QEBitStatus Widget

Simin Chen

26rd August 2019

Contents

Introduction

This document describes in detail the QBitStatus widget which is not an EPICS aware widget, and QEBitStatus widget which inherits from QBitStatus and is an EPICS aware widget.

This document was created as a separate widget specification document. The main reason for this is ease of maintenance and avoiding editing large and unwieldly word documents.

QBitStatus

Description

The QBitStatus widget is used to present a selected set of bits from a data word. It is not EPICS aware. See examples in Figure 1 QBitStatus widget examples below.

Bits are presented as an array of rectangles or circles with presentation properties to control shape, size, orientation, spacing and colour. Other properties allow bit by bit selection of what values display as 'on' and 'off' and if bits are rendered when 'on' or 'off'.

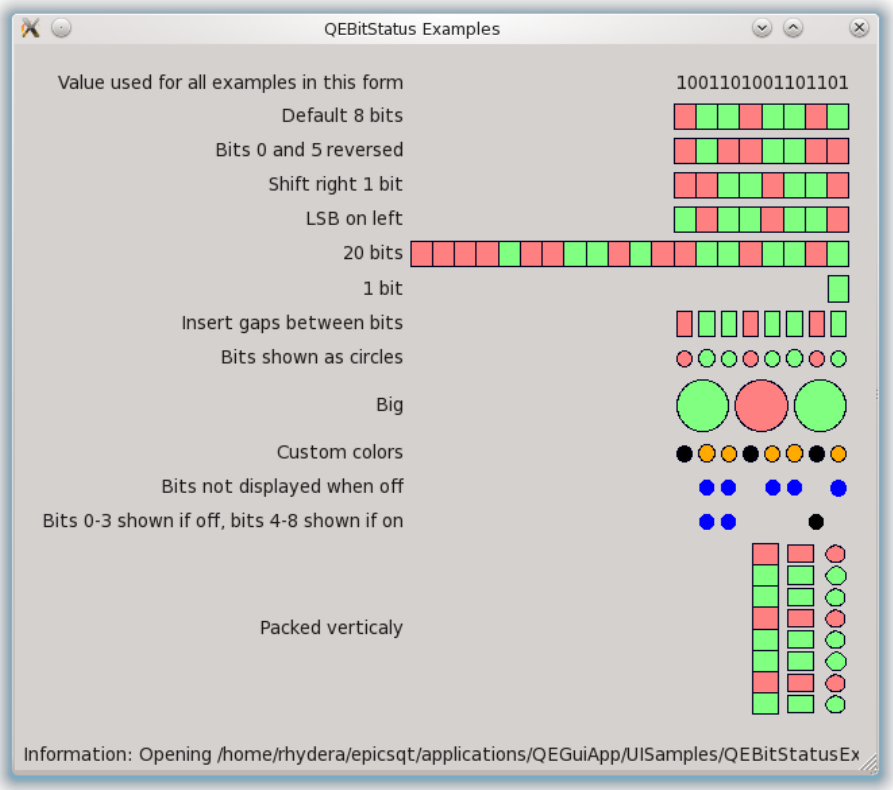


Figure 1 QBitStatus widget examples

Properties

The QBitStatus widget has the following class specific properties.

value : int

Allowed range: -2**31 to 2**31-1

Default value: 0

The value associated with the widget.

numberOfBits : int

Allowed range: 1 to 32

Default value: 8

The number of indicators shown on GUI.

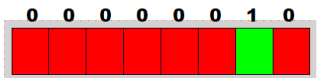
shift : int

Allowed range: 0 to 31

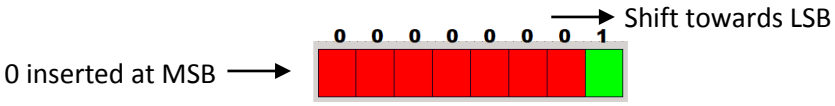
Default value: 0

The number of bits that shift the value (in binary format) towards the Least Significant Bit (LSB). 0 is inserted at the Most Significant Bit (MSB) and the old LSB is lost.

