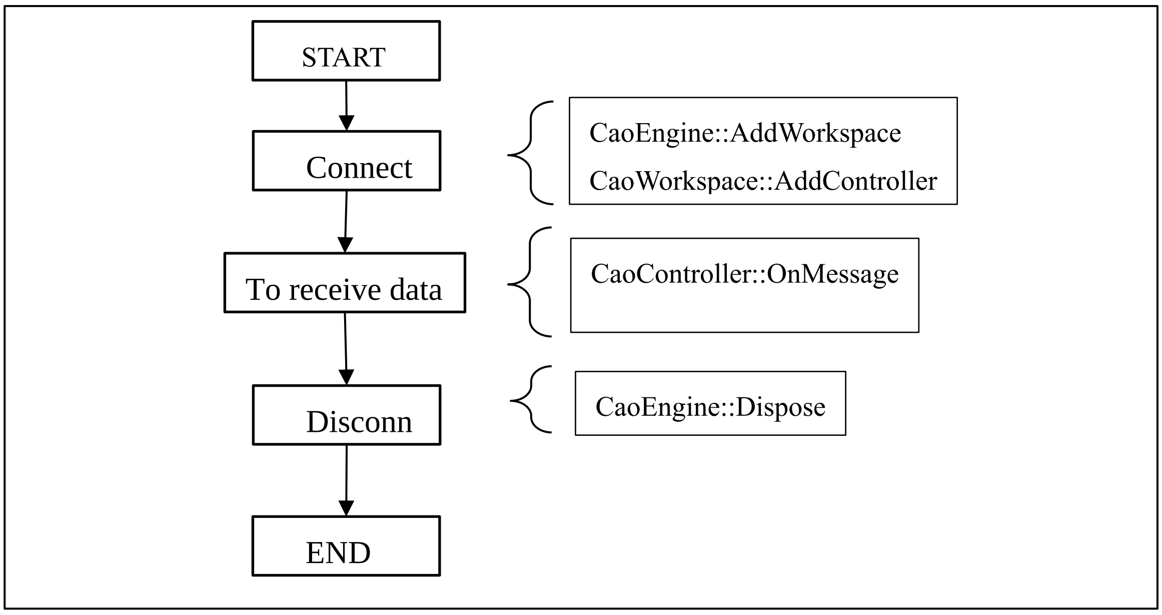

Fig. 4-1 Flow of data reception

Specific codes are given in the following sections.

4.1.1. Sample program

The following is an overview of the sample program.

Sample	SMS55Sample.cs

```
// Object
Private ORiN2.ManagedCAO.CCaoEngine m_caoEngine = null;
Private ORiN2.ManagedCAO.CCaoWorkspace m_caoWorkspace = null;
Private ORiN2.ManagedCAO.CCaoController m_caoController = null;

Public void Main()
{
    // Connection
    This.Connect();

  // Adds the action to take when a Onmessage is received on the controllers.
    m_caoController.OnMessage += new ORiN2.ManagedCAO.OnMesssageEventHandler(OnMessage
Event);
    // Wait 60 seconds to accept OnMessage events
    Thread.Sleep(60000);
    // Disconnect
    This.Disconnect();
}

// Connection method
Private void Connect()
{
    // Generate CaoEngine object
    This.m_caoEngine = new ORiN2.ManagedCAO.CCaoEngine();
    // Generate CaoWorkspace object
    This.m_caoWorkspace = this.m_caoEngine.AddWorkspace("NewWrks", "");
    // Generate CaoController object
    This.m_caoController = this.m_caoWorkspace.AddController("SMS-55",
                                                      "CaoProv.JRCS.SMS-55",
                                              "",
                                              "MyPort=4001");
}

// Disconnect method
```

```
Private void Disconnect()
{
    This.m_caoEngine.Dispose();
    This.m_caoEngine = null;
}

///<summary>
//// OnMessage reception event
///</summary>
Private void OnMessageEvent(object sender, ORiN2.ManagedCAO.OnMessageEventArgs e)
{
    If(e.Message.Number == 0)
    {
         // Get message contents
         Object[] messageData = e.Message.Value as Object[];
         // Date data
         DateTime date = Convert.ToDateTime(messageData[0]);
         // Channel data (status)
         String[] statusArray = messageData[1] as String[];
           // Channel data
           String[] statusArray = messageData[2] as String[];
    }
}
```

4.1.1.1. Connection

To connect to SMS-55, perform the following steps:

(1) Prepare a variable to hold the object.

