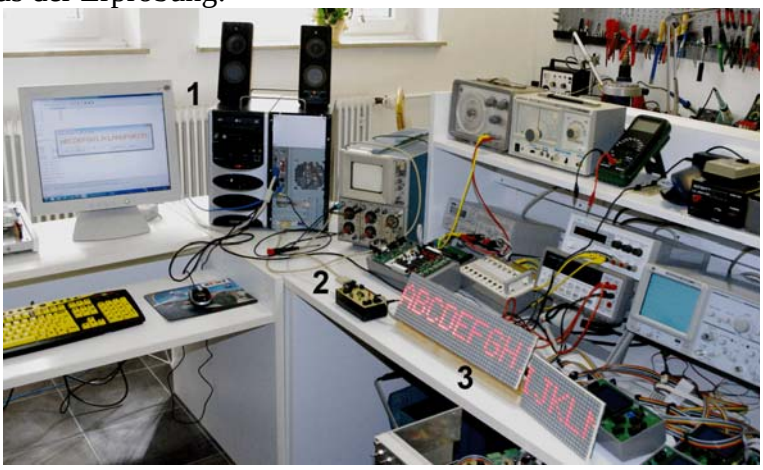
Großanzeige neuer Ausführung (Variante 13a)
Anzeigemodule 10b mit Modulsteuerung 13

Stand: 1.1 vom 9. 8. 2013

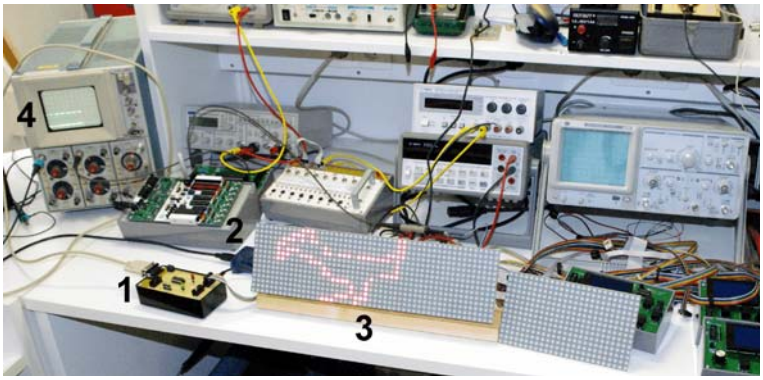
1. Überblick

Die Großanzeige neuer Ausführung (Variante 13a) besteht aus Anzeigemodulen 10b, wobei auf jedes Anzeigemodul eine Modulsteuerung (Großanzeigeadapter, Versuchsadapter) 13 aufgesteckt ist. Die Modulsteuerung beruht auf einem Mikrocontroller Atmel ATmega48. Der Mikrocontroller speichert den jeweils darzustellenden Bildinhalt (Frame Buffer) und führt das zyklische Auffrischen der LED-Anzeige aus (LED Refresh). Die Darstellung statischer Bildinhalte erfordert keine Eingriffe von außen.

Aus der Erprobung:

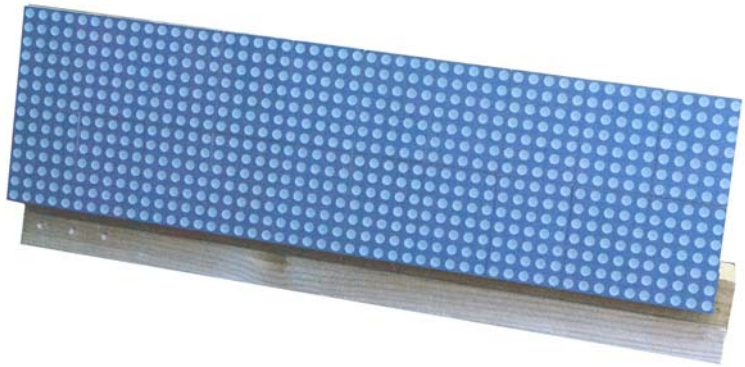


1 - Computer mit Anzeigesteuer- und Entwicklungssoftware; 2 - Hilfsadapter 13; 3 - drei Anzeigemodule.

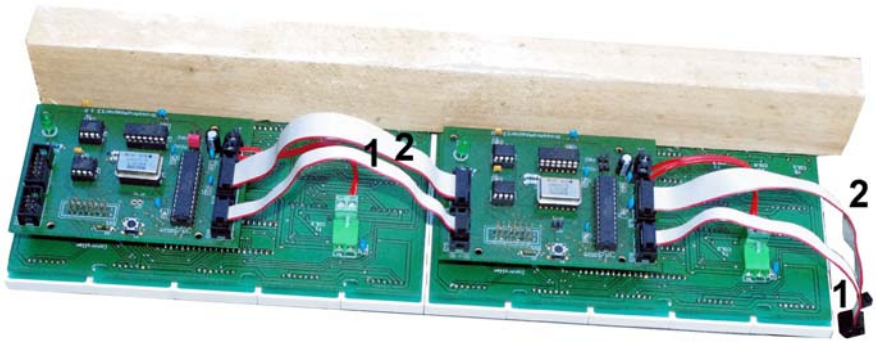


1 - Hilfsadapter 13; 2 - USB-Programmer AVRISPII; 3 - Anzeigemodule; 4 - Kontrolle der seriellen Übertragung.

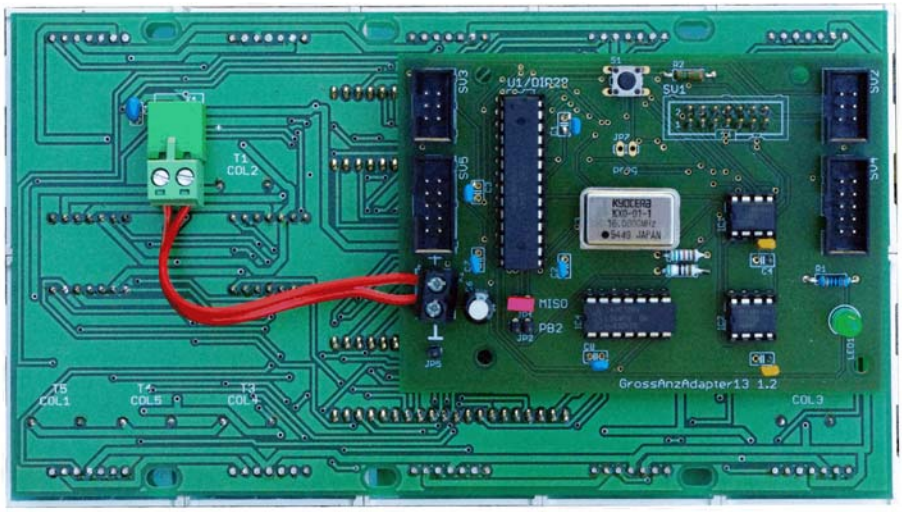
Zwei Anzeigemodule in einem Versuchsaufbau:



