

QR Code プロバイダ デンソー QR コードスキャナ

Version 1.0.4

ユーザーズ ガイド

Jan 20, 2018

【備考】

【改版履歴】

バージョン	日付	内容
1.0.0.0	2006-02-23	初版.
1.0.1.0	2007-05-21	マニュアル修正
1.0.1.1	2010-02-11	エラーコード追加
1.0.1.2	2011-09-19	対応機器追加
1.0.2.0	2012-05-10	@STATE 変数追加
1.0.2	2012-07-17	ドキュメントのバージョンルールを変更
1.0.3	2017-04-24	対応機種追加
1.0.4	2018-01-20	Timeout オプション仕様追加

【対応機器】

機種	バージョン	注意事項
QS20H		
GT10Q		
QB20		
QD20		
QB30		
QD25		
GT20QD		USB-COM インタフェース (工場出荷時) のみ通信可
GT20Q		
AT25Q		
WS1		
AT30		

目次

1. はじめに	4
2. プロバイダの概要	5
2.1. 概要	5
2.1.1. 動作モード	5
2.1.2. 通信フォーマット	6
2.1.3. イメージスキャナ機能	6
2.2. メソッド・プロパティ	7
2.2.1. CaoWorkspace::AddController メソッド	7
2.2.2. CaoController::AddVariable メソッド	8
2.2.3. CaoController::get_AddVariableNames プロパティ	8
2.2.4. CaoController::Execute メソッド	8
2.2.5. CaoController::OnMessage イベント	9
2.2.6. CaoVariable::get_Value プロパティ	9
2.2.7. CaoVariable::put_Value プロパティ	9
2.3. 変数一覧	10
2.3.1. コントローラクラス	10
2.4. エラーコード	10
3. サンプルプログラム	11

1. はじめに

このドキュメントは、デンソー製 QR コード¹スキャナ用の CAO プロバイダである QR Code プロバイダのユーザーズガイドです。

QR Code プロバイダは、2 次元コードスキャナからデータを出力するプロバイダです。また、コマンドを送信することでイメージキャプチャ機能を使用することもできます。

本書は、この QR Code プロバイダの機能と実装されているメソッドについて説明します。

¹ QR コードはデンソーウェーブの登録商標です。

2. プロバイダの概要

2.1. 概要

QR コードスキャナで値を取得する方法として、QR コードをスキャンした時点でその値を出力するプッシュ(イベント)方式と、QR コードスキャナにコマンドを送信し値を取得するプル(ストック)方式があります。

QR Code プロバイダではプッシュ方式で値を取得する手段として、キューとイベントを実装しており、AddController 時に選択することができます。

またプル方式で値を取得する手段として、変数クラスに ReadData を実装しています。

QR Code プロバイダのファイル形式は DLL(Dynamic Link Library)となっており、その詳細は表 2-1 のようになっています。

表 2-1 QR Code プロバイダ

ファイル名	CaoProvQR.dll
ProgID	CaoProv.DENSO.QR Code
レジストリ登録 ²	regsvr32 CaoProvQR.dll
レジストリ登録の抹消	regsvr32 /u CaoProvQR.dll

2.1.1. 動作モード

2.1.1.1. コマンド型モード

コマンド型モードでは、変数“ReadData”で値を取得時に、コマンドを送信してデータを取得します。このモードは本体が連続実行モードに設定されているときのみ使用することができます。

2.1.1.2. イベント型モード

イベント型モードでは QR コードスキャナがデータを取得したときに OnMessage イベントを発生させます。このためクライアントは、このイベントで取得した Message オブジェクトの中に格納されているデータから値を取得することができます。

2.1.1.3. キュー型モード

キュー型モードでは、QR コードスキャナからデータを取得したら、まずはキュー(先入れ先出しバッファ)にキューイングします。クライアントプログラムは、“@QUEUE”システム変数を使って任意のタイミングでキューから値を取得することができます。

² ORiN SDK でインストールした場合は手動で登録/抹消する必要はありません。

2.1.2. 通信フォーマット

QR コードスキャナのデフォルトの通信フォーマットを表 2-2 に示します。フォーマットを変更する場合、「BCC」だけは 必ず禁止 に設定します。コードマーク、桁数を「許可」した場合は、受信データの一部とされます。また、ヘッダやターミネータは受信データには含まれません。

