

Tastenbetätigung mittels PowerBasic Kurzbeschreibung

26. 5. 2015

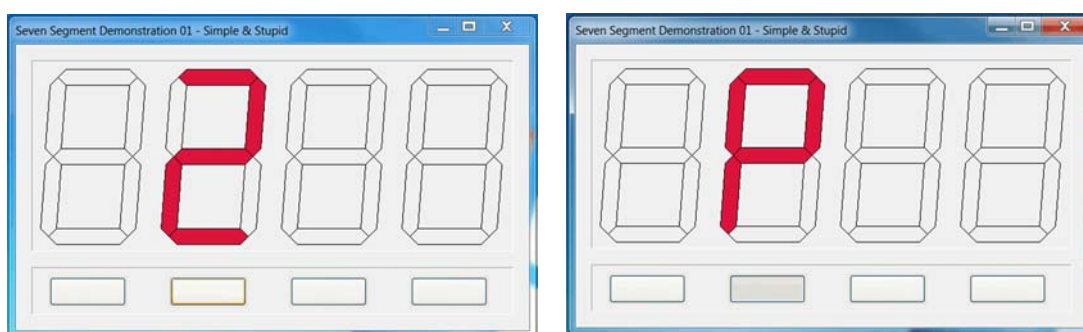
Ziel:

Das Erfassen von Tastenbetätigungen in Analogie zur direkt abgefragten mechanischen Taste (Aktivierung für die Dauer des Tastendrucks).

Prinzip:

Keine Tasten als Steuerelemente (Buttons), sondern als Bitmaps in Graphikfenstern.

Demonstrationsprogramm mit Siebensegmentanzeige:



Das Überstreichen eines Tastensymbols bewirkt, daß die Tastennummer (1 bis 4) in der jeweils darüber liegenden Siebensegmentstelle erscheint. Solange die Taste betätigt ist, wird ein “P” (für “Pressed”) dargestellt.

Auswertung der Mausereignisse:

- WM_MOUSEMOVE: die Maus wurde bewegt
- WM_LBUTTONDOWN: die linke Maustaste wurde betätigt.
- WM_LBUTTONUP: die linke Maustaste wurde losgelassen.

Die Ereignisse liefern die Koordinaten des Mausursors im Nachrichtenparameter lParam:

HIWORD		LOWORD	
31	16	15	0
Y-Koordinate (yPos)		X-Koordinate (xPos)	

Die Koordinaten beziehen sich auf den Ursprung (Punkt 0, 0 = links oben) des jeweiligen Graphikfensters (Client Area).

Für jede Taste sind drei Bitmaps erforderlich, die folgenden Zuständen zugeordnet sind:

1. Ruhezustand (Released Button). Nicht vom Mauscursor überstrichen.
2. Berührt (Selected Button). Vom Mauscursor überstrichen, aber Maustaste nicht betätigt.
3. Betätigt (Depressed Button). Vom Mauscursor überstrichen und Maustaste betätigt.

Wirkprinzip:

Tritt eines der genannten Mausereignisse auf, ist zu prüfen, ob der Mauscursor auf einer der Tasten-Bitmaps zu liegen gekommen ist oder nicht.

Der Mauscursor befindet sich dann auf einer Tasten-Bitmap, wenn gilt:

$\text{Maus_X} > \text{X-Bitmap}$ UND $\text{Maus_X} < \text{X-Bitmap} + \text{Bitmap-Länge}$ UND
 $\text{Maus_Y} > \text{Y-Bitmap}$ UND $\text{Maus_Y} < \text{Y-Bitmap} + \text{Bitmap-Höhe}$

X_Bitmap, Y_Bitmap = Startpunkt (links oben) des Bitmaps im Graphikfeld.

Als Reaktion auf ein Mausereignis sind die Koordinaten alle betreffenden Bitmaps mit den Mauskoordinaten zu vergleichen.

Ergibt sich kein Treffer, sind alle Tasten als losgelassen (Released) darzustellen.

Ergibt sich ein Treffer, sind alle anderen Tasten als losgelassen (Released) darzustellen. Handelt es sich um ein Ereignis WM_LBUTTONDOWN, ist die Taste als betätigt (Depressed) darzustellen, in den anderen Fällen als berührt (Selected).

Jede Taste hat eine Zustandsvariable:

- 0 = Ruhezustand; nicht berührt.
- 1 = berührt, aber noch nicht betätigt worden.
- 2 = betätigt gewesen, aber losgelassen.
- 3 = betätigt.

Tastenbeschriftung:

Variante 1:

Die Beschriftung gehört zu den Tasten-Bitmaps.

Variante 2:

Die Beschriftung wird live in die Tasten-Bitmaps eingetragen.

Der Ablauf:

1. Die Tasten-Bitmap ins Graphikfenster kopieren.
 2. Den Font auswählen.
 3. Den Text positionieren
 4. Den Text eintragen.
-
1. GRAPHIC COPY released,0 TO (%x_button_4,%y_button_4)
 2. GRAPHIC SET FONT hFont1
 3. GRAPHIC SET POS (%x_button_4+10,%y_button_4+5)
 4. GRAPHIC PRINT "KEY 4"