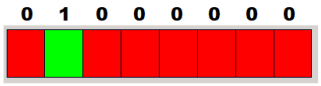
Example case 1, the value = 2 (decimal) = 0000 0010 (binary), orientation is LSB_On_Right, the other setting is default, the indicators are



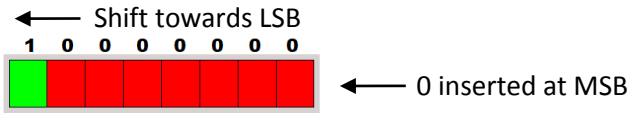
If the shift is set to 1, the value is bit-shifted towards LSB for 1 bit. The indicators become



Example case 2, the PV value = 2 (decimal) = 0000 0010 (binary), orientation is LSB_On_Left , the other setting is default, the indicators are



If the shift is 1, the indicators become



Orientation : Orientations

Allowed values: LSB_On_Right, LSB_On_Bottom, LSB_On_Left, LSB_On_Top

Default value: LSB_On_Right

The orientation of the presentation of the value.

shape : Shapes

Allowed values: Rectangle, Circle.
Default value: Rectangle

The shape of the indicators. The available options includes Rectangle and Circle.

gap : int

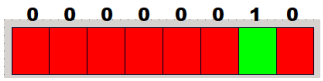
Allowed range: 0 to 80
Default value: 0
The gap between indicator array members.

reversePolarityMask : QString

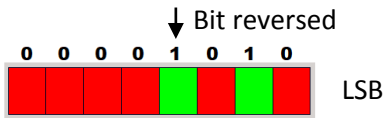
Default value: 00-00-00-00

The string is interpreted as a hexadecimal value to invert the value bits. The bits count from the LSB so the orientation definition will affect the result. The reversion applies after the bit shift.

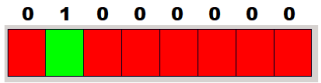
Example case 1, the value = 2 (decimal) = 0000 0010 (binary) after bit shifting, orientation is LSB_On_Right, the other setting is default, the indicators are



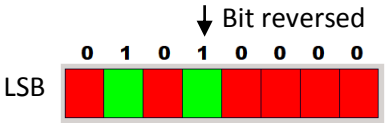
If reversePolarityMask is set to 00-00-00-08 (hexadecimal) = 0000 … 0000 1000 (binary), then the indicators are



Example case 2, the PV value = 2 (decimal) = 0000 0010 (binary) after bit shifting, orientation is LSB_On_Left, the other setting is default, the indicators are



If reversePolarityMask is set to 00-00-00-08 (hexadecimal) = 0000 … 0000 1000 (binary), then the indicators are

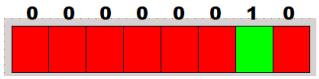


onClearMask : QString

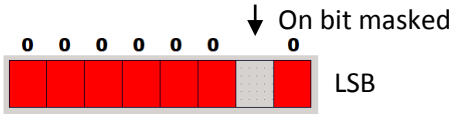
Default value: 00-00-00-00

The string is interpreted as a hexadecimal value to mask (hide) the on bits of the indicators. The masked indicators show the colour defined in clearColour. It doesn't affect the off bits. The bits count from the LSB so the orientation definition will affect the result. The mask applies after the bit shift.

In case 1, the PV value = 2 (decimal) = 0000 0010 (binary) after bit shifting, orientation is LSB_On_Right, the other setting is default, the indicators are

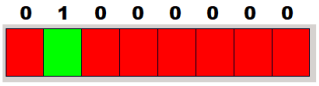


If the onClearMask is set to 2 (decimal) = 0000 0010 (binary), then the indicators become

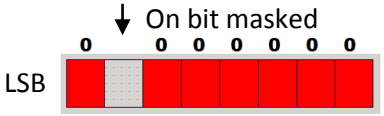


While if the onClearMask is set to other values except 2 (decimal), there is no change on the indicators as they are off bits.

