

.net Bibliothek - LCD232

LCD232 wurde von Tobias Mädél im Jahr 2011 entwickelt.



LCD232 von [Tobias Mädél](#) steht unter einer [Creative Commons Namensnennung-Nicht-kommerziell-Weitergabe unter gleichen Bedingungen 3.0 Unported Lizenz](#).
Über diese Lizenz hinausgehende Erlaubnisse können Sie unter <http://tbSPACE.de> erhalten.

In dieser Datei wird erklärt wie man die .net Framework Library nutzt.

Die LCD232-Lib kann ab .net Framework 2.0 benutzt werden, und funktioniert auch mit Mono bzw. Wine unter Linux und Mac.

Dieses Dokument überschneidet sich inhaltlich recht stark mit der Protokollerklärung.

Rev. 1 – 22.11.2011

Die Controllerklasse trägt die Bezeichnung „controller“ und befindet sich im Namespace „LCD232“.

Sie kann daher entweder am Anfang des Codes mit „Imports LCD232“ geladen werden



oder die Klasse wird zum Erben mit LCD232.controller angesprochen:

```
Dim lcd as LCD232.controller
```

Der Variablenname „lcd“ wird ab sofort in dieser Datei verwendet

Die Lib stellt eine New Funktion mit Überladungen bereit, sodass mit

```
lcd = New LCD232.controller(„COM3“)
```

eine Verbindung zum LCD auf COM3 aufgebaut werden kann.

Weiter gibt es als zweites Argument die Baudrate, welche standardmäßig auf 9600 eingestellt ist.

Das dritte Argument kann die Autoreset steuern.

Nach dem Herstellen der Verbindung sind folgende Funktionen verfügbar:

- ◆ Clear()
- ◆ display()
- ◆ noDisplay()
- ◆ Print(ByVal String)
- ◆ setBacklight(ByVal Integer)
- ◆ setCursor(ByVal Integer, ByVal Integer)
- ◆ setCustomChar(ByVal Integer, ByVal Integer, ByVal Integer, ByVal Integer, ByVal Integer, ByVal Integer, ByVal Integer)
- ◆ setRGB(ByVal Integer, ByVal Integer, ByVal Integer)
- ◆ Write(ByVal Integer)
- ◆ WriteLine(ByVal String, ByVal Integer)

Senden von Text

Zum Senden von Text, wird dieser als String durch die Funktion „WriteLine“ zusammen mit der Zeilennummer beginnend bei 0 übergeben.

```
lcd.WriteLine(„Hallo Welt“,0)
```

```
lcd.WriteLine(„Ich bin ein LCD“,1)
```

Löschen des Bildschirms

Zum Löschen des Bildschirms wird die Funktion „clear“ benutzt.

```
lcd.clear()
```

Setzen der Bildschirmzeile

Zum Setzen der Zeile wird setCursor verwendet

```
lcd.setCursor(0,1)
```

Anzeige deaktivieren

Wenn man das Kommando zum Deaktivieren der Anzeige sendet wird die Anzeige geleert, aber ohne den Inhalt zu löschen. Der im nächsten Abschnitt folgende Befehl schaltet die Anzeige wieder ein, ohne den Inhalt zu löschen.

Deaktivieren der Anzeige:

```
lcd.noDisplay
```

Anzeige aktivieren

Zum Re-Aktivieren der Anzeige wird der nachfolgende Befehl gesendet

Aktivieren der Anzeige:

```
lcd.Display
```

Setzen des Cursors

Zum Setzen des Cursors auf ein Zeichen und eine Linie, wird dieser Befehl benötigt.

```
lcd.setCursor(4,1)
```

Steuern der einfarbigen Hintergrundbeleuchtung

Die Hintergrundbeleuchtung ist an einen 8-bit regelbaren PWM-Anschluss angeschlossen, und kann daher von 0-255 geregelt werden

Komplett ausschalten:

```
lcd.setBacklight(0)
```

Komplett anschalten:

```
lcd.setBacklight(255)
```

Steuern der RGB-Hintergrundbeleuchtung

Die RGB-Beleuchtung ist an 3-PWM-Anschlüsse angeschlossen, und kann daher mit 8-bit (0-255) geregelt werden. Daraus resultieren 16 Millionen verschiedene Farbabstufungen.

`lcd.setRGB(Rotkanal, Grünkanal, Blaukanal)`

`lcd.setRGB(255,0,0)` – Schaltet das LCD auf Rot.

Definieren und Nutzen von eigenen Zeichen

Die HD44780-Controller bieten die Möglichkeit eigene Zeichen im CG-RAM zu definieren.

Dies kann man über LCD232 auch erreichen.

	1	2	3	4	5	Binär	Dezimal
1						00000	0
2		■		■		01010	10
3		■		■		01010	10
4		■		■		01010	10
5						00000	0
6	■				■	10001	17
7		■	■	■		01110	14
8						00000	0

HD44780 Zeichen sind 5x8 Pixel groß, und können so in dem Speicher geladener werden.

Die Zeichen 0-8 sind auf dem CG-RAM gelegt.

Die Dezimalenwerte des Zeichens müssen an das LCD gesendet werden.

Das letzte Argument ist die Position, in die das Zeichen geladen wird.

Smiley in Zeichen 1 laden

Bsp. `lcd.setCustomChar(00,10,10,10,00,17,14,00,01,01)`