QR Code provider

DENSO QR code scanner

Version 1.0.4

User's guide

Jan 20, 2018

[Remarks]

【Revision history】

Version	Date	Content
1.0.0.0	2006-02-23	First edition.
1.0.1.0	2007-05-21	Manual correction
1.0.1.1	2010-02-11	Error code was added
1.0.1.2	2011-09-19	Hardware was added
1.0.2.0	2012-05-10	“@STATE" variable was added.
1.0.2	2012-07-17	Document versioning rules was changed.
1.0.3	2017-04-24	Hardware was added.
1.0.4	2018-01-20	Timeout option was added.

【Hardware】

Model	Version	Notes
QS20H		
GT10Q		
QB20		
QD20		
QB30		
QD25		
GT20QD		Only the USB-COM interface (factory default) can be communicated.
GT20Q		
AT25Q		
WS1		
AT30		

Contents

1. Introduction

This document is a user's guide of the QR Code[1] provider that is the CAO provider for the QR Code scanner manufactured by DENSO.

The QR Code provider is a provider that outputs data from two dimension code scanner. Moreover, the image capture function is available used by sending the command.

This book explains the function of this QR Code provider and the mounted method.

[1] The QR cord is trademark registration of DENSO WAVE.

2. Outline of provider

2.1. Outline

There are two ways for acquiring values by the QR code scanner; a push (event) method and a pull (stock) method. In a push (event) method, the program outputs acquired values at the moment of scanning QR code. On the other hand, in a pull (stock) method, command is sent to the QR scanner to acquire values.

As a way of acquiring values by the push method, the QR code provider is equipped with queue and event. These measures can be selected when AddController is executed.

For the pull method, this provider is equipped with ReadData in variable class as a tool of acquiring values.

The file format of the QR Code provider is DLL (Dynamic Link Library). Table 2-1 lists the details

Table 2-1 QR Code provider

File name	CaoProvQR.dll
ProgID	CaoProv.DENSO.QR Code
Setting registry[2]	regsvr32 CaoProvQR.dll
Delete registry	regsvr32 /u CaoProvQR.dll

2.1.1. Operation mode

2.1.1.1. Command type mode

In the command type mode, the scanner sends command to acquire the data when values are acquired by variable “ReadData". This mode enables only when the QR Scanner is set to continuous execution mode.

2.1.1.2. Event type mode

In the event type mode, OnMessage event is issued when the QR Code Scanner acquires data. Therefore, the client can acquire values from the data that stored in the Message object acquired in this event.

2.1.1.3. Queue type mode

In the queue type mode, when data is acquired from the QR code scanner, first of all, the data is queued. (First-In First-Out buffer). The client program can acquire the value from the queue by using “@QUEUE" system variable in the arbitrary timing.

2.1.2. The Data Transmission format

The default setting of the data transmission format of the QR Code Scanner is shown in Table 2-2. Whenever the format is changed, only "BCC" is set to “Disabled”. When selecting “Permission” for the code

[2] It is not necessary manual registration/deregistration when installing it with ORiN SDK.

ID mark and the number of digits, it is assumed a part of receive data. Neither the header nor the terminator are contained in receive data.

Table 2-2 The default setting of the Data Transmission format

Parameter	Default set point
Header	None
Terminator	CR
Transmission of code ID mark	Disabled
Transmission of the number of digits	Disabled
Transmission of BCC	Disabled(fix)

2.1.3. Image scanner function

The image scanner function is available in the QR Code provider.

To use the image scanner function, sends the image scanner command by Controller::Execute method. Next, the acquired image data is sent to the client on OnMessage event. Data is stored in the Value property of the Message object as a binary array.

Moreover, even whenever the image scanner function is used when starting in the queue type mode, data is not stored in the queue and sent in the event. If the image scanner function is used during the queue type mode operation, the data is always not queued but sent as an event.