In case 2, the PV value = 2 (decimal) = 0000 0010 (binary) after bit shifting, orientation is LSB_On_Left, the other setting is default, the indicators are



If the onClearMask is set to 2 (decimal), then the indicators become



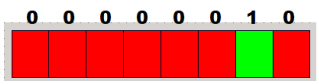
While if the onClearMask is set to other values except 2 (decimal), there is no change on the indicators as they are off bits.

offClearMask : QString

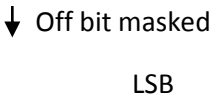
Default value: 00-00-00-00

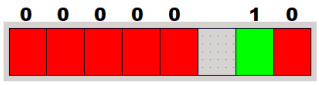
The string is interpreted as a hexadecimal value to mask (hide) the off bits of the indicators. The masked indicators show the colour defined in clearColour. It doesn't affect the on bits. The bits count from the LSB so the orientation definition will affect the result. The mask applies after the bit shift.

In case 1, the PV value = 2 (decimal) = 0000 0010 (binary) after bit shifting, orientation is LSB_On_Right, the other setting is default, the indicators are



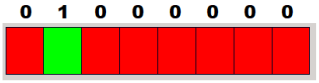
If the offClearMask is set to 4 (decimal) = 0000 0100 (binary), then the indicators become



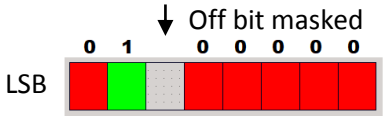


While if the offClearMask is set to 2 (decimal), there is no change on the indicators as it is on bit.

In case 2, the PV value = 2 (decimal) = 0000 0010 (binary) after bit shifting, orientation is LSB_On_Left, the other setting is default, the indicators are



If the offClearMask is set to 4 (decimal) = 0000 0100 (binary), then the indicators become



While if the offClearMask is set to 2 (decimal), there is no change on the indicators as it is on bit.

boarderColour : QColor
Default value: [0, 0, 32]

This property specifies the border colour.

invalidColour : QColor
Default value: [255, 255, 255]

This property specifies the colour to use when the value is deemed invalid.

onColour : QColor
Default value: [0, 255, 0]

This property specifies the colour when a bit is set to 1.

offColour : QColor
Default value: [255, 0, 0]

This property specifies the colour when a bit is set to 0.

clearColour : QColor
Default value: [192, 192, 192, 0]

This property specifies the colour when a bit is set clear.

drawBorder : bool
Default value: true

Show the indicator border.

QEBitStatus

Description

The QEBitStatus widget is used to present a selected set of bits from a single scalar Process Variable (PV). It is based on QBitStatus and is EPICS aware.

Properties

The QEBitStatus inherits directly from QBitStatus and as such inherits all the properties. The widget has the following additional class specific properties.

variable : QString

Refer to QEWidgetSpecifications.docx page 44 for details.

variableSubstitutions : QString

Refer to QEWidgetSpecifications.docx page 44 for details.

elementsRequired : int

Default value: 0

Refer to QEWidgetSpecifications.docx page 46 for details.

arrayIndex : int

Default value: 0

Refer to QEWidgetSpecifications.docx page 47 for details.

variableAsToolTip : bool

Default value: true

Refer to QEWidgetSpecifications.docx page 47 for details.

allowDrop : bool

Default value: false

Refer to QEWidgetSpecifications.docx page 47 for details.

visible : bool

Default value: true

Refer to QEWidgetSpecifications.docx page 47 for details.

messageSourceId : unsigned int

Default value: 0

Refer to QEWidgetSpecifications.docx page 48 for details.

defaultStyle : QString
Style Sheet string to be applied before, i.e. lower priority than, any other style, e.g. alarm style and/or user level style.

userLevelUserStyle : QString
Refer to QEWidgetSpecifications.docx page 48 for details.

userLevelScientistStyle : QString
Refer to QEWidgetSpecifications.docx page 48 for details.

userLevelEngineerStyle : QString
Refer to QEWidgetSpecifications.docx page 48 for details.

userLevelVisibility : UserLevels
Refer to QEWidgetSpecifications.docx page 48 for details.

userLevelEnabled : UserLevels
Refer to QEWidgetSpecifications.docx page 49 for details.

displayAlarmStateOption : DisplayAlarmStateOptions
Refer to QEWidgetSpecifications.docx page 49 for details.