

Kippschalterkleingerät 15b

Programmbeschreibung

Stand: 15. 1. 2016

- Mikrocontroller: Atmel ATmega1281 TQFP 64. Programmspeicher 128 kBytes, Datenspeicher (RAM) 8 kBytes, EEPROM 4 kBytes, Takt 8 MHz (intern), Betriebsspannung 5 V.
- LCD-Anzeige: EA DIP180B-5NLW, 180 • 32 Pixel, drei Controller (PT6520, SED1520, AX6120 o. ä.).

Programmkomponenten:

- Makrosammlung basic ab 02_039
- Makrosammlung Kleinkipp15a_01
- special_subs_kleinkipp_01
- menues_kleinkipp_01
- common_subs ab 01
- raster_chars ab 01
- graph_phy_lcd_180_32

Grundfunktionen:

1. Erzeugen der Kontrastspannung für die LCD-Anzeige
2. Ansteuern der LCD-Anzeige
3. Abfragen der Kippschalter und Stellen der Ausgänge
4. Abfrage der Eingänge und Pegelanzeige
5. Taktimpulsausgabe
6. Konfigurationssteuerung
7. Betriebsspannungserkennung und -einstellung
8. Serielle Kommunikation

Der Konfigurationsschalter:

- Raststellung: Menüsystem. Anzeige auf LCD. Bedienung mit den Kippschaltern.
- Mittelstellung: Normalfunktion.
- Taststellung: Durchschaltfunktion.

In Programmversion 1.x gibt es eine feste Normalfunktion und eine feste Durchschaltfunktion.

Die Normalfunktion:

Anzeige der Eingangspegel. Anzeige und Ausgabe der Ausgangspegel gemäß Schalterstellung.

Die Durchschaltfunktion:

Zyklisches Durchschalten der verschiedenen Arten der Darstellung.

Auswahl über Variable MACHINE_STATE:

machine_state	Darstellung der Eingangspegel
0	Mit spitzen Glyphs
1	Mit runden Glyphs
2	Im Bitindex
3	Mit den Zeichen L, -, H
4	Als Analogwerte (2stellig in Volt, z. B. 2.4)

Die Eingangspegel

Die Eingangssignale werden als Analogsignale behandelt und in digitale 8-Bit-Werte gewandelt.

Wertespeicherung:

- die gewandelten Analogwerte im ADC-Puffer,
- die Pegel- und Änderungsangaben im Änderungspuffer.

ADCBUFF	Inhalt	ADCBUFF	Inhalt
+0	Analogwert Eingang 0	+8	Änderung und Pegel Eingang 0
+1	Analogwert Eingang 1	+9	Änderung und Pegel Eingang 1
+2	Analogwert Eingang 2	+10	Änderung und Pegel Eingang 2
+3	Analogwert Eingang 3	+11	Änderung und Pegel Eingang 3
+4	Analogwert Eingang 4	+12	Änderung und Pegel Eingang 4
+5	Analogwert Eingang 5	+13	Änderung und Pegel Eingang 5
+6	Analogwert Eingang 6	+14	Änderung und Pegel Eingang 6
+7	Analogwert Eingang 7	+15	Änderung und Pegel Eingang 7

Ein Änderungs- und Pegelbyte:

7	6	5	4	3	2	1	0
CHANGE	-	-	-	-	-	LEVEL	

LEVEL: 1 = Low, 2 = High, 3 = X (verbotener Bereich).

Anzeigeprinzip:

- wenn Änderungs- und Pegelbyte = 81H, dann Low anzeigen,
- wenn Änderungs- und Pegelbyte = 82H, dann High anzeigen,
- wenn Änderungs- und Pegelbyte = 83H, dann den verbotenen Bereich anzeigen.

Die Glyphanzeige

Es gibt einen Glyph für jeden der anzuzeigenden Zustände Low, X (verbotener Bereich, unbestimmt) und High. Es sind runde und spitze Glyphs vorgesehen. Anzeigeauswahl mittels Konfigurationsschalter (Tastfunktion). Die Bitmusteradressen werden in den Variablen GLYPH_PTR_LO, GLYPH_PTR_HI und GLYPH_PTR_X übergeben.

Die Anzeigefunktionen:

Funktion	Aufruf	Parameter
Pegelwert zur Anzeige bereitstellen unter Berücksichtigung der Invertierung	Macro getinbuff	Parameter: Offset in Änderungs- und Pegelpuffer (bezogen auf ADCBUFF) Rückgabe: AL: Änderungs- und Pegelbyte AH: Invertierung ja/nein
Den Glyph auswählen	selcon_glyph	AL: Änderungs- und Pegelbyte AH: Invertierung ja/nein Rückgabe: Z: Bitmusteradresse des darzustellenden Glyphs ZF = 0: Änderung. Glyph anzeigen. ZF = 1: keine Änderung. Nicht anzeigen.
Eingangssignale wandeln und alle Glyphs anzeigen	show_inputs_glyph	
Spitze Glyphs zur Anzeige einstellen	spitz	Rückgabe: Bitmusteradressen in GLYPH_PTR_LO, GLYPH_PTR_HI, GLYPH_PTR_X
Runde Glyphs zur Anzeige einstellen	rund	Rückgabe: Bitmusteradressen in GLYPH_PTR_LO, GLYPH_PTR_HI, GLYPH_PTR_X

Die Bitindexanzeige

Je nach Pegel wird die Indexanzeige durch eine Zeichenkette (Item) aus jeweils 3 Zeichen dargestellt. Für jeden Bitindex (7...0) gibt es drei Items, um die Pegel Low, High und X anzuzeigen.

Die Anzeigefunktionen:

Funktion	Aufruf	Parameter
Pegelwert zur Anzeige bereitstellen unter Berücksichtigung der Invertierung	Macro getinbuff	Parameter: Offset in Änderungs- und Pegelpuffer (bezogen auf ADCBUFF) Rückgabe: AL: Änderungs- und Pegelbyte AH: Invertierung ja/nein
Das Item auswählen	selcon_index	AL: Änderungs- und Pegelbyte AH: Invertierung ja/nein Rückgabe: ZF = 0: Änderung. Index anzeigen. ZF = 1: keine Änderung. Nicht anzeigen. Variable CHARTYPE 0 = Normaldarstellung 1 = Inversdarstellung 2 = X-Pegel darstellen
Eingangssignale wandeln und alle Items anzeigen	show_inputs_index	

Die Zeichenanzeige

Je nach Pegel werden die Zeichen L, H oder X dargestellt.

Die Anzeigefunktionen:

Funktion	Aufruf	Parameter
Pegelwert zur Anzeige bereitstellen unter Berücksichtigung der Invertierung	Macro getinbuff	Parameter: Offset in Änderungs- und Pegelpuffer (bezogen auf ADCBUFF) Rückgabe: AL: Änderungs- und Pegelbyte AH: Invertierung ja/nein
Das Zeichen auswählen	selcon_char	AL: Änderungs- und Pegelbyte AH: Invertierung ja/nein Rückgabe: ZF = 0: Änderung. Index anzeigen. ZF = 1: keine Änderung. Nicht anzeigen. AL: Zeichencode
Eingangssignale wandeln und Zeichen anzeigen	show_inputs_char	

Die Analogwertanzeige

Die gewandelten Analogwerte werden als Spannungen in Volt zweistellig dezimal angezeigt. Auflösung: 100 mV. Anzeigeform: a.b. Wandlung über vorberechnete Tabelle (DECIMAL_TABLE). Die Werte werden ständig gewandelt und angezeigt, gleichgültig ob sie sich geändert haben oder nicht.

Die Anzeigefunktionen:

Funktion	Aufruf	Parameter
Einen binären Analogsignalwert in zwei Dezimalstellen wandeln	conv_to_decimal	AL: Binärwert Rückgabe: AL: das höherwertige Dezimalzeichen AH: das niederwertige Dezimalzeichen
Einen Analogwert mit zwei Dezimalstellen und Dezimalpunkt anzeigen	show_analog_exec	CH: X-Koordinate CL: Y-Koordinate Bezugspunkt (x, y): links unten. AL: Analogwert (als Binärzahl)
Eingangssignale wandeln und dezimal anzeigen	show_inputs_analog	

Die Logikpegel der Eingänge

Die Logikpegel werden jeweils durch einen Low-Schwellenwert und einen High-Schwellenwert gekennzeichnet. Es werden 5 verschiedene Logikspezifikationen unterstützt. Der Pegelbewertung liegen die jeweiligen Eingangspegel zugrunde (V_{ILOmax} , V_{IHImin}). Pegelauswahl über Menübedienung, Spannungseinstellung mittels Kalibrierfunktion.

Der AD-Wandler bezieht sich auf eine Referenzspannung $V_{REF} = 5 \text{ V}$.

Spannungswert bezogen auf Referenzspannung = $(255 \cdot \text{Spannung in V}) : 5$
 = Spannung in V $\cdot 51$

Logikpegel gemäß Spezifikation:

Pegel	Variable SLECTED_LEVEL	Schwellenwert Low (Default)		Schwellenwert High (Default)	
TTL	0	0,8 V	41	2 V	102
CMOS 5 V	1	1,5 V / 1,2 V*	77/61	3,5 V	178
CMOS 3,3 V	2	0,8 V	41	2,4 V	122
OWN1. Default: CMOS 2,5 V	3	0,7 V	35	1,7 V	86
OWN2. Default: CMOS 1,8 V	4	0,6 V	30	1,2 V	61

*: Default-Wert 1,2 V anstatt 1,5 V, um den Low-Pegel sicher vom offenen Eingang unterscheiden zu können. Für diese Kompromißlösung sprechen zwei Erwägungen: (1) Die TTL-Spezifikation ist die anwendungspraktisch wichtigere. (2) In der Praxis liegt der Low-Pegel von CMOS-Ausgängen typischerweise nahe bei 0 V (Schaltverhalten Rail-to-Rail).

Pegel bei offenem Eingang = $VIO \cdot \text{Teilverhältnis (Nennwert: 0,3125)}$:

VIO	Pegel bei offenem Eingang (Nennwert)
3,3 V	1,03 V
5 V	1,56 V

Default-Werte:

Spannung in V	0,8	1,2	2,0	2,4	3,5	0,7	1,7	0,6	1,2
Umrechnung	40,8	61,2	102	122,4	178,5	35,7	86,7	30,6	61,2
gerundet	41	61	102	122	178	35	86	30	61

Die Kippschalter abfragen

Die Kippschalter werden in der Hauptsteuerschleife und in den Menüprogrammen abgefragt.

Die Kippschalter in der Hauptsteuerschleife

In der Hauptsteuerschleife werden die Schalterstellungen mit der Ausgangsinvertierung (Variable INVERTED_OUT) antivalent verknüpft und auf den Ausgabeport gegeben. Ausgabe nach dem Prinzip Open Drain über das Richtungssteuerregister des Ports. High-Pegel: Richtung auf Eingabe. Damit Portanschluß hochohmig. Pegel über Pull-up-Widerstand. Low-Pegel: Richtung auf Ausgabe. Datenregister auf 0, damit Ausgabe eines Low-Pegels. Die Ausgabepegel werden angezeigt, wobei die Ausgabeinvertierung berücksichtigt wird. Anzeige ("L" oder "H") entspricht dem jeweiligen Ausgabepegel. Die Anzeige wird nur für jene Schalterstellungen neu aufgebaut, die sich geändert haben.

Die Kippschalterfunktionen in der Hauptsteuerschleife:

Funktion	Aufruf	Parameter
Die Schalter abfragen und entprellen. Rückkehr, wenn keine Änderungen oder stabile Änderungen (kein Prellen)	get_switches	Rückgabe: AL: die aktuellen Schalterstellungen AH: die Änderungen Variable: SWITCHBUFF: die aktuellen Schalterstellungen SWITCHDELTA: die Änderungen
Die Schalterstellungen aus der Hardware abholen	fetch_switches	Rückgabe: AL: die aktuellen Schalterstellungen
Die Schalterstellungen anzeigen (Darstellung mit Zeichen)	show_switches	Variable: SWITCHBUFF: die aktuellen Schalterstellungen SWITCHDELTA: die Änderungen INVERTED_OUT: die Ausgabeinvertierung SWITCHES_Y: die y-Koordinate der Darstellung (unterste Rasterzeile des Zeichens)

Die Taktimpulsausgabe

In dieser Sonderbetriebsart werden über die Ausgänge 3 bis 0 Taktimpulse ausgegeben. Zur Impulserzeugung wird die Zeitgeber-Zähler-Einheit 3 ausgenutzt. Sie wird als Zähler mit einem Takt von 1 MHz betrieben. Beim Zählerumlauf wird eine Unterbrechung ausgelöst (Timer/Counter3 Overflow Interrupt). Die Taktimpulse werden programmseitig gebildet. Die Taktperiode ergibt sich durch entsprechendes Laden des Zählregisters. Um symmetrische Impulse zu erzeugen, wird der Zähler für die Dauer einer halben Taktperiode (in μs) geladen (65355 – halbe Periodendauer). Die Taktperioden an den Ausgängen 3, 2, 1, 0 verhalten sich im Verhältnis 1000 : 100 : 10 : 1 (programmseitige dezimale Frequenzteilung). Es kann eine von drei verschiedenen "Geschwindigkeiten" (Taktfrequenzen) ausgewählt werden (Kippschalterbedienung). Die Taktimpulsausgabe kann sowohl im Menübetrieb als auch im Normalbetrieb (Hauptsteuerschleife) aktiv sein.

Anhalten und Ausschalten der Taktimpulsausgabe

Die Taktimpulsausgabe ist eine einfache Zusatzfunktion mit beschränkten Bedienmöglichkeiten. Über das Taktimpulsmenü kann die Taktimpulsausgabe ein- und ausgeschaltet werden. Ist sie eingeschaltet, so bleibt sie bei beliebigen Menüwechseln und auch bei Rückkehr in den Normalbetrieb aktiv. Im Normalbetrieb wirken dann nur die Kippschalter 7 bis 4 direkt auf die entsprechenden Ausgänge. Die Kippschalter 3, 2, 1 dienen zur Steuerung der Taktimpulsausgabe. Kippschalter 0 ist wirkungslos. Im Normalbetrieb kann die Taktimpulsausgabe angehalten und erneut gestartet werden (Kippschalter 1). Zum Ausschalten (so daß alle Kippschalter direkt auf alle Ausgänge wirken) ist eine Rückkehr ins Taktimpulsmenü erforderlich (Konfigurationsschalter in Raststellung). Beim Anhalten und Ausschalten wird die Impulsausgabe lediglich gestoppt, die augenblicklichen Signalpegel (Low oder High) bleiben gewissermaßen hängen. Ein echter zyklusgenauer Taktstopp mit definiertem Ruhepegel ist nicht möglich.

Taktgeschwindigkeiten:

Geschwindigkeit	Ausgang 3	Ausgang 2	Ausgang 1	Ausgang 0
SLOW	1 s (1 Hz)	100 ms (10 Hz)	10 ms (100 Hz)	1 ms (1 kHz)
MEDIUM	0,5 s (2 Hz)	50 ms (20 Hz)	5 ms (200 Hz)	0,5 ms (2 kHz)
FAST	0,2 s (5 Hz)	20 ms (50 Hz)	2 ms (500 Hz)	0,2 ms (5 kHz)

Die Variable CLOCKBITS:

7	6	5	4	3	2	1	0
CLK ON	CLK HLT	0	0	OUTPUT3	OUTPUT2	OUTPUT1	OUTPUT0

CLK ON = Taktimpulsausgabe aktiv. CLK HLT = Taktimpulsausgabe angehalten. OUTPUT3...0 sind die Quellen der auszugebenden Bits. Die Ausgabe unterliegt der Ausgabeinvertierung (Bits 7...4 von den Kippschaltern 7...4, Bits 3...0 aus der Variablen).

