Note: This version and the previous one were proved only for Windows Phone 7.x. Version of .NET: 4!

4,8cm

480 px

8cm

800 px

x

y

Example

This explanation will help while understanding how TutorialMessage (that’s to say the cloud-cartoon that you can see on the left) is positioned in relation to one ApplicationBarIconButton!!!!!

# Formula per convertire i centimetri in pixel

In this formula, “cm” is the measure in centimeter of the component (naturally it is graphically displayed) of which you want to find the measure in pixel. The formula returns the measure in pixel of the graphical component.

# Hyphotesis

The following one is a general ApplicationBar. So the following formula is generalized:

b

b

k

k

k

d

d

d

d

SW

* SW -> ScreenWidth in pixel. Obtained with Application.Current.Host.Content.ActualWidth
* d -> dimentions of a single ApplicationBarIconButton. Note: width = height. Got throw constructor
* k -> distance in pixel between two ApplicationBarIconButtons
* b -> I am trying to get this, the distance between the screen left side and the first ApplicationBarIconButton
* nIcons (requested in constructor) -> the number of icons. Got with ApplicationBar.Buttons.Count
* indexIcon (requested in constructor) -> the index of the icon that TutorialMessage should point.   
  1 <= indexIcon <= 4

“k” is allways 0.4 cm. Applying the formula on the upper-side I obtain:

# Trying point P(α, β)

# P(α, β)

Example

b

b

k

k

k

d

d

d

d

I will use the following equation in order to retrieve “b”:

Now I will calculate point P(α, β):

In order to calculate β, I will consider that:

Example

0.7cm

Converting 0.7cm to pixel I obtain

So the point P is P(; 70)

# Finding Q, i.e. vertical- and horizontalOffset of TutorialMessage

To position TutorialMessage, based on Popup, I will need to calculate Q(x;y). It depends on P(α;β). To understand it, the following one is an example of TutorialMessage that point to a ApplicationBarIconButton:

So I will calculate Q(x;y):

# Q(x,y)

# P(α, β)

Example

Note: in the previous formula, “TMWidth” is the width of TutorialMessage, requested in its constructor.

Note: in the previous formula, “TMHeight” is the height of TutorialMessage, that’s to say the Height of the Popup, that’s to say the Height of the Grid, that’s to say the Height of the Image of the green cloud-cartoon. It can be obtained by using “img.ActualHeight” (“img.Height” can not be used because height is request before it is shown on display) or with the following method:

So

Note: in the previous formula, you should consider “TutorialMessage/Immagini/verde\_centro.png” image. Its pixel height is 215 and its pixel width is 398. But the developer can choose the Width (“Larghezza” in the class TutorialMessage.cs) of TutorialMessage. So you should consider “TMWidth” as “Larghezza”!

So Q( ; )

But Popup, base of TutorialMessage, only supports Vertical- and HorizontalOffset. To make you understand the following formula, consider the following two images. The first one shows what I found with point Q(x;y) (i.e. it represents the distance of the same point Q(x;y) from the botton-side of the screen); the second one shows what Popup needs (Offsets, that’s to say the distance of Q(x;y) from the top-side of the screen).

Example

I have to find the distance shown in the second image starting from the first image. I will name the distances of the second image “HOffset” and “VOffset”.

# Q(x,y)

Example

Note: in the previous formula “SH” is the screen height in pixel, obtained with Application.Current.Host.Content.ActualHeight.

Note: I applicated a corrent of -10 to VOffset in the code to show the green arrow on the bottom-side of TutorialMessage.

# Porting TutorialMessage to Windows Phone 8.x

So how to switch these equations to Windows Phone 8.x, considering that many screen dimensions are possible? The easiest way is to ask the developer to convert ApplicationBar to a simple group of images.  
But in proportions…

## Converting “k”

Let’s start with “k”, described on page 2. It will always measure 0.4cm on WP7.x devices (i.e. 480x800px screens). I convert it to pixel: 40 px. “k” will became “δ”.

Proportion in general:

Note: “SW” is the screen width of the current WP8.x device.

## Converting “b”

Now I will convert “b”, the distance between the left-side and the first ApplicationBarIconButton. “b” will become “µ”.

## Converting the formula for WP7.x to a general formula

About VerticalOffset:

About HorizontalOffset: