

## 17.10 LCDNumber

Die Klasse `LCDNumber` (`gb.qt4.ext`) implementiert eine Komponente, welche die anzuzeigenden Ziffern (Digits) wie einen LCD-Bildschirm zeigt. In einem 7-Segment-Anzeigeelement kann jedes Segment ein- oder ausgeschaltet werden. Jede Ziffer wird in einem separaten 7-Segment-Anzeigeelement angezeigt. Es werden die Ziffern 0 bis 9 und ein Punkt sowie die Buchstaben A, b, C, d, E und F im hexadezimalen Anzeige-Modus angezeigt.

Die Komponente `LCDNumber` gilt seit Gambas 3.4 als veraltet. Es wird die Empfehlung ausgesprochen, statt der Komponente `LCDNumber` die Komponente `LCDLabel` → Kapitel 17.16 `LCDLabel` zu verwenden. Diese Auffassung teilt der Autor nicht, denn für die Verwendung der Komponente `LCDNumber` sprechen – mit Blick auf die Möglichkeiten der Komponente `LCDLabel` – mindestens folgende Gründe:

- Die Anzeige der Zahl (Typ `Float`) kann dezimal erfolgen oder für den Teilbereich der ganzen Zahlen auch binär oder hexadezimal!
- Mit der Auswertung der Eigenschaft `LCDNumber.Overflow` (read-only) sind Sie in der Lage, den Anzeigebereich dynamisch zu vergrößern, indem Sie die Anzahl der anzuzeigenden Stellen (Digits) erhöhen.
- Ein Rand *kann* hinzugefügt werden, der sich dann in vier *unterschiedlichen* Randformen formatieren lässt.
- In der Anzeige stören keine Segmente, wenn diese nicht sichtbar sind.

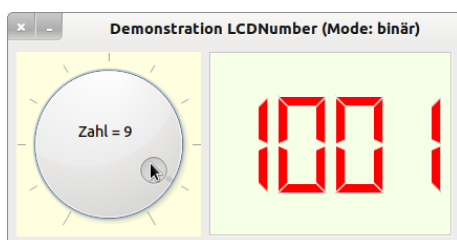


Abbildung 17.10.1: Binäre Anzeige von ganzen Zahlen mit `LCDNumber`

### 17.10.1 Eigenschaften `LCDNumber`

In der folgenden Tabelle werden ausgewählte Eigenschaften der Komponente `LCDNumber` beschrieben:

<code>LCDNumber</code>	Datentyp	Default	Beschreibung
<code>.Digits</code>	Integer	1	Über diese Eigenschaft wird die Anzahl der anzuzeigenden Ziffern festgelegt oder ausgelesen.
<code>.Mode</code>	Integer	1	Gibt den Anzeige-Modus vor (hexadezimal (0), dezimal (1), binär (3)) oder liest den Modus aus.
<code>.Overflow</code>	Boolean	-	Gibt an, ob die Zahl zu groß ist, um exakt angezeigt zu werden (read-only)
<code>.SmallDecimalPoint</code>	Boolean	False	Ermittelt oder legt fest, wie das Komma angezeigt wird. Wenn diese Eigenschaft <code>TRUE</code> ist, dann wird das Komma zwischen zwei Ziffern eingefügt.
<code>.Style</code>	Integer	0	Setzt den Style oder gibt diesen zurück. Konstanten: <code>LCDNumber.Outline</code> (0), <code>LCDNumber.Filled</code> (1) und <code>LCDNumber.Flat</code> (2)
<code>.Value</code>	Float	0.0	Legt den Wert fest, der in der Komponente <code>LCDNumber</code> angezeigt wird oder gibt diesen Wert zurück.

Tabelle 17.10.1.1: Ausgewählte Eigenschaften

Für den Anzeige-Modus werden 3 Konstanten zur Verfügung gestellt:

- 0 → `LCDNumber.Hexadecimal`,
- 1 → `LCDNumber.Decimal`,
- 3 → `LCDNumber.Binary`.

17.10.2 Beispiel LCDNumber

Im Projekt wird die Zahl (LCDNumber1.Value) aus dem Bereich [0,10] durch einen Drehknopf (Komponente Dial) erzeugt und in einer Komponente LCDNumber im dezimalen Anzeigemodus angezeigt. Gleiches gilt auch für die Projekt-Erweiterung, bei der das Anzeige-Element LCDNumber gegen ein LCDLabel ausgetauscht wurde:

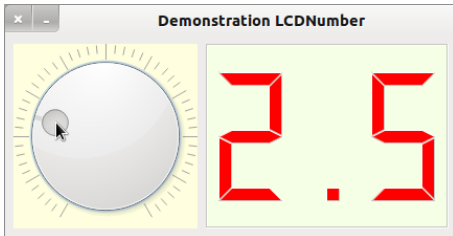


Abbildung 17.10.2.1: Dezimale Anzeige mit LCDNumber

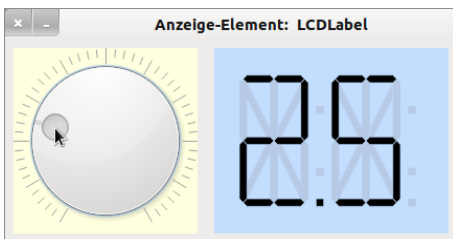


Abbildung 17.10.2.2: Dezimale Anzeige mit LCDLabel

```
' Gambas class file

Private $iRatio As Integer = 10

Public Sub Form_Open()

    FMain.Center

    Dial1.MinValue = 0
    Dial1.MaxValue = 100
    Dial1.PageStep = 10
    Dial1.Step = 1
    Dial1.Value = 25
    Dial1.Wrap = False
    LCDNumber1.Digits = 3
    LCDNumber1.SmallDecimalPoint = False
    LCDNumber1.Mode = LCDNumber1.Decimal
    Form2.Show

End ' Form_Open()

'-----

Public Sub Dial1_Change()

    GetOverflow()
    LCDNumber1.Value = Dial1.Value / $iRatio
    If LCDNumber1.Value >= 9 Then
        LCDNumber1.Background = &HFFBFBF
    Else
        LCDNumber1.Background = &HF5FFE6
    Endif ' LCDNumber1.Value >= 9 ?
End ' Dial1_Change()

Private Sub GetOverflow()
    If LCDNumber1.Overflow = True Then
        Message.Error("Die Zahl ist zu groß, um exakt angezeigt zu werden.")
        Inc LCDNumber1.Digits
        Return
    Endif ' LCDNumber1.Overflow = True ?
End ' GetOverflow()
```

Die Prozedur *GetOverflow()* greift in Abhängigkeit von der voreingestellten Anzahl der Digits und vom verwendeten Modus. Im Event-Handler *Dial1.Change()* wird die Hintergrundfarbe des Anzeige-Elements *LCDNumber* von hellgrün auf hellrot geändert, wenn der anzuzeigende Wert den Grenzwert 9 erreicht oder überschreitet:

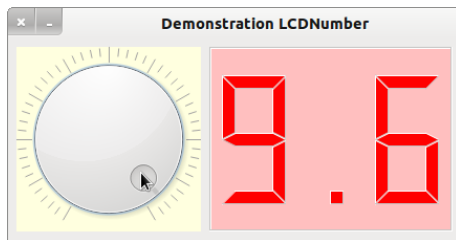


Abbildung 17.10.2.3: Anzeige im Grenzbereich mit LCDNumber