Die Variable CLOCKSPEED:

7	6	5	4	3	2	1	0
0	0	0	0	0	0	FAST	MEDIUM

Geschwindigkeitscodierung im Code 1-aus-n.

Das Menüsystem

Das Menüsystem besteht aus Menübildern (Frames). Zu jedem Frame gehören:

1. die Deskriptoren der Menüanzeigen (Items),
2. ein Menüprogramm, das jeweils eine eigene Steuerschleife enthält.

Eintritt in Menüsystem:

Durch Schalten des Konfigurationsschalters in seine Raststellung.

Verlassen des Menüsystems:

Durch Zurückschalten des Konfigurationsschalters (Mittelstellung).

Menüfunktionen:

Funktion	Aufruf	Parameter
Ein Menüfeld (Item) anzeigen (Normaldarstellung).	Macro showit	Die Adresse des Item-Deskriptors
Ein Menüfeld (Item) anzeigen (Normaldarstellung), wenn ZF gesetzt. Nichts tun, wenn ZF gelöscht.	Macro showitz	Die Adresse des Item-Deskriptors ZF = 0: nichts tun ZF = 1: anzeigen
Ein Menüfeld (Item) anzeigen. Normaldarstellung, wenn ZF gelöscht. Inversdarstellung, wenn ZF gesetzt.	Macro showitcond	Die Adresse des Item-Deskriptors ZF = 0: Normalanzeige ZF = 1: Inversanzeige
Die Schalterstellungen aus der Hardware abholen und speichern	Macro showitcondz	

Die Menübilder (Frames):

Bezeichnung	Funktion
FRAME0	Grundbild. Auswahl weiterer Menüs
FRAME2	VIO-Einstellung (aktuelle Spannungswahl)
FRAME3	Pegelauswahl
FRAME4	Einstellungen retten
FRAME5	Auf Grundeinstellungen (Default-Werte) zurückgehen
FRAME6	Auswahl von Konfigurationsmenüs (SETUP)
FRAME7	Kontrasteinstellung LCD
FRAME8	Kalibrieren (Schwellenspannungen einstellen)
FRAME9	Die Anzeigeform auswählen (Glyphs, Zeichen usw.)
FRAME10	Starteinstellungen konfigurieren
FRAME11	VIO-Starteinstellungen
FRAME12	Funktionsauswahl
FRAME13	Serielle Slave-Schnittstelle
FRAME14	Serielle USB-Schnittstelle
FRAMEINVIN	Eingangsinvertierung
FRAMEINVOUT	Ausgangsinvertierung
FRAMECLOCK1	Taktimpulsausgabe

Die Kippschalter in den Menüprogrammen

Die Kippschalter dienen zur Funktionsauslösung. Funktionen werden nur beim Einschalten ausgelöst. Ist der Kippschalter beim Eintritt in das Menü bereits eingeschaltet, so wird zunächst keine Funktion ausgelöst. Erwartet wird dann eine Betätigung AUS – EIN. Ein Kippschalter, der sich in einer EIN-Stellung befindet (Raststellung), verhindert nicht die Funktionsauslösung mit anderen Kippschaltern. Es ist nicht erforderlich, in die Stellung EIN eingerastete Kippschalter zurückzuschalten, um mit anderen Kippschaltern Funktionen auszulösen.

Kippschalterzustände

Weil die Kippschalter einzelne Funktionen auslösen, müssen sie auch einzeln verwaltet werden. Hierzu wird für jeden Kippschalter ein Zustandsautomat implementiert. Alle Zustandsautomatenfunktionen werden nacheinander aufgerufen. Jedem Kippschalter ist ein Zustandsbyte zugeordnet (SW7STAT bis SW0STAT). Es gibt drei Zustände (0...2).

Kippschalterereignisse

Das Betätigen und das Loslassen eines Kippschalters löst jeweils ein Ereignis aus. Jedes Ereignis wird in einer Bitposition angezeigt. Betätigt = ACTUATED, Losgelassen = RELEASED. Die ausgelösten Ereignisse sind in den Variablen SWITCHES_ACTUATED und SWITCHES_RELEASED abfragbar. Schalter 0 entspricht Bitposition 0 usw. Ein gesetztes Bit bedeutet, daß das Ereignis aufgetreten ist. Anwendungsseitige Auswertung durch Bitabfrage. Vor jeder neuen Schalterabfrage werden die Ereignisse gelöscht.

Die Kippschalter der Taktimpulsausgabe

In dieser Betriebsart wirken die zur Steuerung vorgesehenen Kippschalter (3, 2, 1) statisch. Die Stellungen EIN und AUS haben jeweils bestimmte Wirkungen. Beim Wechsel zwischen Taktimpulsmenü und Normalbetrieb mit Taktimpulsausgabe müssen die Wirkungen der jeweiligen Schalterstellungen erhalten bleiben; es wäre unschön, würden beim Bildwechsel immer wieder Betätigungsfolgen AUS – EIN erforderlich sein, um bestimmte Wirkungen hervorzurufen.

Der Sonderzustand FRAME TRANSFER

Ist dieser Zustand (Bitposition 2 in Variable DIAG_STATE) aktiv, werden beim Menüwechsel (neues Menü oder Rückkehr zum Normalbetrieb) die Zustände und Ereignisse der Kippschalter nicht gelöscht, sondern beibehalten. Das nachfolgende Menü übernimmt die Schalterstellungen so, wie sie im vorhergehenden Menü eingestellt waren. Nach dem Menüwechsel wird der Sonderzustand wieder ausgeschaltet.

Hinweis: Der Sonderzustand wird hier nur genutzt, um das Hin- und Herschalten zwischen Taktimpulsmenü und Normalbetrieb zu unterstützen. Bei anderen Menüwechselfolgen kann es vorkommen, daß auch Schalter der Taktimpulsausgabe eine Betätigungsfolge AUS – EIN benötigen, um wieder wirksam zu werden.

Die Variable DIAG_STATE:

7	6	5	4	3	2	1	0
0	0	0	0	0	FRAME TRANSFER	TOGGLE OUT ENABLE	VIO ANNOUNCE MODE

- VIO ANNOUNCE MODE. Darstellung der VIO-Warnungsanzeigen. 0 = kurzzeitig, 1 = halten bis Konfigurationsschalterbetätigung (Tastfunktion)
- TOGGLE OUT ENABLE. Prüfpulsausgabe über Slave TX (Laufzeitmessung). 0 = keine Prüfpulsausgabe. 1 = Signal Slave TX wird bei jedem Umlauf der Hauptsteuerschleife umgeschaltet (Toggle-Funktion zwecks Zeitmessung mittels Oszilloskop).
- FRAME TRANSFER. Zustände und Ereignisse der Kippschalterbetätigung beim Menüwechsel beibehalten. 0 = bei jedem Menüwechsel (neues Menü oder Rückkehr zum Normalbetrieb) werden die Zustände und Ereignisse gelöscht. Jeweils neue Betätigung erforderlich. 1 = bei jedem Menüwechsel werden die Zustände und Ereignisse beibehalten; die Schalterstellungen werden ins neue Menü übernommen.

Kippschalterzustände in den Menüprogrammen:

Zustand	Erreicht	Verlassen	Anmerkungen
0	<ul style="list-style-type: none"> • Beim Eintritt ins Menüprogramm • Wenn Zustand 2 und losgelassen 	Wenn losgelassen	Warten auf Stellung AUS
1	Wenn Zustand 0 und losgelassen	Wenn betätigt	Warten auf Stellung EIN. Beim Verlassen wird Ereignisbit ACTUATED gesetzt.
2	Wenn Zustand 1 und betätigt	Wenn losgelassen	Warten auf Stellung AUS. Beim Verlassen wird Ereignisbit RELEASED gesetzt.

Kippschalterfunktionen in den Menüprogrammen:

Funktion	Aufruf	Parameter
Einen Schalter abfragen, entprellen und anzeigen. Die Schalterzustände verwalten. Anzeige der Schalterstellungen als Menüfelder (Items).	Macro sw_exec	Parameter: Schalternummer (7...0), Zustandsbyte, Kippschalteranzeigeelement (Item) Rückgabe: Zustandsbyte Ereignisbits
Alle Schalter abfragen, entprellen und anzeigen. Die Schalterzustände verwalten.	scan_switches	Wie Macro sw_exec
Die Schalter der Taktimpulsausgabe abfragen, entprellen und anzeigen. Die Schalterzustände verwalten.	scan_clk_switches	Wie Macro sw_exec
Alle Schalterzustände löschen	clear_switches	Rückgabe: Alle Zustandsbytes gelöscht; alle Kippschalter in Zustand 0.
Die Schalterstellungen aus der Hardware abholen und speichern	all_switches	Rückgabe: AL: die aktuellen Schalterstellungen Variable SWITCHES: die aktuellen Schalterstellungen

Die VIO-Überwachung und Steuerung

Der VIO-Pegel wird in allen Steuerschleifen zyklisch überwacht. Hierzu ist jeweils der Macro vioscan entsprechend eingefügt. Wird kein Problem erkannt, so verhält sich die Analysefunktion wie ein normales Unterprogramm (mit Rückkehr). Wird ein Problem erkannt, so erfolgt eine Anzeige und ggf. Behandlung. Die Analysefunktion bewirkt dann anstelle einer Programmfortsetzung eine harte Verzeigung zu einem Wiederanlaufpunkt, dessen Startadresse im vioscan-Macroaufruf übergeben wurde.

Ist VIO vorhanden? – Das Prinzip der VIO-Überwachung

Hierzu wird der Komparator des Mikrocontrollers ausgenutzt. Die Prüfspannung VIODET wird mit der internen Referenzspannung (Bandgap Reference Voltage) und mit der externen Referenzspannung VIOREF verglichen. VIODET hängt von VIO ab.

- VIODET = 3,6 V, wenn VIO = 5 V.
- VIODET = 1,9 V, wenn VIO = 3,3 V.
- Bandgap Reference Voltage $V_{\text{Bandgap}} = 1,1 \text{ V}$.
- VIOREF = 2,5 V.
- Bandgap Reference Startup Time ist 70 μs .
- AIN0 = Referenzspannung VIOREF = Komparatoreingang AIN0 (+).
- AIN1 = Prüfspannung VIODET = Komparatoreingang AIN1 (-).

Vergleichsablauf:

1. Ist $V_{IODET} < V_{Bandgap}$? – Wenn ja, dann kein VIO vorhanden.
2. Ist $V_{IODET} < V_{IOREF}$? – Wenn ja, sind es 3,3 V. Wenn nein, sind es 5 V.

VIO liegt von außen an. Keine externe Betriebsspannung.

VIO-Zustand = vio_ext5 oder vio_ext3. Änderungen des VIO-Pegels werden erkannt und angezeigt. VIO-Ausfall = Gerät schaltet aus.

VIO liegt von außen an. Externe Betriebsspannung liegt an.

VIO-Zustand = vi_ext5 oder vio_ext3. Änderungen des VIO-Pegels werden erkannt und angezeigt. VIO-Ausfall = Übergang in Zustand vio_no und Reaktion wie eingestellt.

Kein VIO von außen. Externe Betriebsspannung liegt an.

VIO-Zustand = vio_no (entweder von Anfang an oder durch Ausfall der externen VIO-Zuführung).
Reaktionsmöglichkeiten:

- nichts tun (kein VIO),
- den letzten externen VIO-Pegel aufschalten (5 V oder 3,3 V),
- 3,3 V aufschalten,
- 5 V aufschalten.

VIO-Zustände:

VIO-Zustand (VIO_STATE)	Bedeutung
vio_idle	Gar nichts; Anfangszustand
vio_ext5	5 V von außen liegen an
vio_ext3	3.3 V von außen liegen an
vio_no	Es liegt kein VIO an
vio_int5	Es sind 5 V von innen aufgeschaltet
vio_int3	Es sind 3,3 V von innen aufgeschaltet

VIO-Betriebszustände:

VIO-Pegel von außen	Externe Betriebsspannung (5 V)	Gerätefunktion
Nein	Nein	Gerät nicht in Betrieb. Trivialfall
Ja	Nein	Speisung mit VIO
Nein	Ja	Gerät in Betrieb. VIO gemäß Einstellung von innen aufgeschaltet (keine, 5 V, 3,3 V).
Ja	Ja	Externe Speisung (5 V). VIO von außen.

Funktionen der VIO-Überwachung:

Funktion	Aufruf	Parameter
Eine VIO-Analyse in einer Steuerschleife ausführen	Macro vioscan	Parameter: Wiederanlaufadresse im Problemfall
Den VIO-Pegel analysieren	vioscan_exec	
Den VIO-Pegel mittels Komparator analysieren	vioscan_compare	Rückgabe: AL: der aktuelle VIO-Pegel: vio_no: kein VIO vorhanden vio_ext5: 5 V vio_ext3: 3,3 V
Eine VIO-Nachricht anzeigen	Macro viomessage	Parameter: Zeiger auf das VIO-Anzeigeprogramm
Eine VIO-Nachricht anzeigen	show_vioframe	Variable VIOFRAMEX: Start des VIO-Anzeigeprogramms Variable VIORETURN: Rückkehr aus der VIO-Behandlung (Wiederanlauf)

Variable zum Steuern der VIO-Ausfallreaktion:

Variable	Wirkung
NOVIO_ACTION	0: Bedienaufforderung 1: automatisch reagieren
NOVIO_LEVEL	0: nichts aufschalten 1: 5 V aufschalten 2: 3,3 V aufschalten 3: den letzten externen Pegel aufschalten gemäß Variable VIO_OLDEX
VIO_ANNOUNCE_MODE	0: kurzzeitige Anzeige 1: Anzeige halten, bis Konfigurationsschalter betätigt (Taststellung)

Ein VIO-Pegel wurde aufgeschaltet, es wird aber ein abweichender Pegel oder eine Überstrombedingung erkannt (VIO-Konfliktfälle).

Beides weist auf einen Fehlerfall hin (Aufschalten eines anderen Pegels von außen oder Kurzschluß). Die VIO-Aufschaltung wird beendet. Übergang in den Zustand vio_no. Es wird dann die Zuführung eines korrekten VIO-Pegels oder – nach Beseitigen der Fehlerbedingung – das manuelle Aufschalten eines internen VIO-Pegels erwartet (Menübedienung).