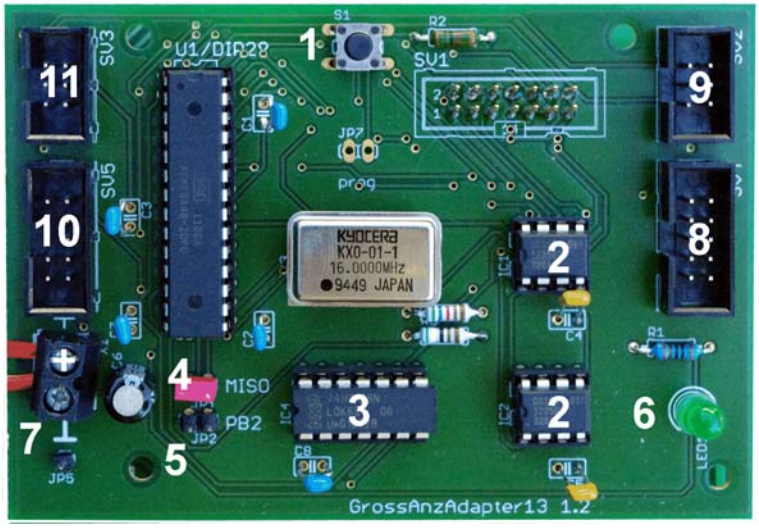
Rückansicht. 1 - Programmierkabel, 2 - Kommunikations- und Stromversorgungskabel.



Ein Anzeigemodul 10b mit aufgesetzter Modulsteuerung 13:



Eine Modulsteuerung 13:



1 - Rücksetztaste; 2 - Digitalpotentiometer; 3 - Kommunikationswegtreiber und Umschaltung; 4 - MISO-Jumper; 5 - Jumper oder Meßpunkt (frei verfügbar); 6 - Anzeige-LED; 7 - Betriebsspannungsklemmen für Anzeigemodul; 8 - ankommendes Kommunikations- und Stromversorgungskabel; 9 - ankommendes Programmierkabel; 10 - abgehendes Kommunikations- und Stromversorgungskabel; 9 - abgehendes Programmierkabel.

Das Anzeigevermögen

Ein Modul ist ein Array aus 25 • 14 Bildpunkten (Pixeln). Die Darstellung ist einfarbig (ein oder aus, keine Helligkeitsmodulation). Beliebig viele Module können horizontal und vertikal aneinandergereiht werden. Die grundsätzlichen Grenzen des Anzeigevermögens ergeben sich vor allem aus folgendem:

- Datenrate. Jeweils 7 Pixel werden als ein Byte binär gespeichert. Ein Bild erfordert somit 2 • 25 = 50 Bytes. Bei zeichenorientierter Datenübertragung (8-Bit-Betrieb) wird jedes Byte mit zwei Hexadezimalzeichen übertragen. Ein Bildinhalt entspricht somit 100 Bytes. Deren Übertragung dauert bei 115 000 Baud etwa 10 ms. Die Aktualisierung einer Anzeige aus 10 Modulen erfordert etwa 100 ms.
- Adressierung. Die Moduladresse ist ein Byte lang. Damit lassen sich maximal 256 Module adressieren.
- Stromversorgung. Über die Kommunikationskabel können nicht mehr als etwa 10 Module sicher versorgt werden.
- Kabellänge. Kommunikationskabelwge (Kabel + Leiterplatten) nicht länger als etwa 2 m, Programmierkabelwege nicht länger als etwa 1 m. Längere Kabelwege sollten in Abschnitte zerlegt werden, die an Adapter mit entsprechenden Treiberstufen angeschlossen sind.

Praktikable Anzeigetafeln bestehen beispielsweise aus 5, 10 (2 • 5) oder 20 (4 • 5) Modulen (das entspricht 14 • 125, 28 • 125 oder 56 • 125 Pixel). Noch größere Anzeigen kann man aus mehreren derartigen Einheiten aufbauen.

Versuchs- und Endausführung
Die Variante 13a ist eine Versuchsausführung. Die Modulsteuerung ist in Form einer experimentellen Adapterplatine auf das Anzeigemodul aufgesteckt. Es gibt keine Vorkehrungen zur mechanischen Befestigung o. dergl. In der Endausführung soll die eigentliche LED-Anzeige mit den LED-Arrays, Treiberschaltkreisen und Treibertransistoren zusammen mit der Steuerung auf einer einzigen Leiterplatte untergebracht werden[1].

Kabelverbindungen
Die Kabel sind von Modul zu Modul geführt (Daisy-Chain-Verkabelung). Eingespeist wird typischerweise (beim Blick von vorn auf die LED-Arrays) in das Modul links außen. Die Verbindungen führen dann von links nach rechts.

Kommunikations- und Stromversorgungskabel
Es sind Flachbandkabel mit 10 Leitungen. Eine Leitung dient der Kommunikation, fünf Leitungen führen Massepotential, vier die Betriebsspannung (VCC = + 5 V). Ein Signalweg (Kabel und Leiterplatten) sollte nicht wesentlich länger als 2 m sein. Das entspricht maximal 10 Modulen.

Kommunikationsprinzip
Serielle Schnittstelle. Informationsfluß in eine Richtung. Es gibt zwei Betriebsarten:

1. Daisy-Chain-Betrieb. Das Modul empfängt ankommende Daten und sendet ggf. Daten zum nächsten Modul (Weitergabe). Die ankommende und die abgehende Kommunikationsleitung sind funktionell voneinander getrennt. Die ankommende Leitung ist mit dem Empfängereingang (RX) des Mikrocontrollers verbunden, die abgehende mit dem Sendeausgang (TX).

2. Busbetrieb. Die ankommenden und die abgehenden Kommunikationsleitungen sind in den Modulen miteinander verbunden. Alle Module empfangen das gleiche Signal. Senden ist nicht möglich.

Programmierkabel
Es sind Flachbandkabel mit 6 Leitungen. Vier Leitungen führen die SPI-Programmiersignale. Eine Leitung ist mit Masse belegt, eine mit der Betriebsspannung (VCC = + 5 V).

Die Programmierkabel werden nur bei Bedarf installiert[2]. Ein Signalweg (Kabel und Leiterplatten) sollte nicht wesentlich länger als 1 m sein. Das entspricht maximal 5 Modulen.

Programmierprinzip
Alle Mikrocontroller sind parallel an die Programmiersignale angeschlossen, aber nur einer liefert das Rücksignal MISO. Hierzu hat jede Modulsteuerung eine Steckmöglichkeit. Auf einer Modulsteuerung ist der entsprechende Jumper zu setzen, auf allen anderen nicht. Alle

---

1: Eine alternative Aufteilung auf zwei Leiterplatten ist Sache der Kostenoptimierung. Dann werden aber die Fragen der Steckverbindungen, der Stromversorgungszuführung und der mechanischen Befestigung von Grund auf berücksichtigt.

2: Also dann, wenn neue Software in Betrieb genommen wird oder Änderungen einzubringen sind.

Mikrocontroller erhalten die gleiche Programmausstattung. Die Parallelprogrammierung verläßt sich darauf, daß alle Controller ein näherungsweise gleiches Programmierverhalten aufweisen[3]. Ansonsten besteht die Möglichkeit, die Programmierkabel zu entfernen und jeden Mikrocontroller einzeln zu programmieren. Die Steckverbinderbelegung ist zu den üblichen Atmel-AVR-Programmern mit sechspoligem Anschluß kompatibel. In einem solchen Fall ist der MISO-Jumper zu stecken.

3: Was dann sehr wahrscheinlich ist, wenn sie alle aus derselben Charge stammen.

2. Grundsätzliche Wirkungsweise

Jedes Modul hat eigene Bildpuffer, die im SRAM seines Mikrocontrollers untergebracht sind. Die Pixelmuster fester Anzeigen können im Programmspeicher (Flash ROM) gespeichert werden. Die Module sind in der Lage, nach dem Einschalten ohne Einwirkung von außen eine feste Anzeige darzustellen. Weitere Anzeigen müssen über die serielle Kommunikationsschnittstelle als Pixeldaten zugeführt oder aus der Menge der fest gespeicherten Pixelmuster ausgewählt werden. Eine einmal gespeicherte oder ausgewählte Anzeige wird selbsttätig angezeigt (autonomer LED Refresh).

Bildpuffer
Jedes Modul hat wenigstens zwei Bildpuffer (FRAME BUFFERS), die binäre Pixeldaten aufnehmen. Aus jeweils einen wird die LED-Anzeige zyklisch erneuert (LED Refresh). Der andere kann währenddessen über die serielle Kommunikationsschnittstelle mit neuen Inhalten gefüllt werden. Ist das Füllen abgeschlossen, wird umgeschaltet (Wechselpufferprinzip). Die jeweils aktuelle Anzeige wird durch die Füll- und Aktualisierungsvorgänge nicht beeinflußt.

Kommandosteuerung
Die Module werden mit Kommandos gesteuert, die über die serielle Kommunikationsschnittstelle geliefert werden.

Kommandoformate
Es gibt zwei grundsätzliche Alternativen:

1. Die 8-Bit-Übertragung. Diese Variante wird vorzugsweise dann genutzt, wenn die Großanzeige von einem PC gesteuert wird. Alle Zeichen müssen ASCII-Zeichen sein. Binäre Daten werden als HEX-Zeichen übertragen. Folgen von Text- und Datenzeichen werden mit einem Zeichen CR abgeschlossen. Damit die Steuerabläufe in den Modulen nicht zu kompliziert werden, sind alle Kommandos, die die Zustände der Kommandoübertragung selbst betreffen (Betriebsartenumschaltung, Konfigurieren, Modulauswahl) mit Sonderzeichen codiert. Dabei ist berücksichtigt worden, daß es keine Konflikte mit den Konventionen der Mehrprozessorkommunikation gibt.

2. Die 9-Bit-Übertragung. Diese Variante kommt nur dann in Betracht, wenn die Großanzeige an ein Kommando- oder Gruppensteuergerät angeschlossen wird. Kommandozeichen haben das 9. Bit gesetzt, Datenzeichen nicht. Somit können binäre Daten direkt übertragen werden[4].

Die folgende Darstellung betrifft ausschließlich die 8-Bit-Übertragung.

4: Doppelte Datenrate gegenüber der 8-Bit-Übertragung. Zudem können nicht-standardgemäße höhere Baudraten gewählt werden.

Nur diese Zeichen werden ausgewertet (reservierte Steuerzeichen):

Zeichen	Code	Wirkung im Modul
CR	0DH	Allgemeines Abschlußzeichen. Pufferinhalt auswerten, danach Puffer rücksetzen.
CAN	18H	Module rücksetzen (Puffer und Auswahlzustände in allen Modulen).
EOT	04H	Nicht unterstützt.
ENQ	05H	Nicht unterstützt.
ETB	17H	Auf Busbetrieb schalten. Puffer und Zustände in allen Modulen rücksetzen.
EM	19H	Auf Daisy Chain schalten. Puffer und Zustände in allen Modulen rücksetzen.
BEL	07H	Startzeichen für Allgemeinauswahl (Broadcast).
DC1	11H	Konfiguration starten.
DC2	12H	Modulauswahl starten.
ACK	06H	Nicht unterstützt.
NAK	15H	Nicht unterstützt.

Kommandoübersicht:

Kommando	Codierung	Bemerkungen
Auf Busbetrieb schalten	ETB	Löscht die Fehlerregister*
Auf Daisy Chain schalten	EM	Löscht die Fehlerregister*
Konfiguration	DC1 – aa – CR	aa = Adresse (2 HEX)
Modulauswahl	DC2 – aa – CR	aa = Adresse (2 HEX)
Allgemeinzugriff (Broadcast)	BEL	
Module rücksetzen	CAN	Löscht anhängige Auswahlzustände und Fehlerregister*
Bildinhalt eintragen	X – 1 – n – CR – Bilddaten – CR	100 Bytes Bilddaten in HEX für 50 Bytes Bildpuffer (binär). n = Nummer des Bildpuffers**
Festbild eintragen	X – 2 – bb – n – CR	bb = Bildnummer (2 HEX). n = Nummer des Bildpuffers**
Bildpuffer umschalten	X – 3 – n – CR	n = Nummer des Bildpuffers**
Potentiometer Absolutwert	X – 4 – p – xy – CR	p = Potentiometernummer (1 HEX). 1 = oberes, 2 = unteres Potentiometer xy = Zählwert (2 HEX).
Potentiometer +1	X – 5 – p – CR	
Potentiometer – 1	X – 6 – p – CR	
Potentiometer speichern	X – 7 – CR	Betrifft beide Potentiometer. Prinzip: zuerst – 1 zählen, dann +1 zählen und speichern

*: ERROR und EVENTS_CHK. **: Wird in der Variante mit nur zwei Bildpuffern (Wechselpufferprinzip) zwar erwartet, aber ignoriert.

Der Mikrocontroller
An den Mikrocontroller sind angeschlossen:

- die LED-Treiberschaltungen des Anzeigemoduls,
- zwei Digitalpotentiometer (Kontrasteinstellung),
- die Kommunikationsleitungen,
- die Kommunikationswegumschaltung,
- eine programmseitig schaltbare Anzeige-LED,
- ein frei verfügbares Signal, das als meßpunkt genutzt oder über einen Jumper (JP2) auf Masse gezogen werden kann,
- die Programmierleitungen,
- eine Rücksetztaste.

Taktfrequenz: 16 MHz (externer Taktgenerator).

Die Rücksetztaste
Sie wirkt auf die gemeinsame Rücksetzleitung (RESET#) der Programmierkabel. Das Betätigen einer Taste setzt alle angeschlossenen Module zurück.

Die E-A-Ports

Port B

7	6	5	4	3	2	1	0
res.	CLK***	SCK	MISO	MOSI	JP2	INC#	LED1
IN*	–	IN*	IN*	IN*	IN/OUT**	OUT	OUT

*: Mit Pull-up. **: Anfänglich IN mit Pull-up.***: Der Mikrocontroller ist auf externen Takt programmiert. Port B6 = Takteingang.

Port C

7	6	5	4	3	2	1	0
–	RESET#	CONNECT	CS_2#	CS_1#	UD	L_ROW_ DATA	U_ROW_ DATA
–	–	OUT	OUT	OUT	OUT	OUT	OUT

RSTDISBL Fuse nicht programmieren. Port C6 = Rücksetzeingang.

Port D

7	6	5	4	3	2	1	0
ROW_ CLK	COLUMN _5#	COLUMN_ 4#	COLUMN _3#	COLUMN _2#	COLUMN _1#	TX	RX
OUT	OUT	OUT	OUT	OUT	OUT	OUT	IN*

*: Mit Pull-up.

Programmiersignale:
SCK, MISO, MOSI, RESET#.

Kommunikationssignale:
RX (empfangene Daten), TX (Sendedaten), COMMECT (Signalwegumschaltung).

Signale der LED-Ansteuerung:
COLUMN_1# bis COLUMN_5# (Ansteuerung der Spaltentreiber (P-Kanal-FETs), L_ROW_DATA, U_ROW_DATA (Datensignale für die LED-Treiber 5451), ROW_CLOCK (gemeinsamer Takt für beide LED-Treiber 5451).

Potentiometersignale:
CS_1#, CS_2# (Erlaubnissignale, Potentiometerauswahl), UD (Richtungssteuerung Aufwärts/Abwärts), INC# (Zählimpuls).

Signale zur internen Anzeige, zum Debugging usw.:
LED1 (interne Anzeige), JP2. Das ist ein frei programmierbares Signal, das u. a. zu Konfigurationszwecken (Abfrage; ein entsprechender Jumper ist vorgesehen) oder als Meßpunkt (Ausgabe) verwendet werden kann, beispielsweise zum Debugging oder zur Leistungsmessung.

Kommunikationswegumschaltung:
Ein Treiberschaltkreis 74x126 dient zugleich zum Treiben und zum Umschalten der abgehenden Kommunikationsleitung:

- Daisy-Chain-Betrieb. Signal CONNCETD auf Low-Pegel. Die abgehende Kommunikationsleitung wird vom Sendesgnal TX des Mikrocntrollers angesteuert. Der Mikrocontroller sendet.
- Busbetrieb. Signal CONNCETD auf High-Pegel. Die abgehende Kommunikationsleitung wird von der ankommenden (RX_IN) angesteuert. Das ankommende Signal wird zum nächsten Modul weitergeleitet (durchgehender Signalweg).

3. Programmorganisation

Der Grundzustand
Im Grundzustand wird eine Endlosschleife ausgeführt, die die LED-Anzeige zyklisch aufrecht erhält (LED Refresh). Nach jedem vollständigen Auffrischzyklus wird abgefragt, ob Ereignisse anstehen. Anstehende Ereignisse werden behandelt.

Der Unterbrechungszustand
Jedes Zeichen, das über die serielle Schnittstelle (Eingang RX) ankommt, versetzt den Mikrocontroller in den Unterbrechungszustand. In diesem Zustand wird das jeweiligen Zeichen analysiert und ggf. (Daisy-Chain-Betrieb) weitergegeben. Funktionen, die nur vergleichsweise wenige Maschinenbefehle erfordern, werden sofort ausgeführt. Alle anderen Funktionen werden im Grundzustand erledigt. Die Funktionsausführung wird über Ereignisbits angefordert. Parameter und Daten werden in eigens zugeordneten Speicherbereichen übergeben.

Grundsatz: Der Mikrocontroller muß den Unterbrechungszustand verlassen haben, bevor das nächste Zeichen über die serielle Schnittstelle eingetroffen ist, denn sonst ergibt sich ein Überlauffehler (Overrun). Im Unterbrechungszustand kann somit nur erledigt werden, was nicht allzu lange dauert.

Anhaltswert: Bei 115 000 Baud dauert die Übertragung eines Zeichens (10 Bits) ca. 87 µs. 80 µs entsprechen bei 16 MHz 1280 Taktzyklen. Also dürfen im Unterbrechungszustand (von der Auslösung bis zur Rückkehr) niemals mehr als höchstens 1000 Maschinenbefehle ausgeführt werden[5].

Registerübersicht
Register enthalten Anzeige- und Merkbits. Alle Register und Variablen (Daten, Zählwerte usw.) der Anwendungssoftware werden im SRAM gehalten. Die architekturseitigen Register des Atmel AVR dienen nur als Arbeitsregister.

Die Anwendungssoftware verwendet die folgenden Register:

- das Zustandsregister (STATE),
- das Ereignismerkregister (EVENTS),
- das Fehlerregister (ERROR),
- das Ereignisfehlerregister (EVENTS_CHK)
- das Moduladreßregister (ADRS).

5: Näherungsbetrachtung mit Sicherheitszuschlägen für Befehle, die 2 oder 3 Taktzyklen erfordern.

STATE

7	6	5	4	3	2	1	0
XMIT_ PENDING	SELECT_ ACTION	CONFIG_ ACTION	PIXEL_ DATA	SELEC_ TED	BROAD_ CAST	CONFI_ GURED	BUS MODE

EVENTS

7	6	5	4	3	2	1	0
ERROR SHOW	–	BUFF SEL	SWITCH BUFFER	LOWER_ POT_SET	UPPER_ POT_SET	GENERATE_ FRAME	NEW_ FRAME

EVENTS_CHK

7	6	5	4	3	2	1	0
–	–	–	SWITCH BUFFER	LOWER_ POT_SET	UPPER_ POT_SET	GENERATE_ FRAME	NEW_ FRAME

ADRS

7	6	5	4	3	2	1	0
MOULE ADRS (0...255)							

ERROR

7	6	5	4	3	2	1	0
SPURIOUS ITRP	ERROR CODE			SELECT DATA CHECK	CONFIG DATA CHECK	FRAMING ERROR	OVERRUN

Pufferbereiche:

Kommandopuffer
Der Kommandpuffer nimmt die einlaufenden Kommandozeichen auf. Hierfür sind 10 Bytes reserviert.

Pixelpuffer
Der Pixelpuffer nimmt die einlaufenden Pixeldaten auf. Im 8-Bit-Betrieb sind dies hexadezimale Zeichen. Ein Modul benötigt 100 Zeichen.

Bildpuffer
Bildpuffer (FRAME BUFFERS) enthalten binär codierte Bilddaten. Ein Byte enthält das Bitmuster von 7 Pixeln bzw. von je einer Spalte einer LED-Punktmatrixanzeige. Jedes Bild erfordert 50 Bytes. Die minimale Ausstattung umfaßt zwei Bildpuffer, die nach dem Wechselpufferprinzip verwaltet werden. Der eine enthält das aktuell dargestellte Bitmuster, der andere kann mit neuen Bildinhalten geladen werden.