2.2. Method and property

2.2.1. CaoWorkspace::AddController method

The OR Code provider refers to the connection parameter for communication and connects the communication at the time of AddConnection. Option specifies a communication form, connection parameters, time-out time.

```
AddController
(
"< controller-name >"              // Controller-name
 "CaoProv.DENSO.QRCode ",          // Provider name. Fixed
"< machine name >"                 // Execution machine name of provider.
"< option >"                       // Option character string
)
```

Following is a list of option string items.

Table 2-3 Option character string of CaoWorkspace::AddController

Option	Meaning
Conn=< Connected parameter >	Mandatory. Set the communication form and connection parameters. For details, refer to 2.2.1.1.
Mode [=< Communication mode >]	Operation mode of provider. ‘1' Event enabled, ‘2’ Queue enabled, ‘4’-Unicord conversion. To specify these modes, OR is omittable. (default: 5) Operates as a event mode if the first bit is effective, and as a command mode when the first one bit is ineffective. Example:‘ 5' -Event enabled and convert to Unicode.
Protocol [=<Communication option >]	Specify the communication protocol, header and terminator. (default: "0:0:0") For details, refer to 2.2.1.2 .
Timeout [=<Time-out time >]	Specify time-out time for sending and receiving. (default:500 Maximum: 60000)

2.2.1.1. Conn option

Following is connection parameter strings for Conn option. The object in the square bracket (“[ ]”) is omittable. The underlined part written in the explanation of each parameter shows the default value that the case of without specifying options.

“com:<COM Port>[:<BaudRate>[:<Parity>:<DataBits>:<StopBits>]]"

<COM Port>	:	COM port number. ‘1'-COM1,‘2'-COM2, …
<BaudRate>	:	Communication speed. 4800,9600,19200,38400,57600,115200.
<Parity>	:	Parity. ‘N'-NONE, ‘E'-EVEN,‘O'-ODD.
<DataBits>	:	Number of data bits. ‘7'-7bit, ‘8'-8bit.
<StopBits>	:	Number of stop bits. ‘1'-1bit,‘ 2'-2bit.

2.2.1.2. Protocol option

Following is parameter strings for Protocol option. The object in the square bracket (“[ ]”) is omittable. The underlined part written in the explanation of each parameter shows the default value that the case of without specifying options.

“Protocol =[<Protocol>[:<Header>[:<Term>]]]"

<Protocol>	:	Communication protocol.
		‘0' - No protocol. ‘1'-ACK/NAK - protocol.
<Header>	:	Header specification.
		‘0' - None, ‘1'-STX(0x05)
<Term>	:	Terminator specification.
		‘0'-CR(0x0D), ‘1'-LF(0x0A), ‘2'-CR+LF(0x0D0A)

2.2.2. CaoController::AddVariable method

Acquire the variable object. Available variable names are the system variables listed in 2.3.1. This method fails if using the variable names which does not listed in 2.3.1.

```
AddVariable
(
"< variable name >"                    // System variable name
"< option >"                           // Option character string(unused)
)
```

2.2.3. CaoController::get_AddVariableNames property

Acquire the list of the system variable name listed in 2.3.1.

2.2.4. CaoController::Execute method

In this method, the command is sent to the QR Code Scanner. Terminator “CR" is added to the specified command and the command are sent to the QR Code Scanner. The execution result is not returned.

In this method, “RAW” is specified in the command name of the first argument, and specify the command of the QR code scanner for the parameter of the second argument. At this time, please specify the parameter by character string type.

Please refer to the manual of each QR Code Scanner for possible command.

```
Execute
(
 "Raw"                    // Command name. Fixed.
"< parameter >"           // Command character string of QR Code Scanner
)
```

Example 1) Image capture (JPEG, 1/4VGA, full-screen, without thumbnail)

caoCtrl.Execute(“Raw", “IMAGEOUT#J#2#F#0")

Example 2) Loading of setting (memory: #1)

caoCtrl.Execute(“Raw", “LOAD#1")

Example 3) Soft reset

caoCtrl.Execute(“Raw", “RESET")

2.2.5. CaoController::OnMessage event

When the QR Code provider receives data, the data is passed to the client as OnMessage event of the CaoController class. At this time, received data is stored in the Message::Value property as it is.