Die LED-Anzeige:

LED-Anzeige	Wirkung
Rot	Kein VIO
Rot blinkend, dann rot	Ein VIO-Ausfall oder Konflikt wurde erkannt
Grün	VIO liegt an (von außen oder innen)
Grün blinkend, dann grün	Ein interner VIO-Pegel ist aufgeschaltet worden

Anzeigen der VIO-Überwachung:

VIO-Betriebszustand	Anzeigefunktion	Anzeige / Wirkung
Von außen werden 5 V zugeführt	VIOANNOUNCE5EX	EXTERNAL VIO 5 V
Von außen werden 3,3 V zugeführt	VIOANNOUNCE3EX	EXTERNAL VIO 3,3 V
VIO von außen ausgefallen	VIOANNOUNCEFAIL	EXT. VIO FAILED - NO VIO SELECT INT.VIO VIA MENUE
Es liegt keine VIO an	VIOANNOUNCENO	**** NO VIO ****
5 V werden intern aufgeschaltet	VIOANNOUNCE5IN	INTERNAL VIO 5 V
3,3 V werden intern aufgeschaltet	VIOANNOUNCE3IN	INTERNAL VIO 3,3 V
Überstromabschaltung (Überlastung)	VIOOVERCURRENT	INTERNAL VIO OVERCURRENT **** NO VIO **** VIO wird abgeschaltet
VIO-Konflikt (innen / außen)	VIOCONTENTION	INTERNAL VIO CONTENTION **** NO VIO **** VIO wird abgeschaltet

Variable im RAM:

Bezeichnung	Bytes	Inhalt
PIXBUFF	720	Pixelpuffer. Länge = PIXBUFFSIZE
SERBUFFCOUNT_USB	2	Füllstandszähler serieller Puffer USB
SERBUFFCOUNT_SLAVE	2	Füllstandszähler serieller Puffer Slave
SERBUFF_USB	250	Serieller Puffer USB. Länge = SERBUFFSIZE
SERBUFF_SLAVE	250	Serieller Puffer Slave. Länge = SERBUFFSIZE
LCTL_CTL_SEL	1	Auswahl des LCD-Controllerschaltkreises. Puffer für E-Signal
XBASE	1	Anfängliche X-Koordinate der Rasterzeichendarstellung
YBASE	1	Anfängliche Y-Koordinate der Rasterzeichendarstellung
CHARTYPE	1	Anzeigezustand. 0 = normal, 1 = invers, 2 = gar nicht (dunkel)
GLYPH_X	1	Anfängliche X-Koordinate der Glyphdarstellung
GLYPH_Y	1	Anfängliche Y-Koordinate der Glyphdarstellung
GLYPH_LEN	1	Länge (Breite) des Glyphs
GLYPH_HEIGHT	1	Höhe des Glyphs
ADCBUFF	16	AD-Wandlung; 8 Wandlungsbytes, 8 Änderungs- und Pegelbytes
GLYPH_PTR_LO	2	Zeiger auf Glyphbitmuster für LO-Anzeige
GLYPH_PTR_HI	2	Zeiger auf Glyphbitmuster für HI-Anzeige
GLYPH_PTR_X	2	Zeiger auf Glyphbitmuster für X-Anzeige
MENU_ITEM_X	1	X-Koordinate des darzustellenden Menü-Items
MENU_ITEM_Y	1	X-Koordinate des darzustellenden Menü-Items
SWITCHES_Y	1	Y-Koordinate der Schalteranzeige
SWITCHBUFF	1	Schalterstellungen im Normalbetrieb (Stimuluseingabe)
SWITCHDELTA	1	Änderungen der Schalterstellungen im Normalbetrieb
SW0STAT	1	Zustandbyte Schalter 0 (Menübetrieb)

Bezeichnung	Bytes	Inhalt
SW1STAT	1	Zustandbyte Schalter 1 (Menübetrieb)
SW2STAT	1	Zustandbyte Schalter 2 (Menübetrieb)
SW3STAT	1	Zustandbyte Schalter 3 (Menübetrieb)
SW4STAT	1	Zustandbyte Schalter 4 (Menübetrieb)
SW5STAT	1	Zustandbyte Schalter 5 (Menübetrieb)
SW6STAT	1	Zustandbyte Schalter 6 (Menübetrieb)
SW7STAT	1	Zustandbyte Schalter 7 (Menübetrieb)
SWITCHES	1	Schalterstellungen im Menübetrieb
SWITCHES_ACTUATED	1	Ereignisbits Schalter betätigt (Menübetrieb)
SWITCHES_RELEASED	1	Ereignisbits Schalter losgelassen (Menübetrieb)
BITBUFF	1	Arbeitspuffer für Bitanzeige
CLOCKSPEED	1	Taktgeschwindigkeit
CLOCKBITS	1	Bitpuffer für Taktimpulsausgabe
CLOCKDIV1	1	Taktteiler (Dezimalzähler) 1. Stufe (1:10)
CLOCKDIV2	1	Taktteiler (Dezimalzähler) 2. Stufe (1:100)
CLOCKDIV3	1	Taktteiler (Dezimalzähler) 3. Stufe (1:1000)
VIO_STATE	1	VIO-Auswahlzustand (Art und Wert des VIO-Pegels)
VIO_OLDEX	1	Bisheriger externer VIO-Pegel (Rettung)
VIOFRAMEX	2	Zeiger auf das auszuführende VIO-Anzeigeprogramm
VIORETURN	2	Zeiger auf Rückkehrpunkt aus VIO-Anzeige (Wiederanlauf)
CURRENT_LEVEL	1	Aktueller Analogwert (Rettung)
PARAM_AREA	–	Anfangsmarke des Konfigurationsparameterbereichs
LO_THRESHOLD	1	Schwellenwert LO-Pegel
HI_THRESHOLD	1	Schwellenwert HI-Pegel
SELECTED_LEVEL	1	Pegelauswahl (TTL, CMOS5, CMOS33, OWN1, OWN2)
INVERTED_IN	1	Invertierung der Eingabe
INVERTED_OUT	1	Invertierung der Ausgabe
NOVIO_ACTION	1	Reaktion, wenn keine externe VIO. 0 = nichts tun / manuelle Einstellung, 1 = automatisch aufschalten
NIOVIO_LEVEL	1	Der automatisch aufzuschaltende VIO-Pegel. 0 = keiner, 1 = 5 V, 2 = 3,3 V, 3 = wie bisheriger externer Pegel (VIO_OLDEX)
DIAG_STATE	1	Zustandsbits zu Sonderzwecken
MACHINE_STATE	1	Maschinenzustand = Art der Anzeige im Normalbetrieb
PWM_PERIOD	2	PWM-Periodendauer der Kontrastspannungserzeugung
PWM_PULSE	2	PWM-Impulsauer der Kontrastspannungserzeugung
PARAM_END	–	Endemarke des Konfigurationsparameterbereichs

Parameterrettung

Wichtige Betriebsparameter werden im EEPROM gespeichert. Nach dem Einschalten werden sie aus dem EEPROM geholt. Einstellungen, die während des Betriebs geändert wurden, müssen von Hand gerettet werden (Menübedienung, Funktionen SAVE).

Grundeinstellungen (Default-Werte)

Default-Werte sind Festwerte der Betriebsparameter. Sie können von Hand in den EEPROM zurückgeschrieben werden (Menübedienung, Funktionen DEFAULT).

Parametereinstellungen nach dem Einschalten

Der EEPROM wird abgefragt, ob er gültige Werte enthält oder nicht. Das wird anhand der Signaturbytes erkannt (gültige Werte, wenn Signaturbytes = AA55H). Enthält er keine gültigen Werte, werden die Default-Werte und gültige Signaturbytes eingetragen.

Variable im EEPROM:

Bezeichnung	Bytes	Inhalt
SIGNATURE	2	Kennung, daß Inhalt gültig = Festwert AA55H
SVAE_AREA	–	Anfangsmarke des Parameter-Rettungsbereichs*
SAVED_LO_THRESHOLD	1	Schwellenwert LO-Pegel (aktuelle Auswahl)
SAVED_HI_THRESHOLD	1	Schwellenwert HI-Pegel (aktuelle Auswahl)
SAVED_SELECTED_LEVEL	1	Pegelauswahl
SAVED_INVERTED_IN	1	Invertierung der Eingabe
SAVED_INVERTED_OUT	1	Invertierung der Ausgabe
SAVED_NOVIO_ACTION	1	Reaktion, wenn keine externe VIO
SAVED_NIOVIO_LEVEL	1	Der automatisch aufzuschaltende VIO-Pegel
SAVED_VIO_ANNOUNCE_MODE	1	Darstellung der VIO-Warnungsanzeigen
SAVED_MACHINE_STATE	1	Maschinenzustand = Art der Anzeige im Normalbetrieb
SAVED_PWM_PERIOD	2	PWM-Periodendauer der Kontrastspannungserzeugung
SAVED_PWM_PULSE	2	PWM-Impulsauer der Kontrastspannungserzeugung
SAVED_LO_THRESHOLD_TTL	1	Schwellenwert LO-Pegel TTL
SAVED_HI_THRESHOLD_TTL	1	Schwellenwert HI-Pegel TTL
SAVED_LO_THRESHOLD_CM5	1	Schwellenwert LO-Pegel CMOS 5 V
SAVED_HI_THRESHOLD_CM5	1	Schwellenwert HI-Pegel CMOS 5 V
SAVED_LO_THRESHOLD_CM3	1	Schwellenwert LO-Pegel CMOS 3,3 V
SAVED_HI_THRESHOLD_CM3	1	Schwellenwert HI-Pegel CMOS 3,3 V
SAVED_LO_THRESHOLD_OW1	1	Schwellenwert LO-Pegel OWN1 (CMOS 2,5 V)
SAVED_HI_THRESHOLD_OW1	1	Schwellenwert HI-Pegel OWN1 (CMOS 2,5 V)
SAVED_LO_THRESHOLD_OW2	1	Schwellenwert LO-Pegel OWN2 (CMOS 1,8 V)
SAVED_HI_THRESHOLD_OW2	1	Schwellenwert HI-Pegel OWN2 (CMOS 1,8 V)

*: Der Parameter-Rettungsbereich (ab SAVE_AREA) muß genauso aufgebaut sein wie der Konfigurationsparameterbereich im SRAM (von PARAM_AREA bis PARAM_END).

Grundeinstellungen (Default-Werte):

Bezeichnung	Inhalt
DEFAULT_LO_THRESHOLD	THRESHOLD_LOW_TTL = 0,8 V = 41
DEFAULT_HI_THRESHOLD	THRESHOLD_HIGH_TTL = 2 V = 102
DEFAULT_SELECTED_LEVEL	TTL = 0
DEFAULT_INVERTED_IN	Keine Invertierung = 0
DEFAULT_INVERTED_OUT	Keine Invertierung = 0
DEFAULT_NOVIO_ACTION	Nichts tun = 0
DEFAULT_NIOVIO_LEVEL	Den bisherigen Pegel aufschalten = 3
DEFAULT_VIO_ANNOUNCE_MODE	Kurzzeitanzeige = 0
DEFAULT_MACHINE_STATE	Spitze Glyphs = 0
DEFAULT_PWM_PERIOD	1000
DEFAULT_PWM_PULSE	50
DEFAULT_LO_THRESHOLD_TTL	THRESHOLD_LOW_TTL = 0,8 V = 41
DEFAULT_HI_THRESHOLD_TTL	THRESHOLD_HIGH_TTL = 2 V = 102
DEFAULT_LO_THRESHOLD_CM5	THRESHOLD_LOW_CMOS5 = 1,2 V = 61
DEFAULT_HI_THRESHOLD_CM5	THRESHOLD_HIGH_CMOS5 = 3,5 V = 178
DEFAULT_LO_THRESHOLD_CM3	THRESHOLD_LOW_CMOS3 = 0,8 V = 41
DEFAULT_HI_THRESHOLD_CM3	THRESHOLD_HIGH_CSMOS3 = 2,4 V = 122
DEFAULT_LO_THRESHOLD_OW1	THRESHOLD_LOW_OWN1 = 0,7 V = 35 (CMOS 2,5 V)
DEFAULT_HI_THRESHOLD_OW1	THRESHOLD_HIGH_OWN1 = 1,7 V = 86 (CMOS 2,5 V)
DEFAULT_LO_THRESHOLD_OW2	THRESHOLD_LOW_OWN2 = 0,6 V = 30 (CMOS 1,8 V)
DEFAULT_HI_THRESHOLD_OW2	THRESHOLD_HIGH_OWN2 = 1,2 V = 61 (CMOS 1,8 V)

Die logische Programmschnittstelle der LCD-Anzeige

Die anwendungsseitigen Pixelzugriffe und die Aktualisierung der physischen Anzeige sind vollständig voneinander getrennt. Das Anwendungsprogramm sieht nur ein ein zweidimensionales Array, das der Pixeldarstellung auf der LCD-Anzeige entspricht (logische Bildspeicherorganisation). Das ist der Pixelpuffer. Es wird stets pixelweise geschrieben.

Eine Anzeige wird in zwei Schritten erzeugt:

1. Pixelweises Aufbauen der Darstellung im Pixelpuffer.
2. Übertragen des Pixelpufferinhalts zur LCD-Anzeige (Update).

Die elementaren anwendungsseitigen Funktionen:

Funktion	Aufruf	Parameter
Die Anzeige initialisieren	graph_display_init	–
Die Anzeige aktualisieren	graph_display_update	–
Ein Pixel in den Pixelpuffer eintragen	pix	CH: X-Koordinate (0...179) CL: Y-Koordinate (0...63) AL: Pixelwert. 0 = löschen, 1 = setzen
Den Pixelpuffer mit einem Festwert füllen (byteweise, also nicht Pixel für Pixel)	graph_fill_buff	AL: Festwert (als Byte). Sinnvolle Werte: 0 = alle Pixel löschen 1 = alle Pixel setzen

Ablauf der Initialisierung:

Der Pixelpuffer wird gelöscht. Zu jedem der drei LCD-Controller werden die Kommandos “Display einschalten” = AFH und “Rücksetzen” = E2H übertragen

Ablauf der Aktualisierung:

Der Pixelpufferinhalt wird nacheinander zu jedem der drei Controller übertragen. Jeder Controller erhält $4 \cdot 60 = 240$ Bytes.

Der Übertragungsablauf:

1. Übertragen der Seitenadresse.
2. Übertragen aller 60 Bytes der Seite.
3. Beenden, wenn alle 4 Seiten übertragen wurden. Sonst Ausrechnen der neuen Pufferanfangsadresse (= alte Pufferanfangsadresse + Gesamtanzahl der Spalten (180)). Den Ablauf mit der nächsten Seitenadresse wiederholen.

Der Pixelpuffer

Der Pixelpuffer wird pixelweise zweidimensional adressiert (x-Adresse = Spaltenadresse, y-Adresse = Zeilenadresse). Der Koordinatenursprung $(x, y) = (0, 0)$ liegt links oben. x-Adresse 8 Bits, Werte 0 bis 179; y-Adresse 5 Bits, Werte 0 bis 31.

Pixelpufferorganisation

Ein Byte enthält 8 Pixel, die untereinander dargestellt werden. Bit 0 entspricht dem obersten Pixel, Bit 7 dem untersten.

Die Pixel werden seitenweise für die gesamte Anzeige gespeichert. Eine Seite umfaßt 180 Bytes. Sie entspricht jeweils 8 Pixelzeilen.

Die niedrigstwertigen 3 Bits der y-Adresse ergeben die Bitadresse im Byte. Die verbleibenden 2 Bits sind die Seitenadresse (Seiten 0 bis 3). Sie werden um 3 Bits nach rechts verschoben, so daß sich eine rechtsbündige Seitenadresse ergibt.

Rechengang:

- Seitenanfangsadresse: Seitenadresse • gesamte Spaltenanzahl (180).
- Byteadresse = Seitenanfangsadresse + x-Adresse (16-Bit-Addition).

Aus der Bitadresse im Byte wird eine Bitmaske hergestellt, die an der betreffenden Bitposition eine Eins enthält. Ist das Bit zu setzen, wird die Bitmaske mit dem Byte im Pixelpuffer disjunktiv verknüpft. Ist das Bit zu löschen, wird die Bitmaske invertiert und mit dem Byte im Pixelpuffer konjunktiv verknüpft.

Die physische Ansteuerung der LCD-Anzeige

Ruhezustand:

- Datenbus auf Ausgabe.
- Übertragungssteuerung (RW) auf Schreiben.
- Registerauswahl (RS) auf Datenzugriff.

Die physischen Steuer- und Zugriffsfunktionen:

Funktion	Aufruf	Parameter
Die Anzeige zurücksetzen	graph_reset	–
Ein Datenbyte schreiben	graph_write	AL: Datenbyte
Ein Datenbyte lesen	graph_read	AL: Datenbyte (Rückgabe)
Ein Kommando schreiben	graph_control	AL: Kommandobyte
Ein Statusbyte lesen	graph_sense	AL: Statusbyte (Rückgabe)
Das BUSY-Bit abfragen und auf BUSY=0 warten	graph_busy	–

Hinweise:

1. Die Lesefunktionen werden typischerweise nicht genutzt.
2. Sofern der einzelne Zugriff länger dauert als die minimale Zykluszeit t_{CYC} gemäß Datenblatt, kann auf das Abfragen des BUSY-Bits verzichtet werden; die Zugriffe lassen sich ohne Abfragen oder Warten aneinanderreihen.
3. Zum Datenlesen: nach dem Einstellen der Adresse ist zunächst ein blinder Lesezugriff erforderlich. S. Datenblatt des LCD-Controllers.

Ein Rasterzeichen darstellen

Die Bitmuster der Rasterzeichen sind in einem Zeichengenerator zusammengefaßt. Ein Bitmuster ist jeweils eine feste Anzahl an Bytes. Die Bitmuster folgen gemäß dem Zeichencode (z. B. ASCII) aufeinander. Die jeweilige Byteadresse kann somit auf einfache Weise aus dem betreffenden Zeichencode errechnet werden.

Zeichenraster:

Rasterformat	Bytes je Zeichen im Zeichengenerator	Genutzte Bits
5 • 7	5	35
7 • 9	8	63

Das Zeichenbitmuster ist ein fortlaufender Bitstrom ohne Rücksicht auf Bytegrenzen. Bit 0 entspricht dem äußersten Pixel rechts oben (Bezugspunkt). Die Bits werden vom Bezugspunkt an nach links und unten fortschreitend in den Pixelpuffer eingetragen.

Die anwendungsseitige Darstellfunktionen:

Funktion	Aufruf	Parameter
Ein Zeichen darstellen mit Weiterschaltung zum nächsten Zeichen	show_char	Variable XBASE: X-Koordinate (0...179) Variable YBASE: Y-Koordinate (0...63) Bezugspunkt (x, y): links unten. AL: Zeichencode Variable CHARTYPE, Bit 0: Art der Darstellung: 0 = Bit aus Zeichengenerator eintragen 1 = Bit aus Zeichengenerator negiert eintragen (Invertierung)
Ein Zeichen darstellen mit Weiterschaltung zum nächsten Zeichen	show_char_exec	CH: X-Koordinate (0...179) CL: Y-Koordinate (0...63) Bezugspunkt (x, y): links unten. AL: Zeichencode Variable CHARTYPE, Bit 0: Art der Darstellung: 0 = Bit aus Zeichengenerator eintragen 1 = Bit aus Zeichengenerator negiert eintragen (Invertierung)
Ein Zeichen aus dem Zeichengenerator darstellen	show_a_char	CH: X-Koordinate (0...179) CL: Y-Koordinate (0...63) Bezugspunkt (x, y): rechts oben. AL: Zeichencode Variable CHARTYPE, Bit 0: Art der Darstellung: 0 = Bit aus Zeichengenerator eintragen 1 = Bit aus Zeichengenerator negiert eintragen (Invertierung)
Ein Pixelfeld löschen	show_pixel_array	CH: X-Koordinate (0...179) CL: Y-Koordinate (0...63) Bezugspunkt links unten. BH: Breite (Spalten) BL: Höhe (Zeilen) Variable CHARTYPE, Bit 0: Art der Darstellung: 0 = Pixel löschen (Feld dunkel) 1 = Pixel setzen (Feld hell)

Weiterschaltung

Nach Ausführung der Funktion zeigt die x-Koordinate (in XBASE und CL) auf die erste freie x-Position neben rechts dem Zeichen. Der Zeichenabstand wird nicht berücksichtigt.

Zeichensatz- und Menüfeldparameter:

Bezeichnung	Bedeutung
char_chargensize	Länge eines Zeichenbitmusters im Zeichengenerator (in Bytes)
char_cols	Zeichenbreite = Anzahl der Spalten
char_rows	Zeichenhöhe = Anzahl der Zeilen
char_space	Zeichenabstand (Spalten)
menue_x_rim	Der linke und rechte Rand der Menüfelder (Breite in Pixeln)
menue_y_rim	Der obere und untere Rand der Menüfelder (Höhe in Pixeln)

Menüelement (menue items) darstellen

Menüelemente können als einfache Zeichenketten oder als Menüfelder dargestellt werden.

Das Menüfeld

Ein Menüfeld ist eine rechteckige Pixelfläche, die mit einer Beschriftung (Zeichenkette) gefüllt werden kann. Sie kann normal oder invers oder auch nicht dargestellt werden (Dunkeltastung). Menüfelder werden mit Menüdeskriptoren beschrieben.

Der Menüdeskriptor:

Byte	Inhalt	Anmerkungen
1.	x-Koordinate	Bezugspunkt links unten. FFH: Wert aus Variable MENUE_ITEM_X entnehmen
2.	y-Koordinate	FFH: Wert aus Variable MENUE_ITEM_Y entnehmen
3.	Breite (in Pixeln)	00H: Wert aus Zeichenkette berechnen $\text{Breite} = \text{Anzahl der Zeichen} \cdot (\text{Zeichenbreite} + \text{Zeichenabstand}) - \text{Zeichenabstand} + 2 \cdot \text{x-Rand}$
4.	Höhe (in Pixeln)	00H: Wert aus Zeichenkette berechnen $\text{Höhe} = \text{Zeichenhöhe} + 2 \cdot \text{y-Rand}$
5. und ggf. weitere	Beschriftung	Zeichenkette
das letzte	00H	Endekennung (vgl. Programmiersprache C)

Die anwendungsseitige Darstellfunktionen:

Funktion	Aufruf	Parameter
Ein Menüfeld darstellen	show_menue_item	Z: Deskriptoradresse (Byteadresse) Variable CHARTYPE: Art der Darstellung: 0 = normale Darstellung 1 = inverse Darstellung 2 = nichts darstellen (Dunkeltastung, Blankoff) 3 = nichts invers darstellen (Hellastung; rechteckige Fläche)
Ein Menüfeld zur Anzeige vorbereiten	show_menue_array	Wie vorstehend. Die Anfangskordinaten werden in die Variablen XBASE und YBASE eingetragen
Die Zeichenkette im vorbereiteten Menüfeld darstellen	show_menue_item_string	Z: Zeichenkettenadresse Variable XBASE: die x-Koordinate (Bezugspunkt) Variable YBASE: die y-Koordinate (Bezugspunkt) Variable CHARTYPE: Art der Darstellung
Eine Zeichenkette in den Pixelpuffer eintragen	show_menue_chars	Z: Zeichenkettenadresse CH: x-Koordinate CL: y-Koordinate. Bezugspunkt links unten Variable CHARTYPE: Art der Darstellung

Einen Glyph darstellen

Ein Glyph ist ein Pixelmuster in einem rechteckigen Pixelfeld, das als Bitkette vorliegt und in den Pixelpuffer eingetragen wird.

Das Pixelmuster wird in Bytes binär codiert. Der Bezugspunkt liegt links oben. Bitposition 7 des ersten Bytes entspricht dem Pixel des Bezugspunktes. Bezugspunktkoordinaten, Höhe und Breite werden als Parameter übergeben.

Die Bytes des Pixelmusters werden zeilenweise interpretiert. Beim Übergang in die nächste Pixelzeile wird das jeweils nächste Byte adressiert.

Diese Konvention erlaubt es, das Pixelmuster aus einer Skizze des Glyphs auf einfache, naheliegende Weise von Hand zu erfassen:

- Die gewünschte Form in ein rechteckiges Raster einzeichnen.
- Das erste Pixel links oben = Bit 7 des ersten Bytes. Dann Bit für Bit umsetzen, bis die gesamte Rasterzeile erledigt ist.
- Für die nächste Rasterzeile ein neues Byte beginnen (Bit 7 usw.).

Die anwendungsseitige Darstellungsfunktionen:

Funktion	Aufruf	Parameter
Einen Glyph darstellen	show_a_glyph	Z: Bitmusteradresse (Byteadresse) Variable: GLYPH_X: die x-Koordinate (Bezugspunkt) GLYPH_Y: die y-Koordinate (Bezugspunkt) GLYPH_LEN: die Breite (in Pixeln) GLYPH_HEIGHT: die Höhe (in Pixeln) CHARTYPE, Bit 0: Art der Darstellung: 0 = normale Darstellung 1 = inverse Darstellung

Portübersicht

Port A – Ausgänge:

7	6	5	4	3	2	1	0
OUT7	OUT6	OUT5	OUT4	OUT3	OUT2	OUT1	OUT0
TS	TS	TS	TS	TS	TS	TS	TS

Ausgabe Open Drain. High: hochohmig; Pegel über Pullup-Widerstand. Low: Ausgabe Datenwert 0.

Port B:

7	6	5	4	3	2	1	0
LCD R/W#	VIO_3EN#	PWM	SWC_B#	SWC_A#	–	PB1	VIO_5EN#
OUT	OUT	OUT	IN	IN	IN	IN	OUT

SWC_B# = Raststellung; SWC_A# = Tastfunktion. PB1 = SCK (Programmer).

Port C – LCD-Bus:

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0
I/O	I/O	I/O	I/O	I/O	I/O	I/O	I/O

Port D:

7	6	5	4	3	2	1	0
SW7#	SW6#	SW5#	SW4#	USB_RX	USB_TX	LCD_RST#	VIO_OC#
IN	IN	IN	IN	OUT	IN	OUT	IN

Port E:

7	6	5	4	3	2	1	0
LCD A0	E3	E2	E1	VIODET (AIN1)	VIOREF (AIN0)	PE1 = SLAVE TX	PE0 = SLAVE RX
OUT	OUT	OUT	OUT	IN	IN	OUT	IN

AIN1, AIN0: Komparatoreingänge. Digitaleingänge ausgeschaltet. PE1 = PDO, PE0 = PDI (Programmer).

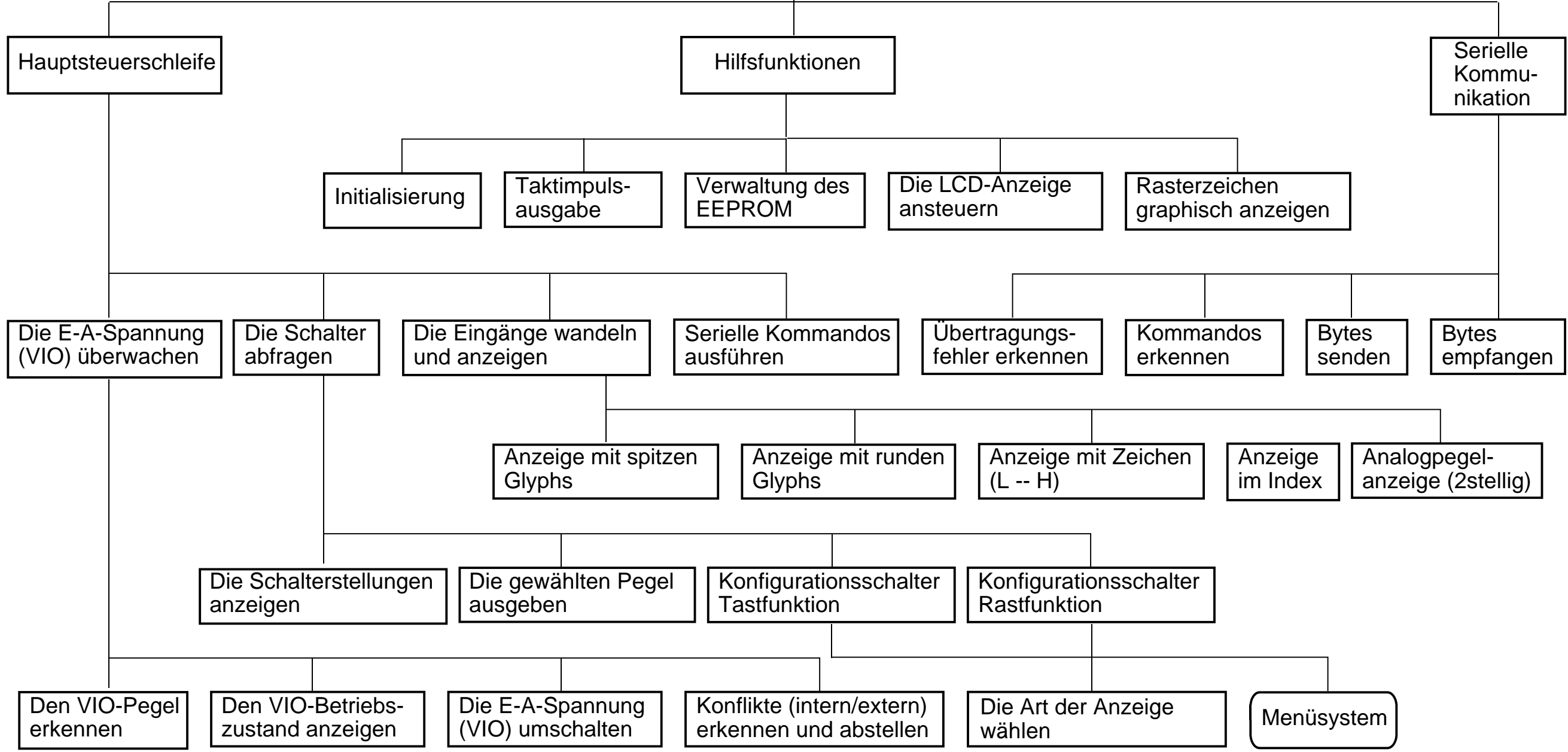
Port F – Eingänge (zum ADC):