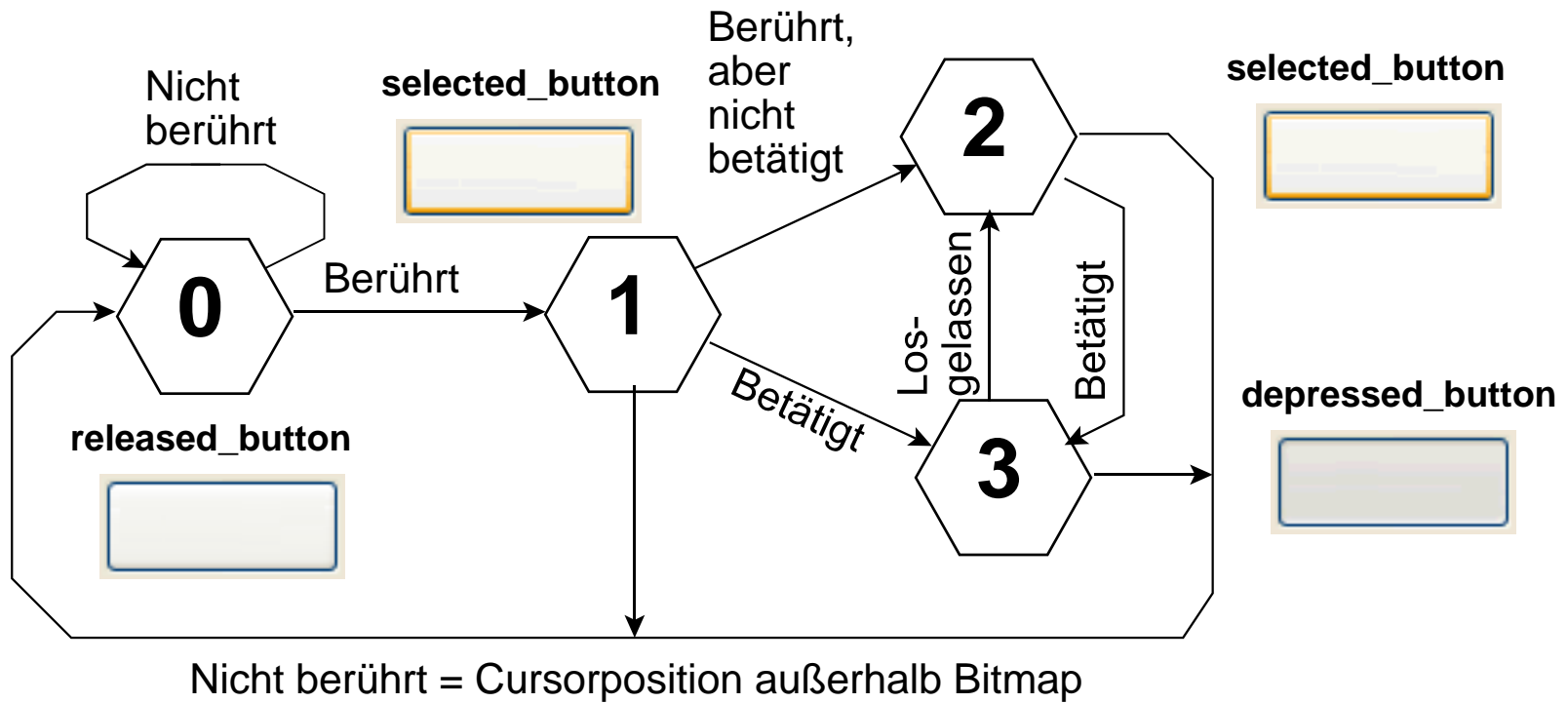
Dieser Ablauf ist bei jeder neuen Bitmapdarstellung erforderlich.

Den Text positionieren (Alternativen):

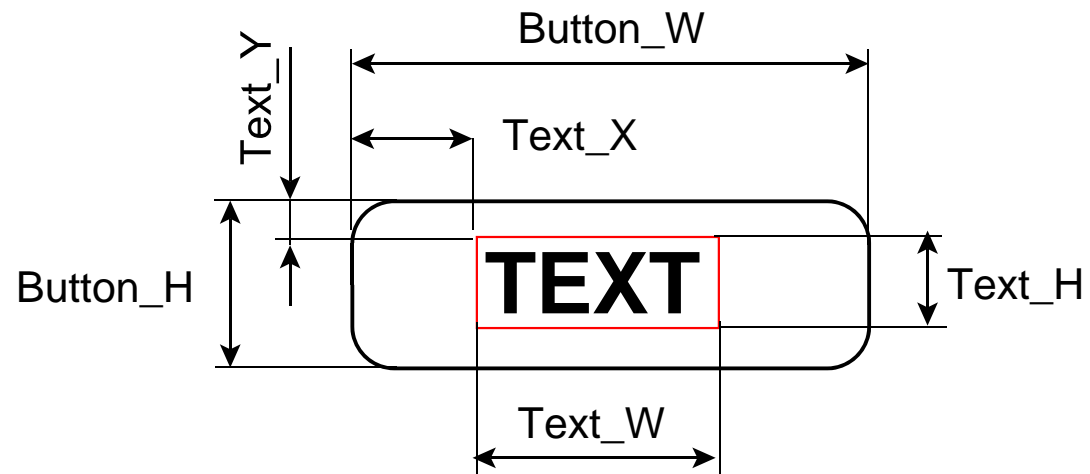
- a) Anfangskordinaten experimentell bestimmen (so, daß es nach etwas aussieht). Ggf. einen Font mit fester Zeichenbreite wählen (z. B. Courier).
- b) Die Textgröße auswerten (GRAPHIC TEXT SIZE) und die Koordinaten (1. Pixel links oben) ausrechnen.

$$\text{Text_X} = \frac{\text{Button_W} - \text{Text_W}}{2}$$

$$\text{Text_Y} = \frac{\text{Button_H} - \text{Text_H}}{2}$$



Tastenbetätigung
 Tastenzustände
 22. 5. 2015



$$\text{Text_X} = \frac{\text{Button_W} - \text{Text_W}}{2}$$

$$\text{Text_Y} = \frac{\text{Button_H} - \text{Text_H}}{2}$$

Tastenbetätigung
 Tastenbeschriftung
 26. 5. 2015