表 2-2: デフォルトの通信フォーマット

パラメータ	デフォルト設定値
ヘッダ	なし
ターミネータ	CR
コードマークの転送	禁止
桁数の転送	禁止
BCC の転送	禁止 (固定)

2.1.3. イメージスキャナ機能

QR Code プロバイダではイメージスキャナ機能を使用することができます。

イメージスキャナ機能を使用するには、Controller::Execute メソッドでイメージスキャナコマンドを送信します。この後、取り込んだ映像データは OnMessage イベントでクライアントに送信されます。データは Message オブジェクトの Value プロパティにバイナリ配列として格納されています。

また、キュー型モードで起動しているときにイメージスキャナ機能を使用しても、データはキューには格納されず 必ずイベントで送信されます。

2.2. メソッド・プロパティ

2.2.1. CaoWorkspace::AddController メソッド

QR Code プロバイダでは AddController 時に、通信用の接続パラメータを参照し、通信の接続を行います。

このときオプションで通信形態、接続パラメータ、タイムアウトの設定を指定します。

```
AddController
(
    "<コントローラ名>",           // コントローラ名
    "GaoProv. DENSO. QRCode ",   // プロバイダ名. 固定.
    "<マシン名>",               // プロバイダの実行マシン名.
    "<オプション>"              // オプション文字列
)
```

以下にオプション文字列に指定するリストを示します。

表 2-3 CaoWorkspace::AddController のオプション文字列

オプション	意味
Conn=<接続パラメータ>	必須. 通信形態とその接続パラメータを設定します. 詳細は 2.2.1.1 に示します.
Mode[=<通信モード>]	プロバイダ動作モード. ‘1’-イベント有効, ‘2’-キュー有効, ‘4’-Unicode 変換. OR をとって指定できます. (デフォルト:5) 1 ビット目が有効のときはイベントモード, 無効のときはコマンドモードとして動作します. 例: ‘5’-イベント有効かつ Unicode 変換します.
Protocol[=<通信オプション>]	通信時のモードやヘッダ, ターミネータの設定をします. (デフォルト:“0:0:0”) 詳細は 2.2.1.2 に示します.
Timeout[=<タイムアウト時間>]	送受信時のタイムアウト時間を指定します. (デフォルト:500 最大値:60000)

2.2.1.1. Conn オプション

以下に Conn オプションの接続パラメータ文字列を示します。ここで角括弧("[]")内は省略可能を示します。また、各パラメータの解説中の下線部はオプションを指定しなかったときのデフォルト値を示します。

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

<COM Port> : COM ポート番号. ‘1’-COM1, ‘2’-COM2, ...
 <BaudRate> : 通信速度. 4800, 9600, 19200, 38400, 57600, 115200.
 <Parity> : パリティ. ‘N’-NONE, ‘E’-EVEN, ‘O’-ODD.
 <DataBits> : データビット数. ‘7’-7bit, ‘8’-8bit.
 <StopBits> : ストップビット数. ‘1’-1bit, ‘2’-2bit.

2.2.1.2. Protocol オプション

以下に Protocol オプションのパラメータ文字列を示します。ここで角括弧("[]")内は省略可能を示します。また、各パラメータの解説中の下線部はオプションを指定しなかったときのデフォルト値を示します。

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

<Protocol> : 通信プロトコル.
‘0’-ノープロトコル, ‘1’-ACK/NAK プロトコル.
 <Header> : ヘッダ指定.
‘0’-なし, ‘1’-STX(0x05)
 <Term> : ターミネータ指定.
‘0’-CR(0x0D), ‘1’-LF(0x0A), ‘2’-CR+LF(0x0D0A)

2.2.2. GaoController::AddVariable メソッド

変数オブジェクトを取得します。このメソッドで指定できる変数名は 2.3.1 に示しているシステム変数のみです。これ以外の変数名を使用した場合、このメソッドは失敗します。

```
AddVariable
(
    "<変数名>",                // システム変数名
    "<オプション>"            // オプション文字列（未使用）
)
```

2.2.3. GaoController::get_AddVariableNames プロパティ

2.3.1 に示しているシステム変数名の一覧を取得します。

2.2.4. GaoController::Execute メソッド

このメソッドでは、QR コードスキャナにコマンドを送信します。このとき指定したコマンドには、ターミネータ“CR”を追加して QR コードスキャナに送信します。実行結果は返しません。