7	6	5	4	3	2	1	0
IN7	IN6	IN5	IN4	IN3	IN2	IN1	IN0
IN	IN	IN	IN	IN	IN	IN	IN

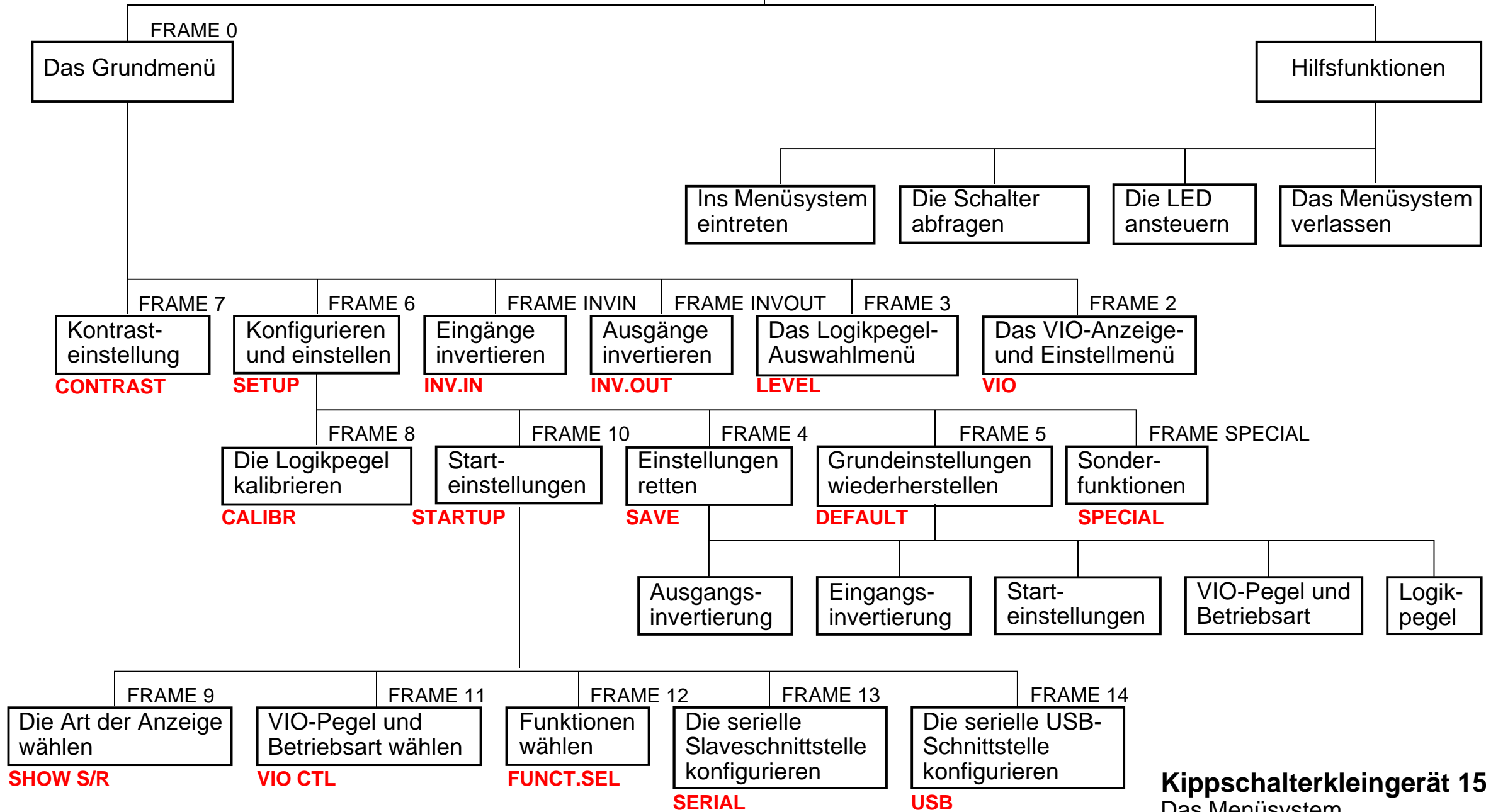
Port G:

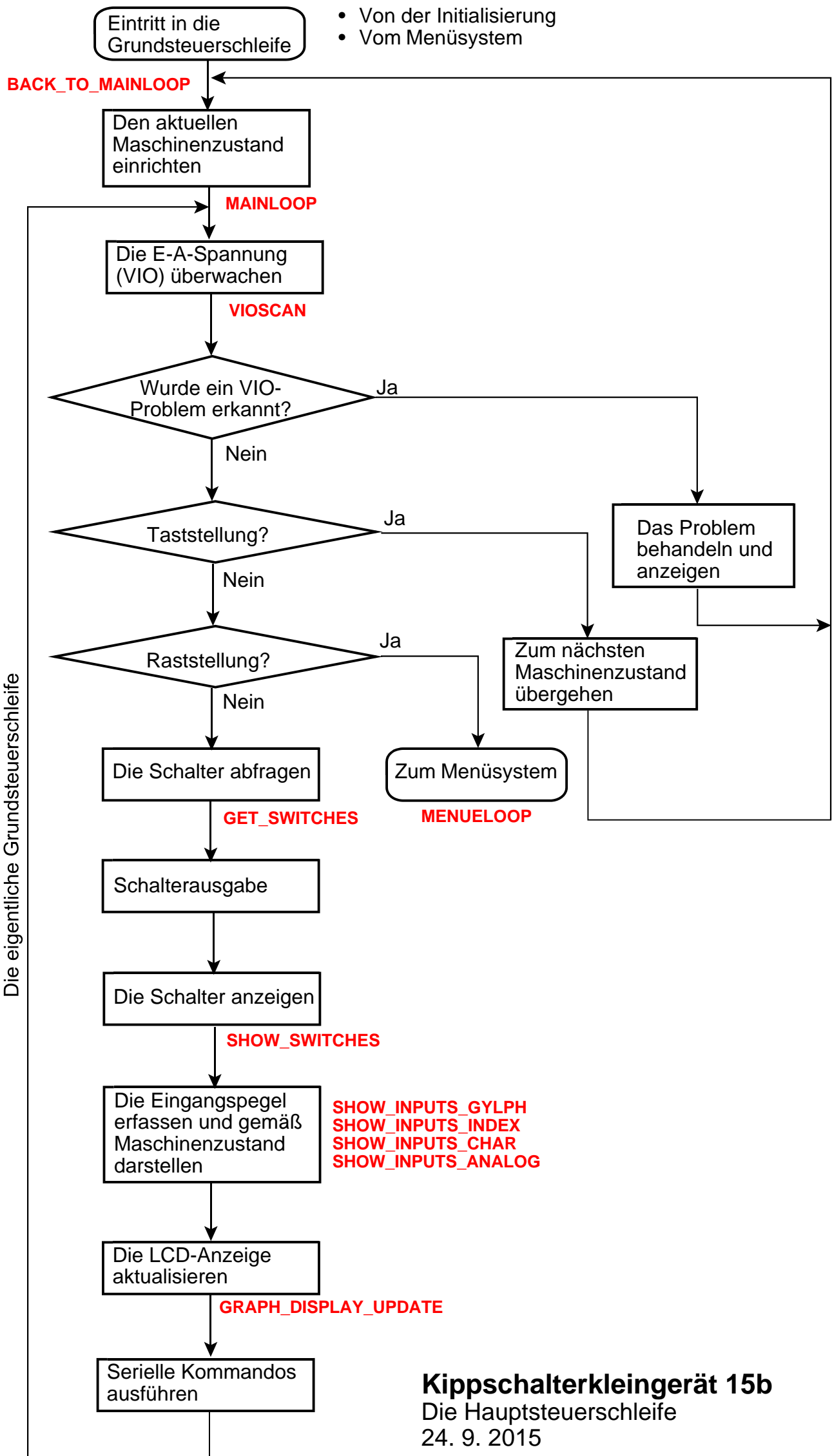
5	4	3	2	1	0
LED_B	LED_A	SW3#	SW2#	SW1#	SW0#
OUT	OUT	IN	IN	IN	IN

Kippschalterkleingerät 15b Programmübersicht

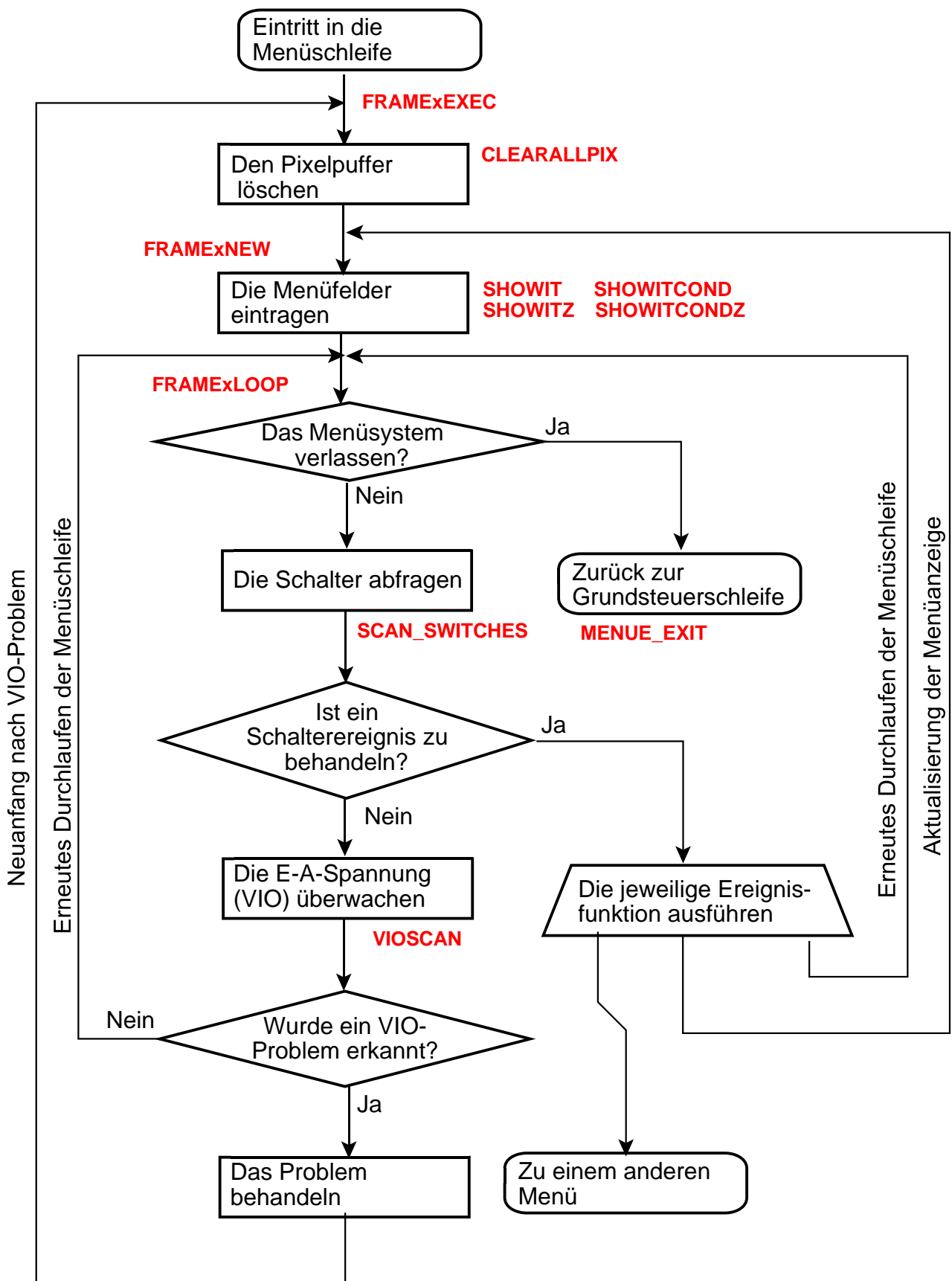


Kippschalterkleingerät 15b Menüsystem

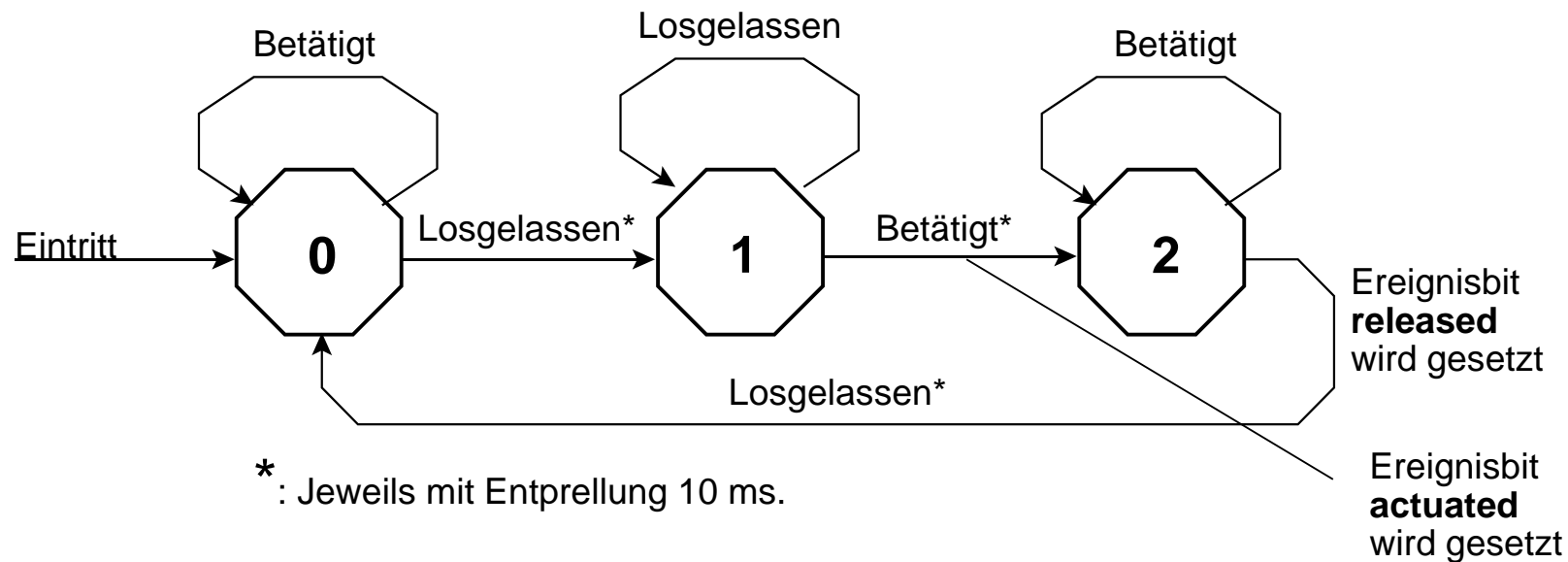




- Von der Initialisierung
- Vom Menüsystem



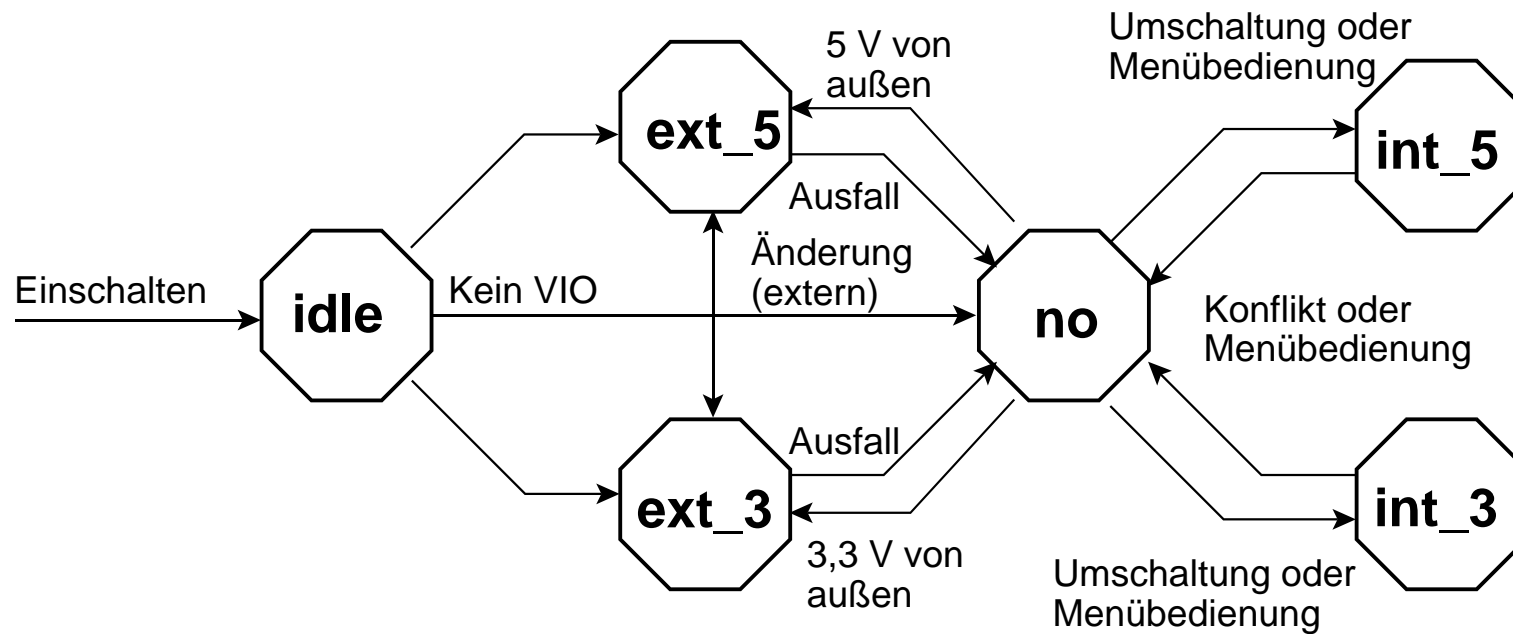
Kippschalterkleingerät 15b
 Die prinzipielle Menüsleife
 13. 10. 2015