The objects required to connect to the controller are CaoEngine object, CaoWorkspace object, and CaoController object. CaoWorkpace object does not need to have a variable to obtain CaoController object from CaoWorkspaces. You will also need a CaoVariable for accessing the variable. The following is a code example for C#.

Sample usage (C#)

```
// Variables for CaoEngine
Private ORiN2.ManagedCAO.CCaoEngine m_caoEngine = null;
```

```
// Variables for CaoWorkspace
Private ORiN2.ManagedCAO.CCaoWorkspace m_caoWorkspace = null;
// Variables for CaoController
Private ORiN2.ManagedCAO.CCaoController m_caoController = null;
```

(2) Creates a CaoEngine.

CaoEngine is generated using the New keyword.

Sample usage (C#)

```
// Generate CaoEngine object
This.m_caoEngine = new ORiN2.ManagedCAO.CCaoEngine();
```

(3) Gets or generates a CaoWorkspace container.

When you create a CaoEngine object, it defaults to generating one CaoWorkspaces object and one CaoWorkspace object. The following is a sample code/default CaoWorkspace for creating a new CaoWorkspace.

Sample usage (C#)

```
// Generate CaoWorkspace object
This.m_caoWorkspace = this.m_caoEngine.AddWorkspace("NewWrks", "");
```

(4) Creates a CaoController.

To generate a CaoController object, set the provider name to use and the parameters to use. For SMS-55 providers, specify the UDP port on which to receive data in MyPort option. The following is a code example:

Sample usage (C#)

```
// Generate CaoController object
This.m_caoController = this.m_caoWorkspace.AddController("SMS-55",
                                          "CaoProv.JRCS.SMS-55",
                              "",
                                         "MyPort=4001");
```

4.1.1.2. To receive data

(1) SMS-55 Provider issues a OnMessage event when it receives transmit data from the device. To receive a OnMessage event, set CaoController object's OnMessage event handler to the action to take when the event is received. The following is a code example:

Sample usage (C#)

```
// Adds the action to take when a Onmessage is received on the controllers.
```

```
m_caoController.OnMessage += new ORiN2.ManagedCAO.OnMesssageEventHandler(OnMessageEvent);
```

(2) When OnMessage event is issued, the event registered above is executed, and the retrieved data can be retrieved by Value property of OnMessage. Here is an example code:

Sample usage (C#)

```
///<summary>
//// OnMessage reception event
///</summary>
Private void OnMessageEvent(object sender, ORiN2.ManagedCAO.OnMessageEventArgs e)
{
        If(e.Message.Number == 0)
        {
              // Get message contents
              Object[] messageData = e.Message.Value as Object[];
              // Date data
              DateTime date = Convert.ToDateTime(messageData[0]);
              // Channel data (status)
              String[] statusArray = messageData[1] as String[];
                // Channel data
                String[] dataArray = messageData[2] as String[];
        }
}
```

4.1.1.3. Disconnect

To disconnect from the controller, you can erase the generated objects and delete the objects that you want to erase from the collection class that manages the objects. However, you do not need to explicitly remove it if you use ORiN.ManagedCAO. The following is a code example:

Sample usage (C#)

```
    // Remove all objects from CaoEngine
    This.m_caoEngine.Dispose();
    // Clear CaoEngine
    This.m_caoEngine = null;
```

5. SMS-55 Provider Error Codes

This provider has the following unique error codes masked with the 0x8011****. (Refer to Table 5-1 Unique Error Codes Table.)

For information about common ORiN2 errors, see the Error Codes section in ORiN2 Programming Guide.

Table 5-1 Unique Error Codes

Error Number	Description
0x80110001	MyPort optional error MyPort optional specification is incorrect.
0x80110002	ReceiveWait optional error ReceiveWait optional specification is incorrect.