

Der Frequenz-Zähler FreqVolt wurde durch ein Grafik-LCD erweitert und kann von 2 Hz bis ca. 20 kHz als Oszilloskop dienen.

Mit einem zusätzlichen NAND kann der R8C/13 Frequenzen bis ca. 25 MHz messen und auf dem LCD und der seriellen Schnittstelle ausgeben.

Im Schaltplan FreqVolt (*.sch Eagle und *.tif) ist ausserdem ein kleiner Testgenerator mit NE555 von 0.6 Hz bis 105 kHz enthalten.

Auszug aus Header von FreqOszi.c :

```
//
// Quelle      : FreqVolt.c      April 2006 Elektor-Forum von Wolfgang und
Didi
// Autor       : Didi Lamken      05.08.2006
// Hardware    : Original Applikations-Board von Elektor
//              Pin16             P4.5             Eingang Taster gegen GND
//              Pin15,14,13,12 P1.0,1,2,3 Ausgang LED2,3,4
//              LCD-Anzeige 2*16 Zeichen an P0.2...P0.7
//
// Frequenzzähler zusätzlicher NAND Schmitt-Trigger 74LS132 mit
//              NAND Pin1 Eingang 1: zu messendes Signal
//              NAND Pin2 Eingang 2: Torimpuls Z Ausgang TZout Pin 20 am
R8C/13
//              NAND Pin3 Ausgang : Eingang von Timer X cntr0 Pin 8 am
R8C/13
//              Pin7 GND und Pin14 +5V
//
//              Minigenerator mit NE555 0.6 Hz ... 105 kHz
//              siehe Schaltplan im Anhang
//
// Oszilloskop  Grafik-LCD 128 * 64 Pixel wie in Elektor 03/2006
//              serielle Ansteuerung mit SPI-Schnittstelle,
//              Versorgung mit eigenem 3V3-Spannungsregler
//              zu Elektor 03/2006 gibt es Abweichungen
//              Signal DB7=SID TxD0/P1.4 -> P1.2 damit COM/UART0 benutzt
werden kann
//              Signal RS      P1.7      -> P1.1 Eingang des Frequenzzählers
//              über Jumper umschaltbar
//
// Funktion    : Über ein NAND ( 1/4 74132 ) werden die Impulse an
//              Pin8 = P17/INT1#/CNTR0 gegeben und mit dem Timer X
//              im Event Counter Modus gezählt: Vorteiler prex bis 100,
//              Teiler tx bis 100, wenn TimerX nullt, gibt es einen Interrupt

//              (intx_routine mit int-vector 22) und die Zehntausender
//              werden mit cnt10m gezählt. Das Ergebnis wird dann als
//              long int berechnet und evtl. auf LCD und COM ausgegeben.
//              Der 2. NAND-Eingang wird von Pin20 = P31/TZout/CMP11 vom
//              Timer Z im "Programmable Wait One-shot Generation Mode"
//              gesteuert. Die Torzeiten 0.1ms, 1ms und 10ms werden durch
//              die Teiler prez und tzsc eingestellt, für die längeren Zeiten
//              100ms, 1s, 10s und 100s wird der Timer Y mit Teiler
//              prey und typr vorgeschaltet.
//
//              - LED1 an Port1.0 zeigt die Torzeiten an
//              - Funktionen für Frequenz und Spannung
//              Ausgabe als Integer und formatierter String
//              - Ausgabe auf a) LCD und b) COM mit 9600 8N1
//              - mit Taster an P4.5 kann die Torzeit eingestellt werden:
```

```

                                FreqOszi-Readme.txt
//      fix range = mode = 0...5 = 0.1msec...10sec
//      mode = 6 = Autorange, Wahl der Torzeit je nach gemessener
Frequenz
//      mode = 7 = Betrieb als Oszilloskop mit Torzeit 1 Sec
//      - es werden 128 Messwerte des AD-Wandlers an AN6 in einen Buffer
//        gelesen, Minimum und Maximum der Messwerte werden gesucht
//      die 128 Messwerte werden auf dem Grafik-LCD ausgegeben
//      - durch Wartezeiten beim Erfassen der Messwerte je nach Frequenz

//      kann von ca. 2 Hz bis ca. 500 Hz immer ca. 1.5 Perioden auf
//      dem Display dargestellt werden, um 800 Hz gibt es leichte
//      Nichtlinearitäten, ab 900 Hz wird unverzögert gemessen
//      ab ca. 20 kHz kann das Display nicht mehr auflösen.
//
//      - Sinus und Dreieck werden nicht gut getriggert ( NAND 74132 ?
//      ),
//      Rechteck wurde von 0.6 Hz bis 5 Mhz getestet.
//      Laut Wolfgang Paar funktioniert es bis 25 MHz.
//
//      - alle Funktionen bis auf die für den Grafik-Controller KS0713
//      befinden sich in der Hauptquelle, da der HEW Probleme
//      mit mehreren Quellen bereitete
//

```