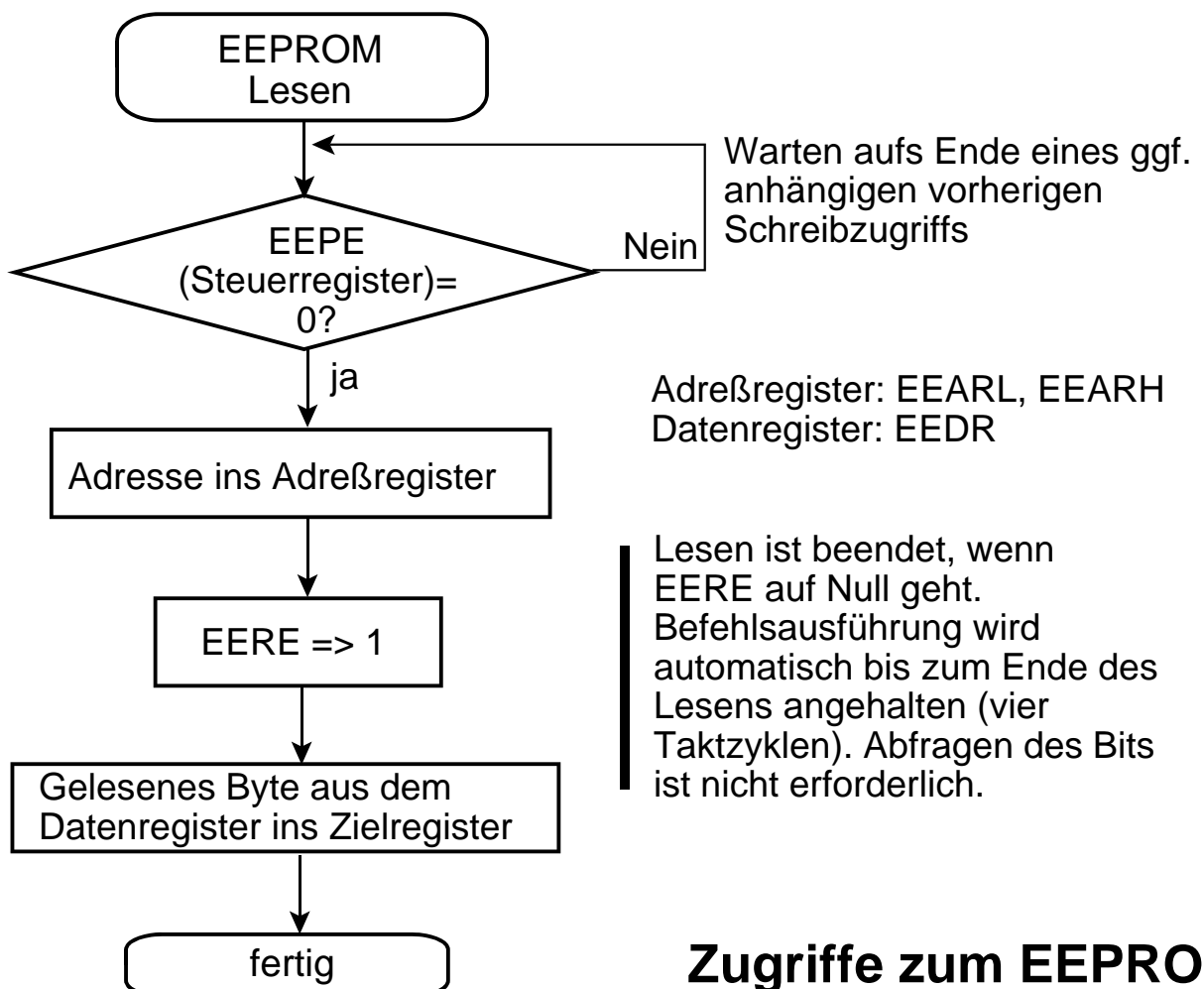
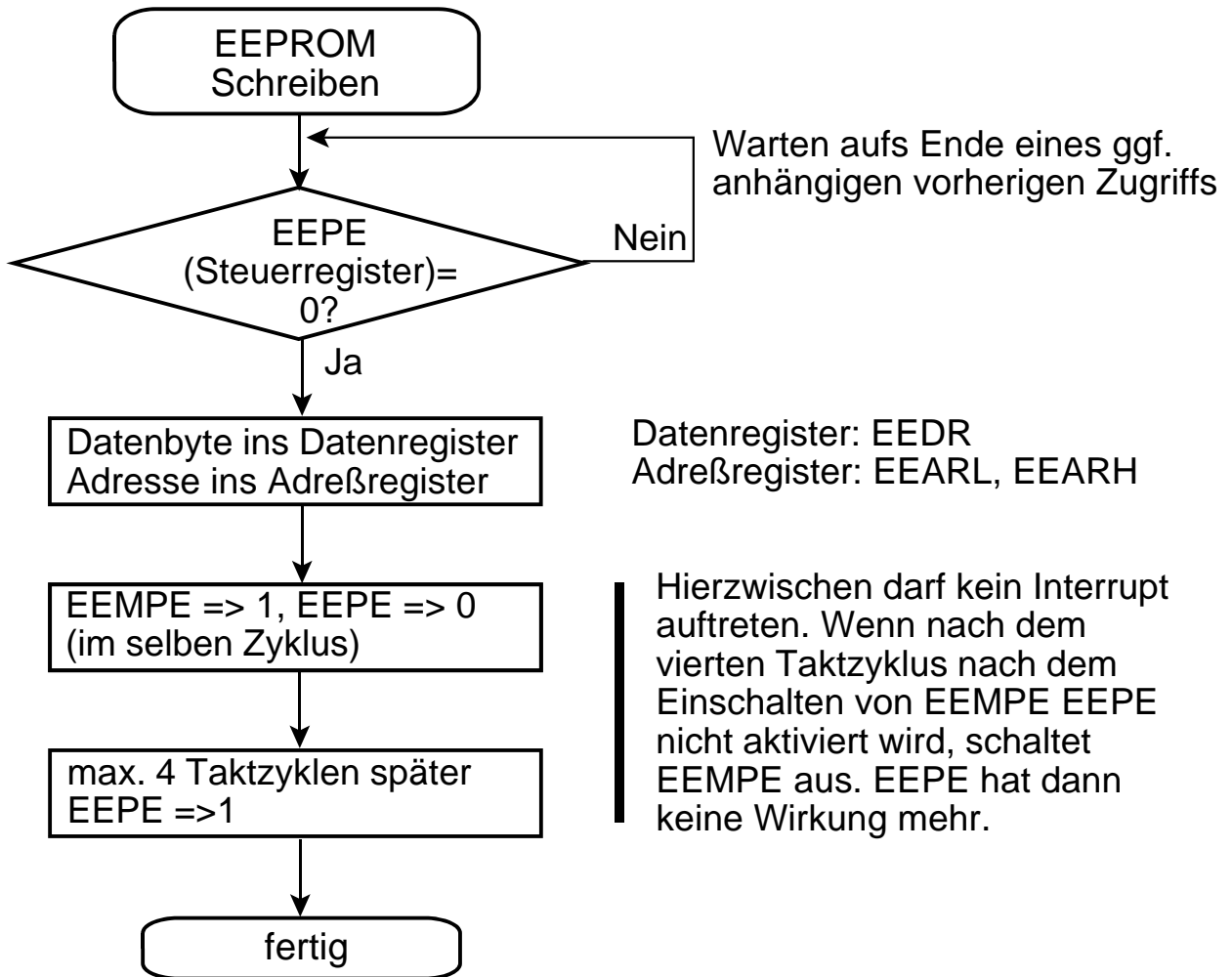
Funktionsauslösung nur bei Übergang aus dem Zustand AUS = losgelassen in den Zustand EIN = betätigt.
 Beim Eintritt in die Funktion muß ein bereits betätigter Schalter zunächst in Stellung AUS gebracht werden (loslassen). Deshalb der Anfangszustand 0.

Jeder Schalter hat ein eigenes Zustandsbyte und zwei Ereignisbits:
 1. Betätigt = actuated
 2. Losgelassen = released.
 Für beide Ereignisarten gibt es jeweils ein Ereignisbyte (switches_actuated, switches_released).