Speicherorganisation
Der SRAM ist folgendermaßen auftgeteilt:

ERROR: Fehlerregister
STATE: Zustandsregister
ADRS: Adreßregister
EVENTS: Ereignismerkregister
EVENTS_CHK: Ereignisfehlerregister
Füllstand Kommandopuffer
Füllstand Pixelpuffer
Testbildauswahl (Ereignis GENERATE_FRAME)
Potentiometerwert oben (Ereignis UPPER_POT_SET)
Potentiometerwert unten (Ereignis LOWER_POT_SET)
Kommandopuffer: 10 Bytes
Pixelpuffer: 100 Bytes
Bildpuffer 1: 50 Bytes
Bildpuffer 2: 50 Bytes
Rettungsbereich für Fehleranzeige: 14 Bytes
Reserve
Stack

Ereignisübersicht
Ereignisse treten im Unterbrechungszustand auf. Sie werden im Grundzustand behandelt. Für jedes Ereignis ist ein Bit im Ereignismerkregister (EVENTS) vorgesehen. Parameter werden in eigens zugeordneten SRAM-Zellen übergeben. Die Ereignismerkbits werden im Unterbrechungszustand gesetzt. Im Grundzustand werden sie abgefragt und – nachdem das jeweilige Ereignis behandelt wurde – wieder gelöscht

Ereignisse:

Ereignis	Wirkung	Einschaltzeitpunkt
NEW_FRAME	Aufbau eines Bildpufferinhaltes aus dem Pixelpuffer	CR am Ende einer Pixelübertragung.
GENERATE_FRAME	Erzeugen eines internen Testbildes. Die binäre Bildnummer steht im SRAM.	CR am Ende eines Kommandos X2
UPPER_POT_SET	Das obere Potentiometer auf einen Absolutwert einstellen. Der binäre Wert steht im SRAM.	CR am Ende eines Kommandos X4 mit entsprechender Potentiometerauswahl

Ereignis	Wirkung	Einschaltzeitpunkt
LOWER_POT_SET	Das untere Potentiometer auf einen Absolutwert einstellen. Der binäre Wert steht im SRAM.	CR am Ende eines Kommandos X4 mit entsprechender Potentiometerauswahl
SWITCH_ BUFFER	Den aktiven Bildpuffer zur Anzeige umschalten	Kommando X3

Ereignisfehler

Ein Ereignisfehler (EVENT CHECK) wird dann erkannt, wenn im Unterbrechungszustand ein Ereignis auftritt, die vorhergehende Anforderung aber noch nicht erledigt wurde. Das Ereignis tritt neu auf, das zugehörige Bit im Ereignismerkregister ist aber noch gesetzt (Überlaufbedingung). In einem solchen Fall wird das betreffende Bit im Ereignisfehlerregister (EVENTS_CHK) gesetzt. In das Fehlerregister (ERROR) wird der Fehlercode 5 eingetragen.

Hinweis: Die Ereignisfehlererkennung ist vor allem ein Hilfsmittel zum Inbetriebnehmen von Anwendungssoftware der Ansteuerung (die in einem PC oder einem weiteren Mikrocontroller (Kommandogerät) läuft). Die steuernde Software muß den Mikrocontrollern der Module genügend Zeit lassen, die Ereignisse zu behandeln. Da es keine gegenseitige Verriegelung (Interlocked Handshaking) gibt, gelingt dies nur, indem ggf. ausreichende Verzögerungszeiten vorgesehen werden.

4. Kommandobeschreibung

Auf Busbetrieb schalten: ETB
Das erste Modul gibt ETB weiter und schaltet dann um. Das zweite Modul gibt ETB weiter, schaltet dann um usw. Das letzte Modul gibt ETB aus (Rücksendung)[6] und schaltet dann um. Vor dem Umschalten wird jeweils die komplette Übertragung abgewartet. Ein Modul schaltet erst dann um, nachdem das letzte Bit die serielle Schnittstelle verlassen hat.In allen Modulen werden das Fehlerregister, das Ereignisfehlerregister und anhängige Zustände gelöscht; die Puffer werden zurückgesetzt (Module rücksetzen; s. CAN). Befindet sich das Modul schon im Busbetrieb, wird nichts weiter getan.

Hinweise:

1. Umschalten heißt: das CONNECT-Signal schaltet auf High. Dadurch wird der Sendetreiber des Moduls inaktiv, und es ergibt sich ein durchgehender Signalweg.

2. Den Busbetrieb erst dann aufnehmen, nachdem alle Module umgeschaltet haben (die Rücksendung vom letzten Modul abwarten oder entsprechend Zeit lassen).

Auf Daisy Chain schalten: EM
Im Busbetrieb wird das Kommando von allen Modulen nahezu gleichzeitig ausgeführt. In allen Modulen werden das Fehlerregister, das Ereignisfehlerregister und anhängige Zustände gelöscht; die Puffer werden zurückgesetzt (Module rücksetzen; s. CAN). Befindet sich das Modul schon im Daisy-Chain-Betrieb, wird das Kommando weitergegeben, aber sonst nichts weiter getan.

Hinweise:

1. Umschalten heißt: das CONNECT-Signal schaltet auf Low. Dadurch wird der Sendetreiber des Moduls aktiv. Das ankommende Signal wird empfangen, das abgehende vom Modul getrieben (Weitergabe).

2. Die Ausführungszeiten in den einzelnen Modulen hängen vom jeweiligen Verarbeitungszustand ab (Interruptlatenzzeiten). Deshalb einige ms warten, bevor der Betrieb im Daisy-Chain-Modus aufgenommen wird.

Konfiguration: DC1 – Adresse – CR
DC1 setzt den Konfigurationszustand (CONFIG_ACTION). Die Adresse besteht aus zwei HEX-Zeichen. Sie gelangen in den Kommandopuffer. Abschluß mit CR.

Im Daisy Chain wird nach CR die empfangene Adresse zur eigenen Adresse. Das gesamte Kommando wird mit einer um 1 erhöhten Adresse weitergegeben. Die Kommandoausführung ist dann beendet, wenn die gesamte Weitergabe erfolgt ist. Der Konfigurationszustand wird

6: Es ist eine Frage der Systemkonfiguration, ob die Rücksendung ausgewertet wird (z. B. zu Prüfzwecken) oder nicht.

verlassen. Das Modul befindet sich nunmehr im Zustand “konfiguriert” (CONFIG_ACTION aus, CONFIGURED ein). Das Statusregister enthält 02H (nur CONFIGURED, sonst nichts[7]).

Befindet sich die Einrichtung im Busbetrieb, hat das Kommando nur die Wirkung, das Stausregister auf 02H zu setzen. Die Moduladresse wird nicht verändert. Sonst hat das Kommando keine Wirkung.

Hinweis: Das Konfigurieren pflanzt sich durch die gesamte Kette der Module fort. Nach dem Auslösen eines Konfigurationskommandos genügend Zeit lassen.

Modulauswahl: DC2 – Adresse – CR
DC2 setzt den Auswahlzustand (SELECT_ACTION). Ggf. anhängige andere Auswahlzustände (SELECTED, BROADCAST) werden gelöscht. Die Adresse besteht aus zwei HEX-Zeichen. Sie gelangen in den Kommandopuffer. Abschluß mit CR. Der Auswahlzustand wird beendet (SELECT_ACTION aus). Dann wird die übertragene Adresse mit der konfigurierten Moduladresse verglichen. Bei Übereinstimmung erkennt sich das Modul als ausgewählt (SELECTED = 1).