このメソッドでは、第 1 引数のコマンド名に“Raw”，第 2 引数のパラメータに QR コードスキャナのコマンドを

指定します。このときパラメータは必ず文字列型で指定してください。

指定可能なコマンドの種類は、各 QR コードスキャナのマニュアルを参照してください。

```
Execute
(
    "Raw"                // コマンド名。固定。
    "<パラメータ>"      // QR コードスキャナのコマンド文字列
)
```

例1: イメージキャプチャ(JPEG 形式, 1/4VGA, 画面全体, サムネイルなし)

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

例2: 設定のロード(メモリ: #1)

```
caoCtrl.Execute("Raw", "LOAD#1")
```

例3: ソフトリセット

```
caoCtrl.Execute("Raw", "RESET")
```

2.2.5. CaoController::OnMessage イベント

QR Code プロバイダがデータを受信すると、CaoController クラスの OnMessage イベントとしてクライアントにデータを受け渡します。このとき、Message::Value プロパティに受信データをそのまま格納します。

OnMessage イベントは、イベント型モード、イメージスキャナ機能、受信エラーの 3 通りがあり、それぞれは Message::Number プロパティにより区別されます。以下に Message::Number プロパティとデータ種別の対応を示します。

表 2-4 Message::Number プロパティの値とデータ種別の対応

Number プロパティ	データ種別	データ型
1	QR コードデータ	文字列
2	イメージデータ	バイナリ配列
3	エラーイベント	文字列

2.2.6. CaoVariable::get_Value プロパティ

2.3.1 に示したシステム変数に対応した値を取得します。

2.2.7. CaoVariable::put_Value プロパティ

システム変数名が、“@QUEUE”のときのみ実装されています。

指定した値が VT_BSTR ならば、そのデータをキューに格納します。

指定した値が VT_EMPTY ならば、バッファをクリアして、キューを空の状態にします。

2.3. 変数一覧

2.3.1. コントローラクラス

表 2-5 コントローラクラス ユーザ変数一覧

変数名	データ型	説明	属性	
			get	put
ReadData	VT_BSTR	QR コードスキャナに読み込みコマンドを送信し、データを取得します。 この変数を使用するときには、QR コードスキャナを連続読み取りモードに設定してください。	○	-
VerF	VT_BSTR	QR コードスキャナの設定パラメータのバージョンを取得します。	○	-

表 2-6 コントローラクラス システム変数一覧

変数名	データ型	説明	属性	
			get	put
@MAKER_NAME	VT_BSTR	“DENSO WAVE”	○	-
@TYPE	VT_BSTR	“QS20H”	○	-
@VERSION	VT_BSTR	QR コードスキャナのソフトウェアバージョンを取得します。	○	-
@QUEUE	VT_BSTR	スキャンした値を格納するキュー	○	○
@QUEUE_SIZE	VT_I4	キューの現在サイズ	○	-
@QUEUE_MAX	VT_I4	キューの最大値サイズ 固定値:100	○	-
@STATE	VT_I4	接続状態 0:切断状態, 1:接続状態, 2:エラー	○	-

2.4. エラーコード

QR Code プロバイダでは、固有のエラーコードはありません。ORiN2 共通エラーについては、「[ORiN2 プログラミングガイド](#)」のエラーコードの章を参照してください。

3. サンプルプログラム

以下に QR コードのデータをテキストに出力する場合のサンプルを示します。

List 3-1 SampleText.frm

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

Private Sub Form_Load()

    Set eng = New CaoEngine

    ' QR Code との接続処理
    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

' データの受信
Private Sub ctrl_OnMessage(ppCaoMess As CAOLib.ICaoMessage)

    text1.Text = ppCaoMess.Value

End Sub
```

以下にイメージスキャン機能で取り込んだ画像をファイルに保存する場合のサンプルを示します。

List 3-2**SampleImage.frm**

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

Private Sub Form_Load()

    Set eng = New CaoEngine

    ' QR Code との接続処理
    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

' イメージスキャンコマンドの送信
Private Sub Command1_Click()

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

End Sub

' 画像の受信
Private Sub ctrl_OnMessage(ByVal pICaoMess As CAOLib.ICaoMessage)
    ' 画像データを一時ファイルに保存
    Open "picture.jpg" For Binary As #1
    Dim byData() As Byte
    byData = pICaoMess.Value
    Put #1, , byData
    Close #1
End Sub
```