Kippschalterkleingerät 15b
 Schalterzustände
 16. 9. 2015

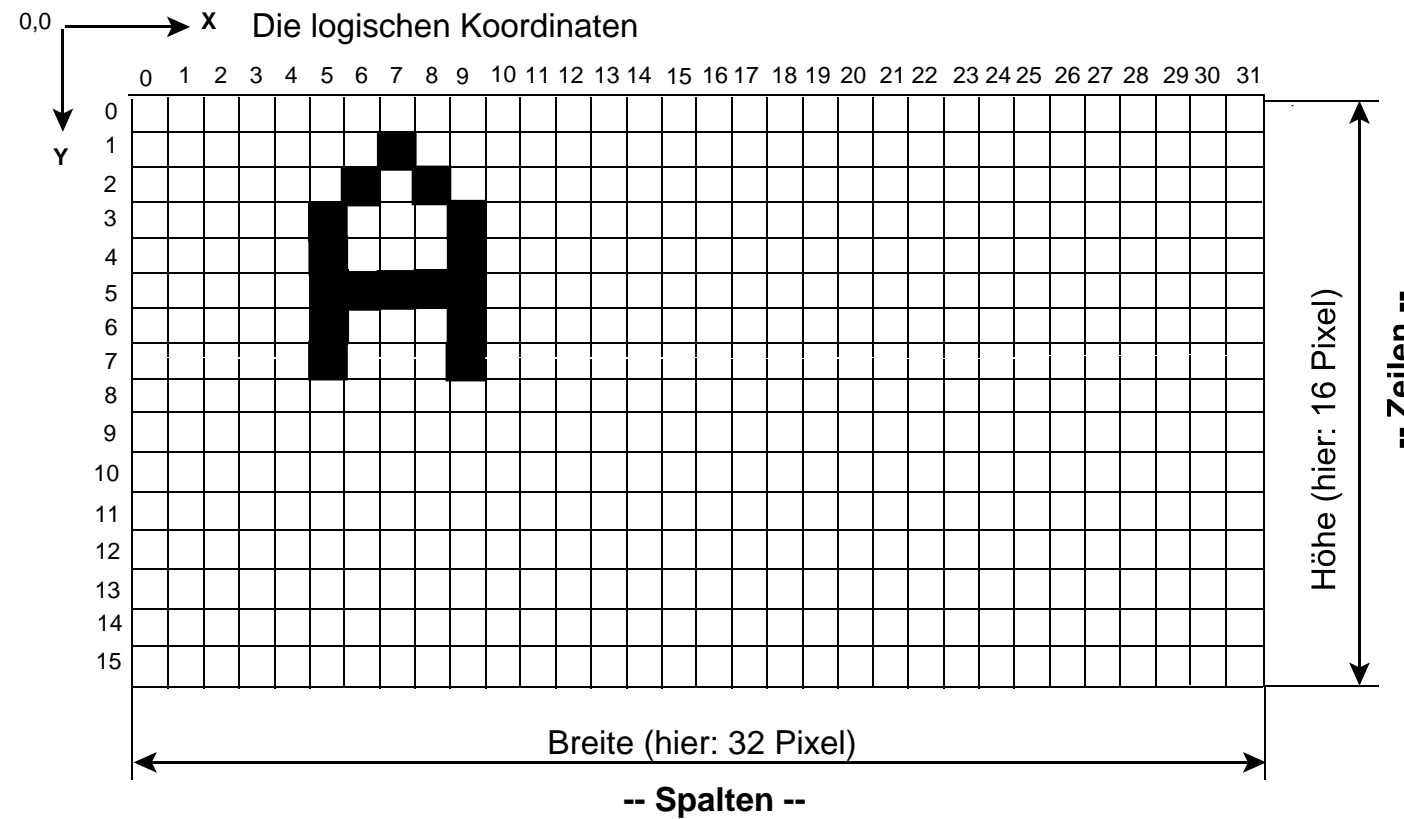


Kippschalterkleingerät 15b
 VIO-Zustände
 13. 10. 2015



**Zugriffe zum EEPROM
Atmel AVR**
(Nach Atmel-Datenblättern)
16. 9. 2015

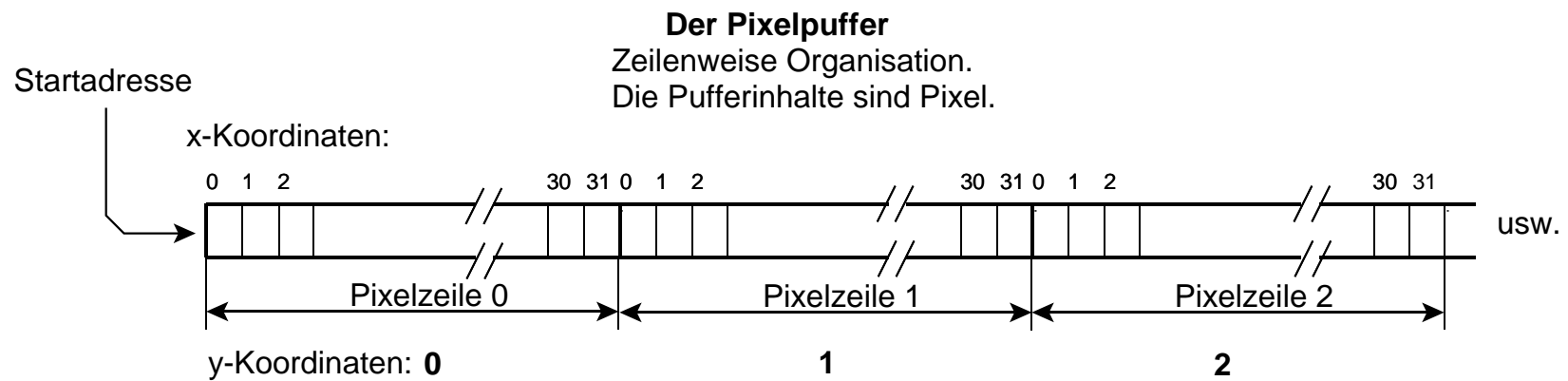
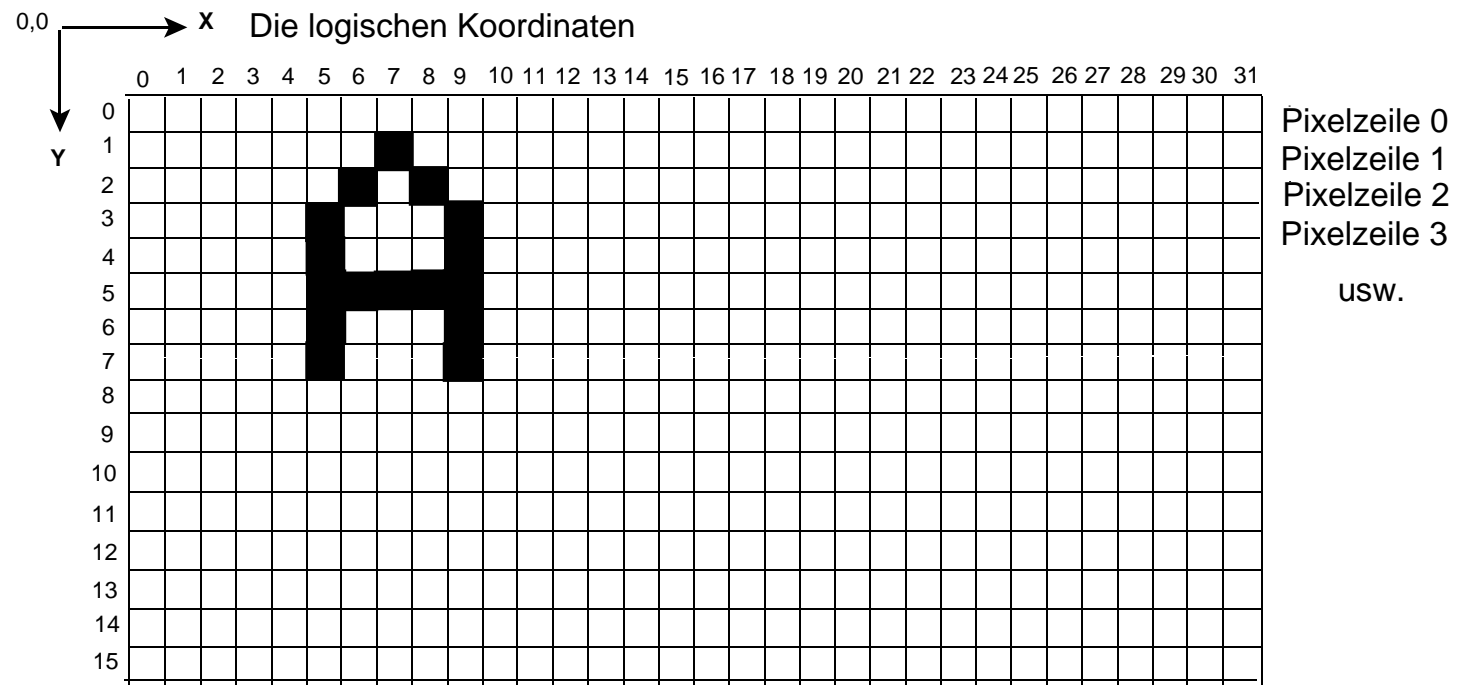
Koordinatenursprung
links oben (vgl. Windows)

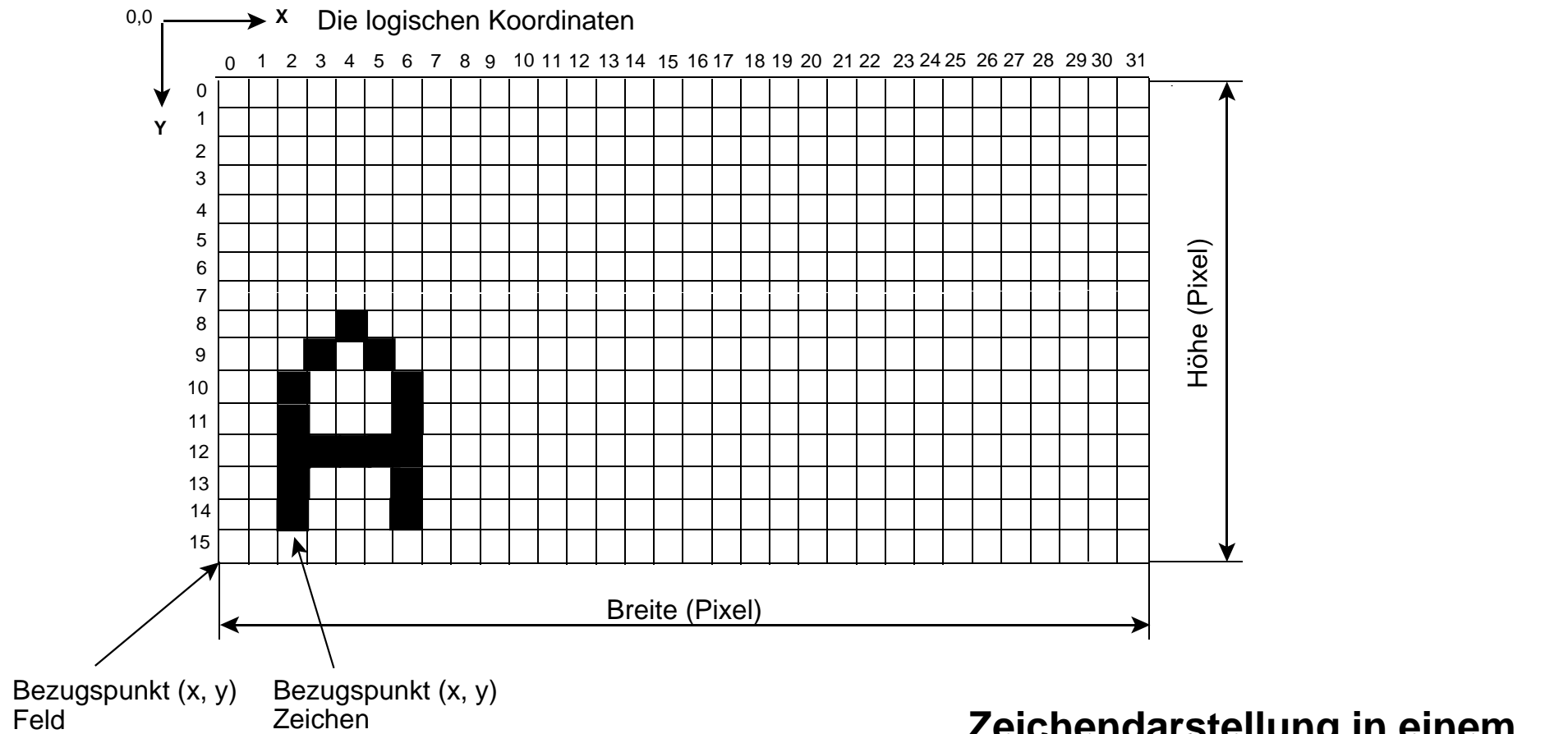


Das Anzeigeformat: Spalten • Zeilen.
Im Beispiel: 32 • 16 Pixel

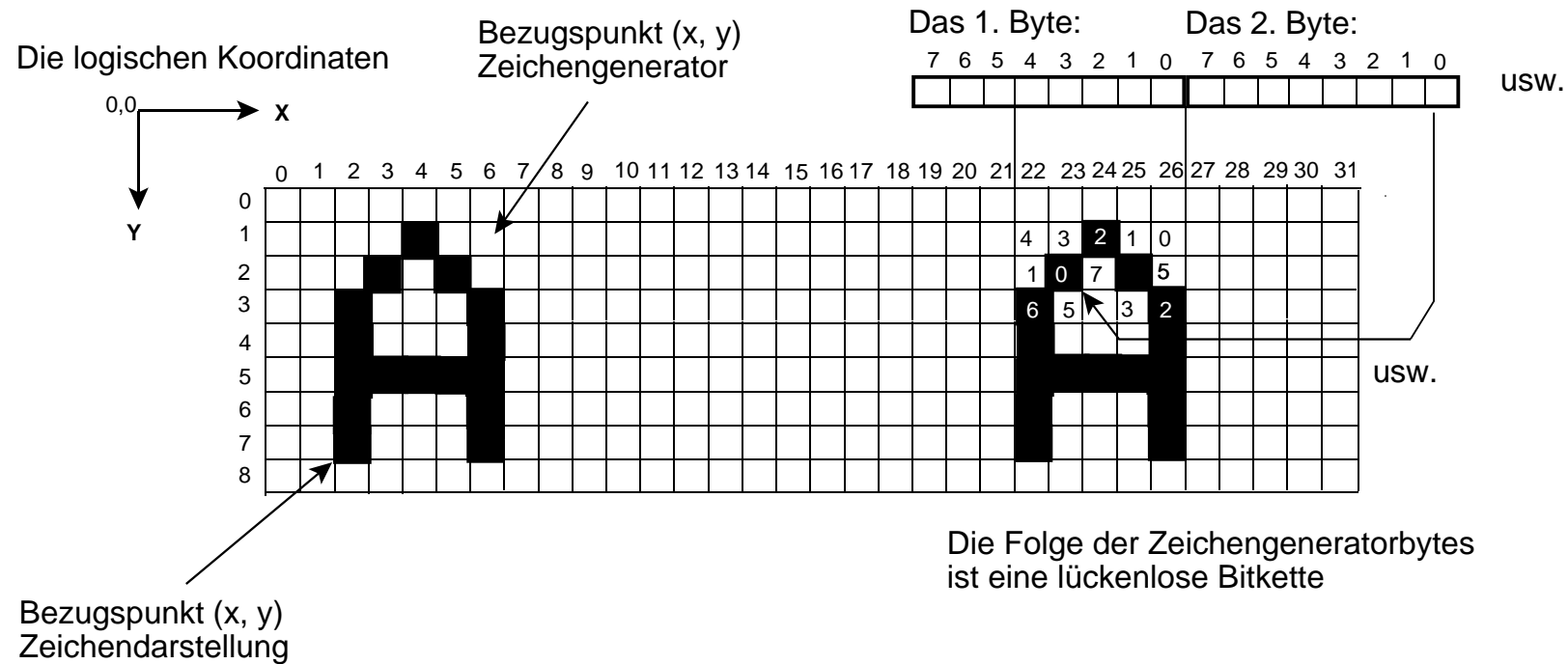
**Die logische Organisation einer
einfachen graphischen Anzeige**
Stand: 10. 10. 2015

Koordinatenursprung

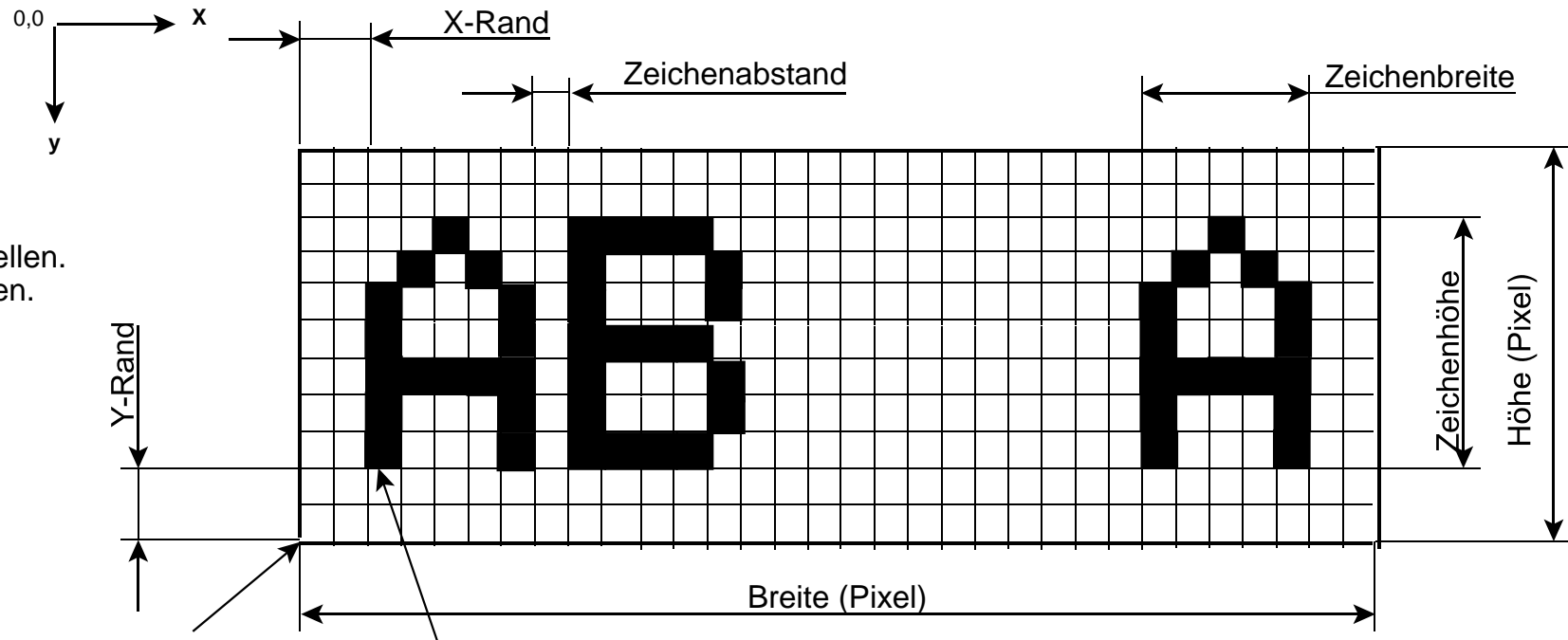




**Zeichendarstellung in einem
graphischen Feld (Pixel Array)**
Stand: 1.9. 15



**Graphische Zeichendarstellung
mittels Zeichengenerator**
Stand: 1. 9. 15



Programmablauf:

1. Das Menüfeld darstellen.
2. Die Zeichen eintragen.

Bezugspunkt (x, y)
Menüfeld

Bezugspunkt (x, y)
Zeichen

Zeichensatzparameter:
char_chargensize: Länge eines
Zeichenbitmusters im Zeichengenerator (in
Bytes)

Ein Menüdeskriptor im Flash:

- X-Koordinate des Bezugspunktes
- Y-Koordinate des Bezugspunktes
- Breite
- Höhe
- Text. Abschluß mit 00H (vgl. Programmiersprache C)

Wenn Breitenwert = 0, wird die Breite aus der Anzahl der Zeichen berechnet.

Breite = Anzahl der Zeichen • (Zeichenbreite + Zeichenabstand) - Zeichenabstand + 2 • X-Rand

Wenn Höhenwert = 0, wird die Höhe aus der Zeichenhöhe berechnet.