Allgemeinauswahl (Broadcast): BEL
Das Kommando bewirkt, daß alle Module gleichzeitig ausgewählt werden (Auswahlzustand BROADCAST). Der Modulauswahlzustand (SELECTED) bleibt dabei erhalten. Die Kommandos CAN, DC1 und DC2 bewirken, daß der BROADCAST-Zustand verlassen wird.

Module rücksetzen: CAN
Das Kommando wirkt in allen Modulen. Es beendet anhängige Konfigurations- und Auswahlzustände (CONFIG_ACTION, SELECT_ACTION, SELECTED, BROADCAST, PIXEL_DATA). Fehlerregister und Ereignisfehlerregister werden gelöscht. Kommandopuffer und Pixelpuffer werden zurückgesetzt. Die Pufferinhalte bleiben dabei erhalten.

Bildinhalt eintragen: X – 1 – n – CR – Bilddaten – CR
n (1 Byte) dient zur Bildpufferauswahl. Das Byte wird in der Version 1.x zwar erwartet, aber ignoriert. Das Kommando X1 überführt ausgewählte Module (SELECTED oder BROADCAST) in den Zustand der Pixeldatenübertragung (PIXEL_DATA). Die an CR anschließenden hexadezimalen Daten gelangen in den Pixelpuffer. Es sind maximal 100 Bytes zulässig. Das abschließende CR bewirkt, daß der Zustand der Pixeldatenübertragung verlassen wird (PIXEL_DATA aus), und es aktiviert das Ereignis NEW_FRAME, das im Grundzustand behandelt wird. Die Behandlung besteht darin, die übertragenen Bytes in binäre Bilddaten umzuwandeln und im jeweils zugeordneten Bildpuffer (FRAME BUFFER) zu speichern. Danach wird der Pixelpuffer zurückgesetzt, so daß ein neuer Bildinhalt übertragen werden kann.

Hinweis: Werden neue Bilddaten zu zeitig übertragen, kann es einen Pixelpufferüberlauf oder einen Ereignisfehler geben (ERROR-Codes 4 oder 5).

7: Ggf. anhängige andere Auswahlzustände (SELECTED, BROADCAST) werden hiermit gelöscht.

Festbild eintragen: X – 2 – Bildnummer – n – CR
Die Bildnummer besteht aus zwei HEX-Zeichen. n (1 Byte) dient zur Bildpufferauswahl. Das Byte wird in der Version 1.x zwar erwartet, aber ignoriert. Ist das jeweilige Modul ausgewählt (SELECTED oder BROADCAST), so aktiviert das abschließende CR das Ereignis GENERATE_FRAME, das im Grundzustand behandelt wird. Die Behandlung besteht darin, das ausgewählte Bild im jeweils zugeordneten Bildpuffer (FRAME BUFFER) zu erzeugen.

In der Version 1.1 werden folgende Anzeigen unterstützt:

Bildnummer	Anzeige
2	Alle LEDs aus
3	Alle LEDs ein
4	Registeranzeige (REG DUMP)
5	Der Text “TEST”
6	Verschachtelte Rechtecke (SQUARES)

Alle anderen Nummern werden ignoriert (keine Wirkung).

Bildpuffer umschalten: X – 3 – n – CR
n (1 Byte) dient zur Bildpufferauswahl. Das Byte wird in der Version 1.x zwar erwartet, aber ignoriert. Ist das jeweilige Modul ausgewählt (SELECTED oder BROADCAST), so aktiviert das abschließende CR das Ereignis SWITCH_BUFFER, das im Grundzustand behandelt wird. Die Behandlung besteht darin, die Bilddarstellung (LED Refresh) auf den ausgewählrten Bildpuffer umzuschalten.

Hinweis: In der Version 1.x wird auf den jeweils anderen Bildpuffer umgeschaltet (Wechselpufferprinzip).

Potentiometer Absolutwert: X – 4 – p – xy – CR
p, x, y sind HEX-Zeichen. p = Potentiometernummer (1 = oben, 2 = unten). XY = Zählwert. Der größte sinnvolle Wert ist 99. Bei größeren Werten läuft das Potentiometer gegen den oberen Anschlag. Ist das jeweilige Modul ausgewählt (SELECTED oder BROADCAST), so aktiviert das abschließende CR das entsprechende Ereignis (UPPER_POT_SET oder LOWER_POT_SET), das im Grundzustand behandelt wird. Die Behandlung besteht darin, das Potentiometer mit 99 Impulsen zum unteren Anschlag zu bringen und dann auf den übergebenen Zählwert hochzuzählen.

Potentiometer +1: X – 5 – p – CR
p ist die Potentiometernummer als HEX-Zeichen (1 = oben, 2 = unten). Ist das jeweilige Modul ausgewählt (SELECTED oder BROADCAST), so bewirkt das abschließende CR, daß die Potentiometerstellung um 1 erhöht wird (Aufwärtszählung).

Potentiometer –1: X – 5 – p – CR
p ist die Potentiometernummer als HEX-Zeichen (1 = oben, 2 = unten). Ist das jeweilige Modul ausgewählt (SELECTED oder BROADCAST), so bewirkt das abschließende CR, daß die Potentiometerstellung um 1 vermindert wird (Abwärtszählung).

Potentiometer speichern: X – 7 – CR
Ist das jeweilige Modul ausgewählt (SELECTED oder BROADCAST), so bewirkt das abschließende CR, daß die Potentiometerstellung zunächst um 1 vermindert und dann um 1 erhöht und dabei gespeichert wird. Das Kommando gilt für die beiden Potentiometer des Moduls.

Potentiometerkommandos im Überblick:

Funktion	Codierung	Pufferfüllstand
Absolutwert einstellen	X – 4 – p – xy – CR	5
Um 1 erhöhen	X – 5 – p – CR	3
Um 1 vermindern	X – 6 – p – CR	3
Speichern (beide)	X – 7 – CR	2

- p, x, y sind HEX-Zeichen.
- p = Potentiometernummer in HEX (1 = oben, 2 = unten).
- xy = Wert in zwei HEX-Zeichen. Sinnvolles Maximum = 99. Beim Zählen um 99 läuft das Potentiometer beim Erhöhen auf den Endwert (99), beim Vermindern auf den Anfangswert (0). (Dieses Gegen-den-Anschlag-Laufen ist eine Funktion des Potentiometers.)

Grundsatzverhalten der Datenübertragung (Datenzeichen, keine Steuerzeichen):

- Eintragen in den Kommandopuffer, wenn nicht PIXEL_DATA, aber CONFIG_ACTION oder SELECT_ACTION oder SELECTED oder BROADCAST.
- Eintragen in den Pixelpuffer, wenn PIXEL_DATA.

