



TELEDYNE DALSA
Everywhereyoulook™

Sherlock 7 Technical Resource

Teledyne DALSA Corp., Industrial Products (IPD)
www.goipd.com 978.670.2002 (U.S.A.)

August 9, 2012

Decoding QR Codes



QR Code is a matrix code created by the Japanese corporation Denso-Wave in 1994. The "QR" is derived from "Quick Response", as the creator intended the code to allow its contents to be decoded at high speed. QR Codes are common in Japan.

Barcode QR returned string

The Barcode QR algorithm returns a character string in which each group of three characters contains

- 1) the hexadecimal code corresponding to the numeric value of ONE non-ASCII barcode character. This hexadecimal code is an ASCII representation (in two consecutive characters) of a number - which naturally can't be put as such in a string.
- 2) an ASCII space.

For example, if the QR code contains the string "AB", the Barcode QR algorithm returns the character string "63|64| " or "63|64|<space>". (Remember that this is a character string, not an array of individual characters or decimal numbers!)

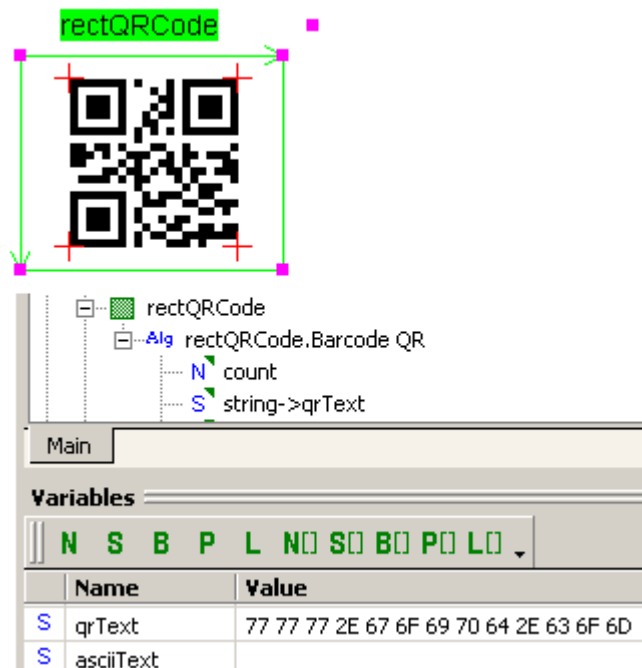
6	3	<space>	6	4	<space>
63 = JIS8 'A'			64 = JIS8 'B'		
63 = UTF-16 'A'			or	64 = UTF-16 'B'	

The string returned by the Barcode' QR algorithm must be decoded such that each group of three characters is converted to a single (non-ASCII) character. The easiest way to do this is with a JavaScript code module.

Sherlock code

In the Sherlock investigation, create two string variables, **qrText** and **asciiText**.

Save the **string** reading of the Barcode QR algorithm to **qrText**.



JavaScript code

Add a JavaScript code module after the ROI containing the Barcode QR algorithm, and type this code into it.

```
// Get the Barcode QR string
srcText = Vars.qrText

// Verify that the string length is a multiple of 3
if((srcText.length % 3) == 0)
{
    // Create an empty string to contain the trimmed, un-decoded code
    trimmedText = new String

    // Create an empty string to contain the decoded code
    destText = new String

    // Get the ASCII/UTF-8 portion of the string in trimmedText
    for (srcIdx = 0; srcIdx < srcText.length; srcIdx += 3)
    {
        trimmedText += srcText[srcIdx]
        trimmedText += srcText[srcIdx + 1]
        nonPrintableCharCode = srcText.charCodeAt(srcIdx + 2)
        Sherlock.Monitor("Non-printable char code is " + nonPrintableCharCode)
    }

    // From every two characters make an integer and interpret it as ASCII code
    for (srcIdx=0, dstIdx=0; srcIdx<trimmedText.length; srcIdx+=2, dstIdx++)
    {
        charCode = parseInt(trimmedText.substring(srcIdx, srcIdx + 2), 16)
        destText += String.fromCharCode(charCode)
    }

    // Write the decoded code to Sherlock
    Vars.asciiText = destText
}
else
{
    Sherlock.Monitor("Code length is not a multiple of 3 - cannot decode.")
}
```

Variables	
N S B P L N S B P L	
Name	Value
qrText	77 77 77 2E 67 6F 69 70 64 2E 63 6F 6D
asciiText	www.goipd.com

7	7	sp*	7	7	sp	7	7	sp	2	E	sp	6	7	sp	6	F	sp	6	9	sp	etc.
77	'w'		77	'w'		77	'w'		2E	'.'		67	'g'		6F	'o'		69	'i'		
*sp = space																					