Höhe = Zeichenhöhe + 2 • Y-Rand

char_cols: Zeichenbreite = Anzahl der Spalten

char_rows: Zeichenhöhe = Anzahl der Zeilen

char_space: Zeichenabstand = Abstand zwischen den Zeichen

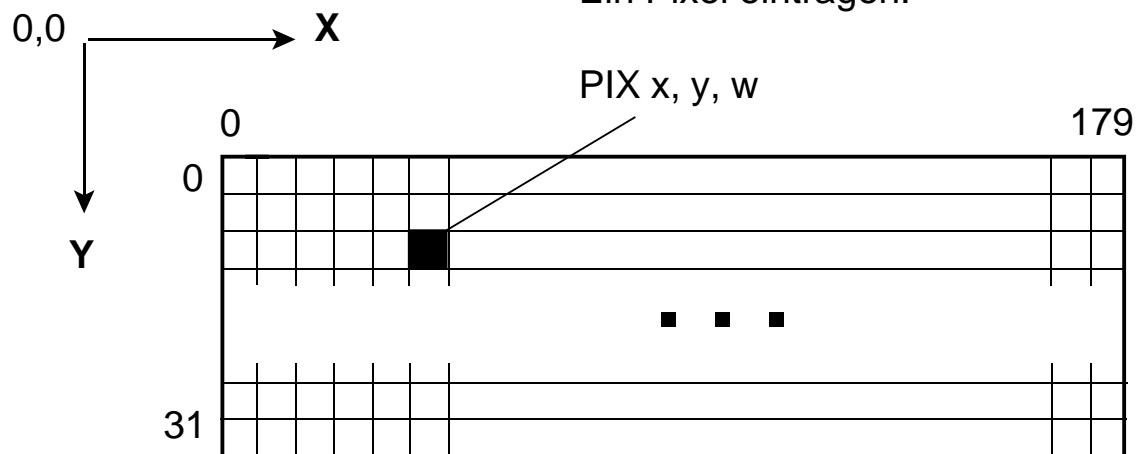
menue_x_rim: X-Rand = Randbreite Menüfeld rechts und links

menue_y_rim: Y-Rand = Randbreite Menüfeld oben und unten

Ein Menüfeld
Stand: 10. 10. 15

-- Der Pixelpuffer --

Ein Pixel eintragen:



-- Die logische Programmschnittstelle --

GRAPH_DISPLAY_INIT

PIX x, y, v

GRAPH_FILL_BUFF fill_char

GRAPH_PIXEL_ARRAY x, y, w, h, v

GRAPH_DISPLAY_UPDATE

fill char: sinnvolle Werte sind 0 (alle Pixel löschen) und 1 (alle Pixel setzen).

v = 0: Pixel (x, y) löschen.

v = 1: Pixel (x, y) setzen

-- Die LCD-Anzeige --

Aktualisieren durch Übertragen des gesamten Bildinhaltes

GRAPH_DISPLAY_UPDATE

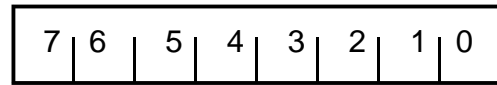


LCD-Anzeige 180 • 32

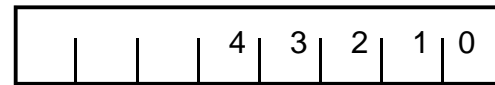
Die logische Programmschnittstelle
Stand: 10. 10. 2015

Die logischen Koordinaten:

Die X-Adresse:



Die Y-Adresse:

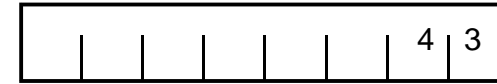


LCD-Anzeige 180 • 32

Pixeladressierung im Pixelpuffer

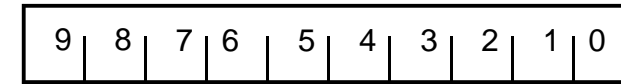
Stand: 10. 10. 2015

Seitenadresse → : 8 =



• 180 =

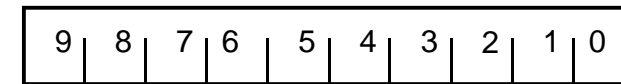
Seitenanfangsadresse im Pixelpuffer



Seitenanfangsadresse

X-Adresse → + =

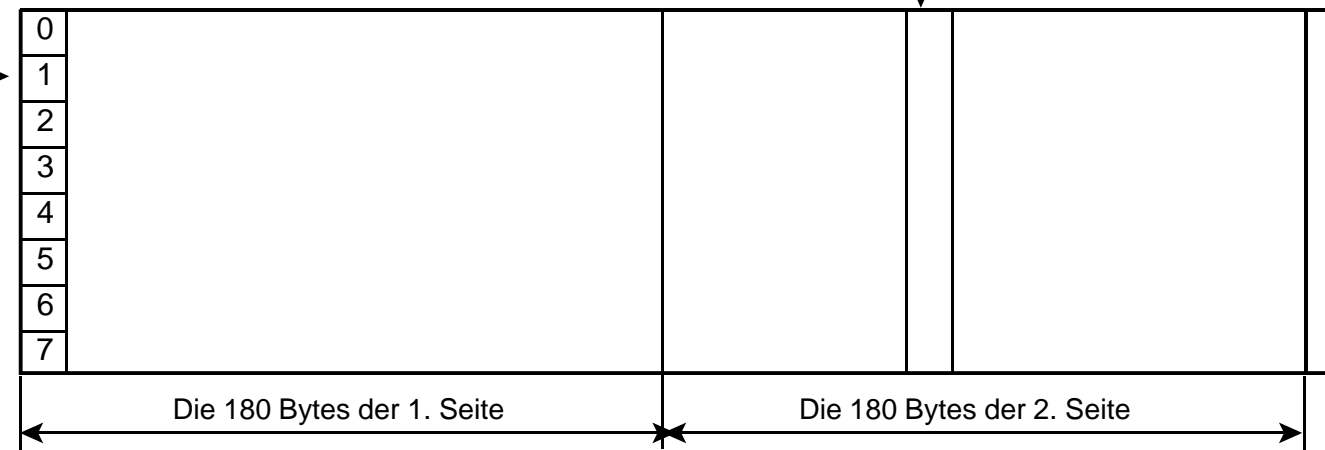
Byteadresse im Pixelpuffer



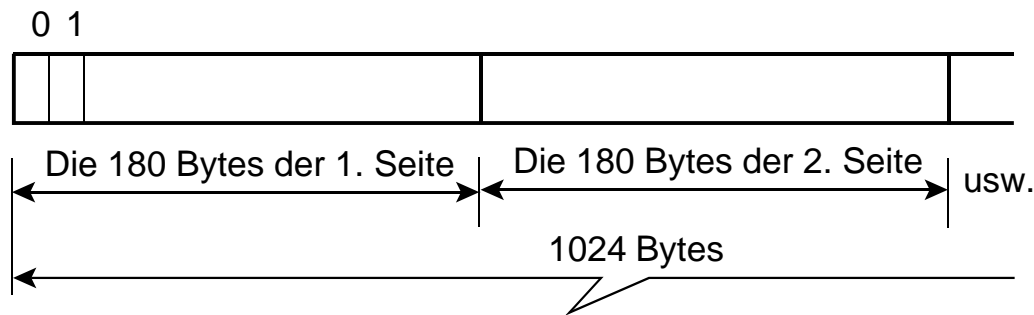
Byteadresse

Bit im Byte

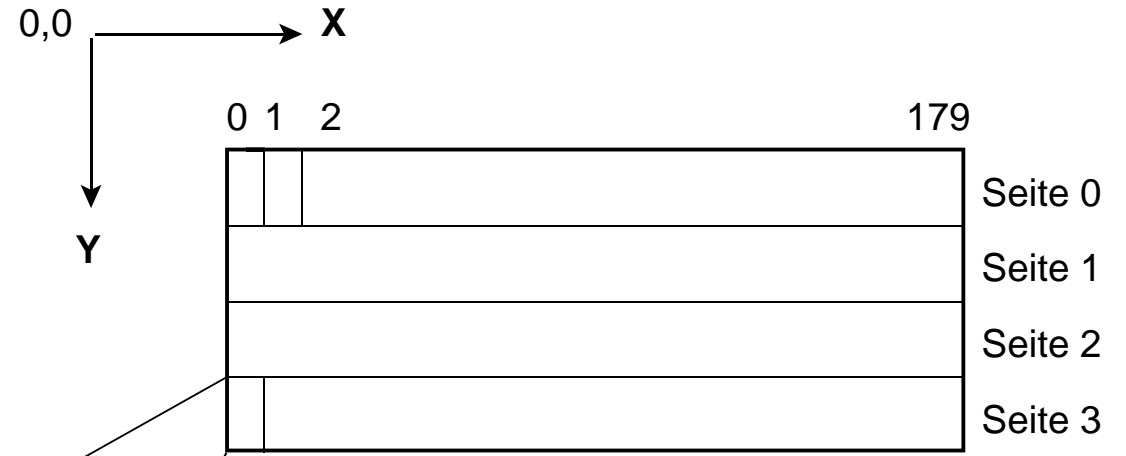
Der Pixelpuffer



Der Pixelpuffer im RAM (insgesamt 720 Bytes):

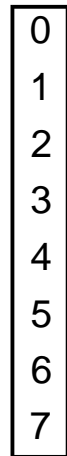


Die logischen Koordinaten:



Ein Datenbyte:

Die Datenbytes der Displayschnittstelle enthalten acht untereinander stehende Pixel. Die automatische Adreßzählung im Display läuft aber in Zeilenrichtung.

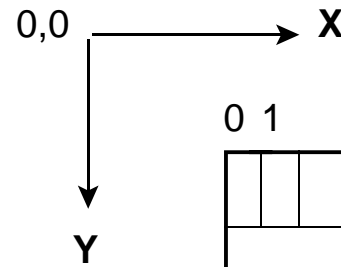


LCD-Anzeige 180 • 32

Adressierung des Puffers in der Anzeige

Stand: 3. 3. 14

Die logischen Koordinaten:

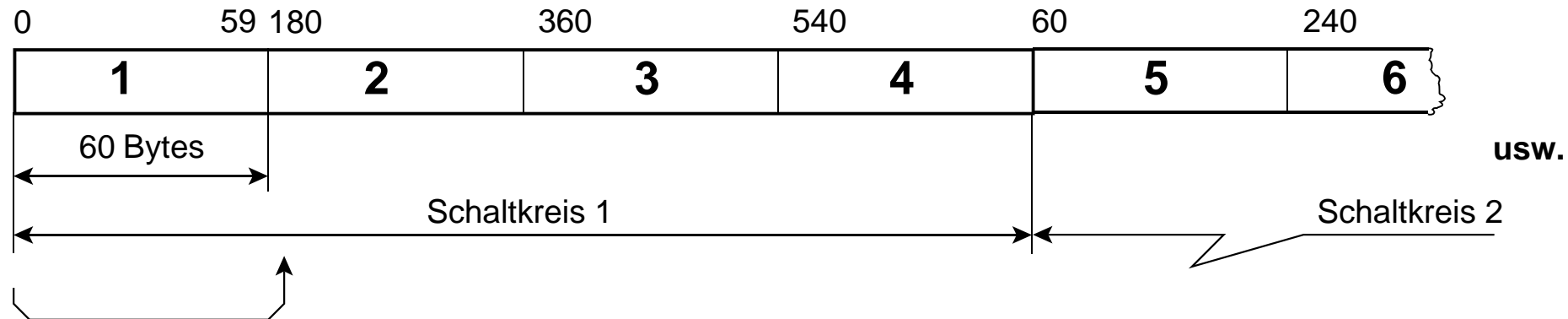


1 bis 12 sind Blöcke zu je 60 Bytes.

	0	1	60	120	179		
		1		5		9	Seite 0
		2		6		10	Seite 1
		3		7		11	Seite 2
		4		8		12	Seite 3

Die blockweise Byteübertragung zu den 3 Schaltkreisen:

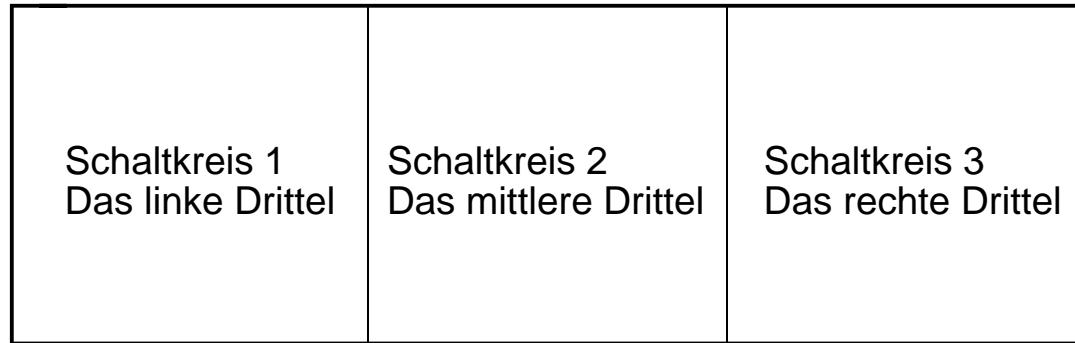
Schaltkreis 1
Signal E1 Schaltkreis 2
Signal E2 Schaltkreis 3
Signal E3



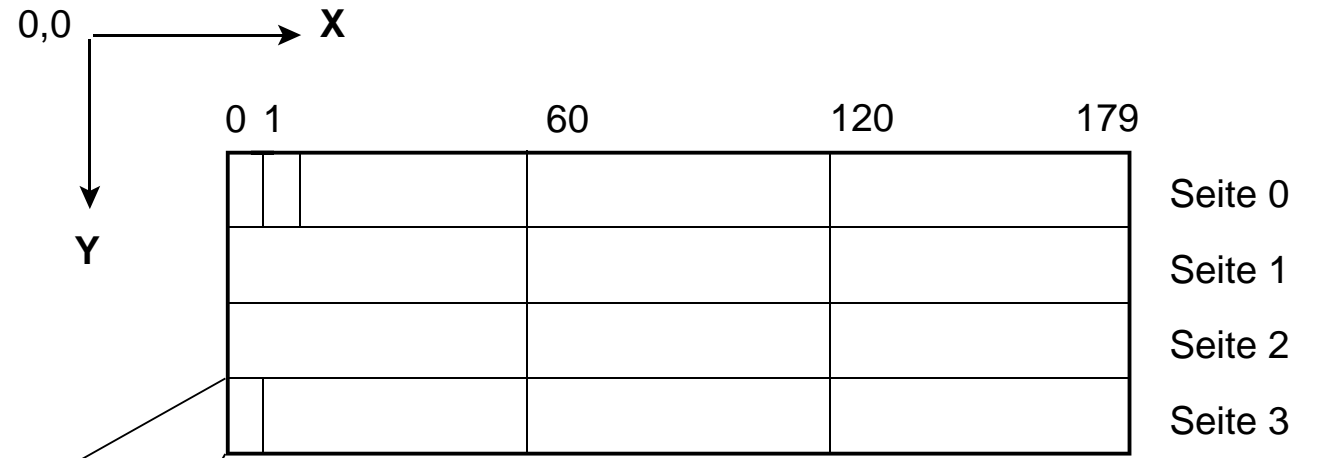
Anfangsadresse des nächsten Blocks =
Anfangsadresse des vorhergehenden + 180

LCD-Anzeige 180 • 32
Adressierung des Puffers in der Anzeige
Stand: 10. 10. 2015

Die LCD-Anzeige 180 • 32 enthält drei Steuerschaltkreise:



Die logischen Koordinaten:



Ein LCD-Datenbyte:

Die Datenbytes enthalten acht untereinander stehende Pixel.

Die automatische Adreßzählung im Display läuft aber in Zeilenrichtung.

- 0
- 1
- 2
- 3
- 4
- 5
- 6
- 7

Schaltkreis 1
Signal E1 Schaltkreis 2
Signal E2 Schaltkreis 3
Signal E3

LCD-Anzeige 180 • 32

Adressierung des Puffers in der Anzeige
Stand: 10. 10. 2015