

## **Entwurf: PicoTerm-TFT für 4,3“ Display mit 480x272 2023-12-20**

Ein Pico-Board mit RP2040 wandelt per RS232 eingehende Zeichen in Schriftzeichen um und stellt diese auf der Anzeige dar.

Ohne weitere Anpassung werden eingehende Zeichen schwarz auf weissem Hintergrund angezeigt. Dabei werden Steuerzeichen wie CR, LF, Clear, Home und Alarm beachtet. Am Bildende erfolgt ein automatischer 'scroll'.

Ein Farbwert besteht aus je zwei Bit für Rot (Bits 0,1), Grün (Bits 2,3) und Blau (Bits 4,5) und den Attributen TRANSPARENT (Bit 6) und BLINKEN (Bit 7).

Bei TRANSPARENT werden nur die aktiven Pixel in der Vordergrundfarbe beschrieben, der Hintergrund wird nicht verändert.

Es existieren zwei Bildseiten, die mit gleichem Inhalt beschrieben und zyklisch mit der Blinkfrequenz gewechselt werden. Wird die Schriftfarbe mit dem Attribut BLINKEN versehen, wird die 2. Bildseite mit 'alt\_color' beschrieben, was beim Seitenwechsel den Blinkeffekt erzeugt.

Im Textmodus werden folgende Steuerzeichen verwendet:

7	Piep, Dauer in 10 ms gemäß Vorgabe
10	LF, Zeilenvorschub
11	HOME, Textcursor auf links oben
12	CLR_HOME, Bild löschen und Textcursor auf links oben
13	CR, Cursor auf Zeilenanfang
27	ESC, für Grafikbefehle und div. Einstellungen

Alle anderen Zeichen im Bereich 0 – 0xff werden gemäß Zeichensatz ausgegeben. Dabei werden die Zoom-Faktoren für x und y beachtet, der Textcursor entsprechend erhöht und am Seitenende die Zeilen nach oben verschoben (scroll).

Mit ESC-Steuersequenzen können Farbeinstellungen verändert und Zeichen als Grafik frei positioniert angezeigt werden. Daneben gibt es Funktionen für Setzen eines Pixels, zeichnen von Linie und Rechteck sowie das Füllen einer rechteckigen Fläche.

Die Baudrate nach dem Einschalten beträgt 57,6 kBd und kann bis auf 500 kBd hochgesetzt werden. Das Datenformat ist 8N1.

Die (vorläufigen) Befehle:

ESC ESC	* als Sync-Zeichen senden
ESC @	Reset auf Einschaltzustand
ESC C zeile spalte	Textcursor auf (zeile) und (spalte) setzen: (1,1) ist oben links
ESC c 1 n	alternative Hintergrundfarbe für Blinken 0 – 0xff
ESC c 3 n	Vordergrundfarbe 0 – 0xff
ESC c 4 n	Hintergrundfarbe 0 – 0xff
ESC c 5 n	Zommfaktoren: zoom_x = n%15, zoom_y = (x>>4)%15 y = x, wenn y == 0
ESC c 6 n	Zeichensatz: 0 = CP850 8x17 teilweise umkodiert für unicode 1 = CP850 8x16, 2 = CP437 6x10 modifiziert, 3 = CP437 8x16 4 = CP866 8x16
ESC c 7 n	Blinkfrequenz = n * 10ms
ESC c 8 n	Hintergrundbeleuchtung: 0 – 8, (dunkel – hell)
ESC c b n	Baudrate per Index: 0 = 9k6 .... 10 = 500 kBd
ESC c O n	Piepdauer = n * 10ms
ESC c t n	0 = touch-aus 1 = touch-ein Werte auf Displaygröße skaliert 2 = touch-ein Rohdaten der ADCs zur nachträglichen Skalierung
ESC c p n	Sichern und Restaurieren von Anzeigeeinstellungen aufsteigend ab n = 0: PUSH_ZOOM, POP_ZOOM, CLR_ZOOM aufsteigend ab n = 3: PUSH_XY, POP_XY, CLR_XY aufsteigend ab n = 6: PUSH_FONT, POP_FONT, CLR_FONT aufsteigend ab n = 9: PUSH_COLOR, POP_COLOR, CLR_COLOR
ESC g l x1y1 x2y2 farbe	Linie von x1y1 nach x2y2 zeichnen, jeweils 2 Byte für x und y
ESC g r x1y1 x2y2 farbe	Rechteck mit Diagonale x1y1 nach x2y2 zeichnen, 2 Byte für x und y
ESC g f x1y1 x2y2 farbe	Rechteck mit Diagonale x1y1 nach x2y2 füllen, 2 Byte für x und y
ESC G c x1y1 zeichen	Zeichen an x1y1 ausgeben, jeweils 2 Byte für x und y
ESC G p x1y1 farbe	Pixel an x1y1 setzen, jeweils 2 Byte für x und y
ESC G t x1y1 farbe	Grafikcursor auf x1y1 setzen, jeweils 2 Byte für x und y
ESC V	Version und Bildgröße ausgeben