Grundsatzverhalten CR:

- PIXEL_DATA: Übertragungsende. Ereignis NEW_FRAME ein. Der Ereignisbehandler setzt den Pixelpuffer zurück, wenn er fertig ist.
- CONFIG_ACTION: Konfigurationsablauf. Ausschalten. Kommandopuffer rücksetzen.
- SELECT_ACTION: Auswahlablauf (Adreßvergleich). Demgemäß SELECED ein- oder ausschalten. Kommandopuffer rücksetzen.
- SELECTED (und NICHT PIXEL_DATA): Kommandoauswertung. Kommandopuffer rücksetzen.
- BROADCAST (und NICHT PIXEL_DATA): Kommandoauswertung. Kommandopuffer rücksetzen.

Grundsatzverhalten CAN:

- Kommandopuffer und Pixelpuffer rücksetzen (jeweils nur Zeiger auf 0; Inhalt bleibt).
- Die Auswahlzustände CONFIG_ACTION, SELECT_ACTION und BROADCAST rücksetzen. Die Modulauswahl (SELECTED) bleibt erhalten.

Grundsatzverhalten Allgemeinauswahl (BROADCAST):
Alle Module sind gleichermaßen ausgewählt und führen so die nachfolgenden Kommandos parallel aus (bei Daisy Chain mit einem Zeitversatz infolge der Weitergabe). Dieser Auswahlzustand wird zurückgesetzt durch folgende Kommandos:

- Auf Busbetrieb schalten (ETB)
- Auf Daisy Chain schalten (ET)
- Konfiguration (DC1)
- Modulauswahl (DC2)
- Puffer rücksetzen (CAN)

Grundsatzverhalten der Modulauswahl:
Das Kommando DC2 bewirkt eine Modulauswahl über einen Adreßvergleich. Ist das Modul nicht konfiguriert, beibt das Kommando wirkungslos. Ein bisher bestehender Auswahlzusand (SELECTED) wird anfänglich gelöscht. Entspricht die Auswahladresse keiner der konfigurierten Moduladressen, wird kein Modul ausgewählt. Der Auswahlzustand (SELECTED) wird zurückgesetzt durch folgende Kommandos:

- Auf Busbetrieb schalten (ETB)
- Auf Daisy Chain schalten (ET)
- Konfiguration (DC1)
- Modulauswahl (DC2), wenn nicht konfiguriert oder Adressen ungleich.

Grundsatzverhalten der Weitergabe (Daisy Chain):
Weitergeben (Propagate) heißt, daß ein Modul Zeichen an das nächste Modul sendet (TX nach RX). Das kann es nur im Daisy-Chain-Betrieb geben.

Die folgenden Steuerzeichen (Kommandocodes) werden immer weitergegeben:

- CAN (Puffer rücksetzen)
- EM (auf Daisy Chain schalten[8])
- BEL (Allgemeinauswahl)
- DC2 (Modulauswahl)

Die Weitergabe im Kommando DC1 (Konfiguration):
Das Modul gibt die empfangenen Zeichen nicht sofort weiter, sondern führt das Konfigurationskommando zunächst selbst aus. Est dann wird das gesamte Kommando mit einer um 1 erhöhten Moduladresse weitergegeben.

Die Weitergabe im Kommando ETB (auf Busbetrieb schalten):
Wenn sich das Modul im Daisy-Chain-Betrieb befindet, wird das Kommandozeichen sofort weitergegeben. Dann wird die Zeichenübertragung abgewartet. Nachdem das Zeichen die USART des Moduls verlassen hat, wird die Betriebsart umgeschaltet.

Weitergeben von Textzeichen und CR:
Die Weitergabe entfällt, wenn die nachfolgenden Module mit den Zeichen nichts anfangen können (Zeitersparnis)

Textzeichen und CR werden weitergegeben:

8: Wenn bereits im Daisy-Chain-Betrieb.

- wenn das Modul nicht konfiguriert ist (CONFIGURED aus),
- wenn die Allgemeinauswahl aktiv ist (BROADCAST),
- wenn das Modul nicht ausgewählt ist (SELECTED aus),
- wenn ein Modulauswahlkommando ausgeführt wird (SELECT_ACTION).

Textzeichen und CR werden nicht weitergegeben:

- wenn das Modul ein Konfigurationskommando ausführt (CONFIG_ACTION, ausgelöst durch DC1),
- wenn das Modul ausgewählt ist (SELECTED), und die vorgenannten anderen Zustände nicht vorliegen. Regel: ein ausgewähltes Modul wertet die empfangenen Zeichen selbst aus und gibt sie nicht weiter.

Das Konfigurieren
Ein Modul ist konfiguriert (CONFIGURED), wenn es eine Moduladresse zugewiesen bekommen hat. Hierzu muß im Daisy-Chain-Betrieb ein Kommando Konfiguration (DC1) ausgeführt werden. Nach dem Einschalten oder Hardware-Rücksetzen ist das Modul nicht konfiguriert. Hat es eine Moduladresse erhalten, so bleibt es konfiguriert. Die Betriebsart kann beliebig zwischen Daisy-Chain- und Busbetrieb umgeschaltet werden, ohne daß der Konfiguriert-Zustand und die Moduladresse verloren gehen. Im Daisy-Chain-Betrieb ist ein erneutes Konfigurieren jederzeit möglich.

Hinweise:

1. Das Konfigurieren kostet Zeit, weil sich das Kommando durch die gesamte Kette der Module fortpflanzen muß. Also nur dann konfigurieren, wenn unbedingt nötig.

2. Der Konfiguriert-Zustand (CONFIGURED) kann nur dann wieder ausgehen, wenn beim erneuten Konfigurieren ein Konfigurationsfehler (CONFIG DATA CHECK) erkannt wird.

Der Anfangszustand (Einschalten, Hardware-Rücksetzen)
Daisy Chain. Nicht konfiguriert. Alle Register und Puffer gelöscht. Das erste Kommando ist typischerweise ein Konfigurationskommando (DC1 mit der Adresse des ersten Moduls (z. B. 00H)).

Das Senden der Zeichen (der Sendezustand)
Zeichen werden nur im Unterbrechungszustand gesendet, nicht im Grundzustand. Wenn die Software ein Zeichen zum Senden ins Datenregister der USART geschrieben hat, versetzt sie das Modul in den Sendezustand (XMIT_PENDING). Befindet sich das Modul im Sendezustand, darf ein weiteres Zeichen erst dann gesendet werden, wenn das Datenregister der USART wieder frei ist. Das wird ggf. abgewartet (Abfrage des Hardwarebits TXC). Im Normalfall setzt die Software die Kommandoausführung fort, nachdem sie das Zeichen ins Datenregister der USART eingetragen und den Sendezustand eingeschaltet hat. Die eigentliche Datenübertragung überlappt sich dann mit der weiteren Programmausführung und dem Empfangen des nächsten Zeichens. Ausnahmen von diesem Ablauf sind lediglich das Kommando ETB (hier wird das Ende der Zeichenübertragung innerhalb der Kommandoausführung abgewartet) und das Kommando DC1 (hier werden alle Kommandozeichen innerhalb einer Kommandoausführung zusammenhängend gesendet).

5. Fehlererkennung und Fehlersignalisierung

Die Software wertet die Fehlersignale der USART aus und führt einige Plausibilitätsprüfungen durch. Fehlersignale werden im Fehlerregister (ERROR) gespeichert, Ereignisfehler ergänzend im Ereignisfehlerregister (EVENTS_CHK).

