

## Die Bibliothek für das TI-Board

### **void Init\_TI\_Board(void);**

Initialisiert das TI-Board für die Nutzung. Es wird:

- der System Zeitgeber auf 1 ms eingestellt
- GPIO Port E Bits 0..7 als Eingabe konfiguriert
- GPIO Port G Bits 0..15 als Ausgabe konfiguriert
- die SPI2 Schnittstelle für Kommunikation mit dem TFT Display konfiguriert
- der USART1 initialisiert für die Kommunikation mit dem Terminal auf dem PC
  - o 115200 Baud
  - o 8 Datenbits
  - o Kein Parity
  - o kein Hardwarehandshake
- der TFT-Bildschirm initialisiert

Diese Methode muss zu Beginn eines Programms aufgerufen werden.

### **void Delay(\_\_IO uint32\_t nTime);**

- Zeitverzögerung für nTime ms.

### **uint32\_t get\_uptime(void);**

- liefert die Zeit seit der Initialisierung des Boards in Millisekunden.

## **Das EA-eDIPTFT32 Display**

### **void TFT\_Init(void)**

- Initialisiert das TFT-Display. Das TI-Board muss vorher mit Init\_TI\_Board() initialisiert worden sein. Der Terminal-Mode wird eingeschaltet und der Bildschirm wird gelöscht

### **void TFT\_cursor\_on(void)**

- schaltet den Cursor des TFT-Bildschirms ein

### **void TFT\_cursor\_off(void)**

- schaltet den Cursor des TFT-Bildschirms aus

### **void TFT\_gotoxy(int x, int y)**

- setzt den Cursor an die Position x,y.
- x → 1 ...
- y → 1 ...

### **void TFT\_buzzer\_on(int duration)**

- schaltet den Summer ein.
- Zeit in 1/10 mSec.
- 1 bedeutet unbegrenzt

### **void TFT\_buzzer\_off(void)**

- schaltet den Summer aus

### **void TFT\_cls(void)**

- löscht den Schirm und setzt den Cursor an die Position 1,1

### **void TFT\_newline(void)**

- setzt den Cursor auf die nächste Zeile

### **void TFT\_carriage\_return(void)**

- setzt den Cursor an den Zeilenanfang

### **void TFT\_putc(char c)**

- gibt ein Zeichen an der aktuellen Cursorposition aus.

### **void TFT\_puts(char \*s)**

- gibt eine Zeichenkette auf dem Bildschirm aus

### **void TFT\_set\_font(int font)**

- wählt den zu verwendenden Zeichensatz und Schriftgröße aus
  - #define FONT4x6 1 // monospaced
  - #define FONT6x8 2 // monospaced
  - #define FONT7x12 3 // monospaced
  - #define GENEVA10 4 // proportional
  - #define CHICAGO14 5 // proportional
  - #define SWISS30B 6 // proportional
  - #define BIGZIF50 7 // nur Ziffern
  - #define BIGZIF100 8 // nur Ziffern

### **void TFT\_set\_font\_color(int clr)**

- setzt die Farbe für die Zeichen 1..16
  - #define tTRANSPARENT 0
  - #define BLACK 1 // RGB: 0 0 0
  - #define 2 /\* RGB: 0 0 255 \*/
  - #define RED 3 // RGB: 255 0 0
  - #define GREEN 4 // RGB: 0 255 0
  - #define MAGENTA 5 // RGB: 255 0 255
  - #define CYAN 6 // RGB: 0 255 255
  - #define YELLOW 7 // RGB: 255 255 0

- #define WHITE 8 // RGB: 255 255 255
- #define DARKGREY 9 // RGB: 111 111 111
- #define ORANGE 10 // RGB: 255 143 0
- #define PURPLE 11 // RGB: 143 0 255
- #define DEEPPINK 12 // RGB: 255 0 143
- #define MINT 13 // RGB: 0 255 143
- #define LAWNGREEN 14 // RGB: 143 255 0
- #define SKYBLUE 15 // RGB: 0 143 255
- #define GREY 16 // RGB: 175 175 175

**void TFT\_set\_window(int font, int x\_start, int y\_start, int num\_x, int num\_y)**

- definiert ein Fenster. Die folgenden Ausgaben werden in diesem Fenster dargestellt.

## Bibliothek für die Simulation

**void SER\_Init (void);**

- Initialisiert den UART1 für die Ausgabe in das „**Serial Window**“. Diese Methode muss am Anfang des Programms aufgerufen werden und es muss das „**Serial Window**“ in der Entwicklungsumgebung geöffnet werden.

**void getline (char \*line, int n);**

- Liest eine Textzeile von der Tastatur.
  - char \* line ... Feld für die Aufnahme der Zeichenkette
  - n ..... Anzahl der Zeichen, die in das Feld eingefügt werden können.  
Die Eingabe wird beendet, wenn die Eingabetaste betätigt wird, oder die Anzahl der Zeichen erreicht ist.