OnMessage event is classified three types (the event type mode, the image scanner function, and the reception error), and of each is distinguished by the Message::Number property. The correspondence of the Message::Number property and classification is shown as follows.

Table 2-4 Correspondence of classification and number of Message::Number property

Number property	Classification	Data type
1	QR code data	String
2	Image data	Binary array
3	Error event	String

2.2.6. CaoVariable::get_Value property

Acquire the value which correspond with system variables shown in 2.3.1

2.2.7. CaoVariable::put_Value property

This property is mounted only when the system variable name is “@QUEUE".

If the specified value is VT_BSTR, the data is stored in the queue.

The queue is made empty clearing the buffer if the specified value is VT_EMPTY.

2.3. Variable list

2.3.1. Controller class

Table 2-5 Controller class user variable list

Variable identifier	Data type	Explanation	Attribute	
			get	put
ReadData	VT_BSTR	Sends the reading command to the QR cord scanner then acquires data. Please set the QR Code Scanner to the continuous reading mode when you use this variable.	√	-
VerF	VT_BSTR	Acquires the version of the setting parameter of the QR Code Scanner.	√	-

Table 2-6 List of system variable of controller class

Variable identifier	Data type	Explanation	Attribute	
			get	put
@MAKER_NAME	VT_BSTR	“DENSO WAVE"	√	-
@TYPE	VT_BSTR	“QS20H"	√	-
@VERSION	VT_BSTR	Acquire the software version of the QR Code Scanner.	√	-
@QUEUE	VT_BSTR	Queue that stores scanned value	√	√
@QUEUE_SIZE	VT_I4	Current size of queue	√	-
@QUEUE_MAX	VT_I4	The maximum value size of queue Fixed value: 100	√	-
@STATE	VT_I4	Connected state 0:Disconnected 1: Connected 2: Error	√	-

2.4. Error code

In the QR Code provider, there is no peculiar error code. Please refer to the chapter of the error code of "ORiN2 Programming guide" for the ORiN2 commonness error.

3. Sample program

The sample below shows how to output the data of the QR code to the text.

List 3-1 SampleText.frm

```
Private eng As CaoEngine
Private WithEvents ctrl As CaoController

Private Sub Form_Load()

    Set eng = New CaoEngine

    'Connection procedure with QRCode scanner
    Set ctrl = eng.Workspaces(0).AddController("Sample", _
                                               "CaoProv.DENSO.QRCode", _
                                               "" _
                                               , _
                                               "Conn=com:1:38400:N:8:1, _
                                                Mode=5, _
                                                Protocol=0:0:0")
End Sub

' Data reception
Private Sub ctrl_OnMessage(ppCaoMess As CAOLib.ICaoMessage)

    text1.Text = ppCaoMess.Value

End Sub
```

The following sample shows how to save the image taken by the image scanning function in the file.

List 3-2 SampleImage.frm

```
Private eng As CaoEngine
Private WithEvents ctrl As CaoController

Private Sub Form_Load()

    Set eng = New CaoEngine

    'Connected processing with QRCode
    Set ctrl = eng.Workspaces(0).AddController("Sample", _
                                               "CaoProv.DENSO.QRCode", _
                                               "", _
                                               "Conn=com:1:38400:N:8:1, _
                                                Mode=5, _
                                                Protocol=0:0:0")
End Sub

'Transmission of image scanning command
Private Sub Command1_Click()

    ctrl.Execute "IMAGEOUT#J#2#F#0"

End Sub

'Reception of images
Private Sub ctrl_OnMessage(ByVal pICaoMess As CAOLib.ICaoMessage)
    'The picture data is stored in the temporary file.
    Open "picture.jpg" For Binary As #1
    Dim byData() As Byte
    byData = pICaoMess.Value
    Put #1, , byData
    Close #1

End Sub
```