Fehleranzeige
Ist der Inhalt des Fehlerregisters nicht gleich Null, so gelangt das Modul in den Zustand der Fehleranzeige. Hierbei werden die Register auf den LEDs angezeigt (REG DUMP).

Ein Fehlerzustand verhindert nicht die weitere Kommandoausführung. Es hängt aber von der Art des Fehlers ab, ob sie erfolgreich ist oder nicht.

Die Registeranzeige auf den LEDs (REG DUMP):

Bit	ERROR	STATE	ADRS	EVENTS	EVENTS_CHK
0	OVERRUN	BUS MODE	MA0	NEW FRAME	NEW FRAME
1	FRAMING ERROR	CONFIGURED	MA1	GENERATE FRAME	GENERATE FRAME
2	CONFIG DATA CHECK	BROADCAST	MA2	UPPER POT SET	UPPER POT SET
3	SELECT DATA CHECK	SELECTED	MA3	LOWER POT SET	LOWER POT SET
4	ERROR CODE 0	PIXEL DATA	MA4	SWITCH BUFFER	SWITCH BUFFER
5	ERROR CODE 1	CONFIG_ACTION	MA5	BUFF SEL	
6	ERROR CODE 2	SELECT_ACTION	MA6		
7	SPURIOUS ITRP	XMIT_PENDING	MA7	ERROR_SHOW	
–					
–					
–					
–					
–					
–					

MA = Moduladresse.

Löschen des Fehlerregisters und des Ereignisfehlerregisters
Beide Register werden gelöscht mit den Kommandos ETB, EM und CAN, also bei jeder Betriebsartenumschaltung und beim Pufferrücksetzen.

Ist das Fehlerregister nach einer Fehleranzeige gelöscht worden, wird versucht, die ursprüngliche LED-Anzeige wieder herzustellen. In Abhängigkeit von der Fehlersituation kann es Fälle geben, in denen dies nicht vollkommen gelingt.

Die Fehlerbits FRAMING_ERROR und OVERRUN werden bei jedem Zeichenempfang aus der USART neu gestellt.

Grundsatzverhalten der internen Anzeige-LED
Die LED wird von der Software angesteuert, und zwar ausschließlich im Grundzustand. Es gibt die folgenden Anzeigen:

- Schnelles Blinken: Modul nicht konfiguriert (CONFIGURED aus). Eine halbe Blinkperiode = 100 Umläufe der Grundschleife (LED Refresh). Blinkfrequenz ca. 8 Hz.
- Langsameres Blinken: Die Software hat einen Fehler erkannt (ERROR-Register nicht Null). Die Register werden auf der LED-Anzeige dargestellt. Eine halbe Blinkperiode = 250 Umläufe der Grundschleife (LED Refresh). Blinkfrequenz ca. 3 Hz.
- Dauernd ein: Modul konfiguriert (CONFIGURED ein) und kein Fehler. Alles o.k.
- Dauernd aus: Hardwarefehler.

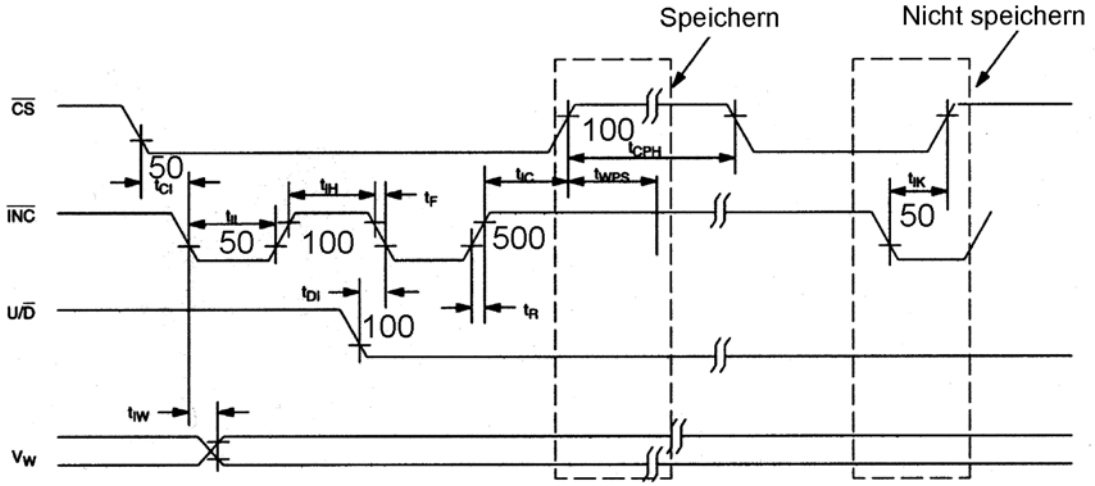
6. Die Digitalpotentiometer

Zur Kontrasteinstellung sind zwei Digitalpotentiometer Maxim DS1804 vorgesehen. Sie werden über die Signale CS#, U/D# und INC# angesteuert. Die Potentiometer haben 100 Widerstandsstufen. Sie können die aktuell ausgewählte Stufe speichern. Ein Abwärtszählen mit hinreichend vielen Impulsen fährt das Potentiometer an den unteren Anschlag (Stufe 0), ein Aufwärtszählen an den oberen (Stufe 99).

Schaltkreisausauwahl:
Der Mikrocontroller steuert zwei CS-Signale an. En Potentiometer ist ausgewählt, wenn CS# Low-Pegel führt.

Richtungssteuerung:
U/D# auf Low = Abwärtszählung, U/D# auf High = Aufwärtszählung.

Zählen:
Mit Zählimpulsen am Anschluß INC#.



(Bildquelle: Datenblatt DS1804 (Maxim))

Anforderungen an das Zeitverhalten:

- Zählimpulse (INC#): 50 ns aktiv, 100 ns Pause.
- CS# muß 50 ns vor der ersten Flanke von INC# aktiv sein.
- U/D# muß 100 ns vor der ersten Flanke von INC# eingestellt sein.
- Nicht speichern: CS# muß bei aktiven INC# inaktiv werden, und zwar frühestens 50 ns nach der Vorderflanke.n
- Speichern: CS# muß nach der letzten Rückflanke von INC# noch 500 ns aktiv bleiben.

7. Grundlagen der Bildspeicherorganisation

Ein Modul stellt ein graphisches Anzeigefeld dar, das 350 Pixel enthält, die in 14 Zeilen zu 25 Pixeln angeordnet sind. Wertebereich in X-Richtung: 0...24, in Y-Richtung 0...13. Der Koordinatenursprung (0, 0) liegt links oben (vgl. Windows).

Die Pixel werden binär gespeichert (1 Pixel = 1 Bit). Der Bildspeicher ist byteweise organisiert. Ein Byte enthält 7 Pixel. Es werden nur die Bits 6...0 genutzt. Die 25 Spalten erfordern einen Bildspeicher von 50 Bytes. Die Bytes folgen so aufeinander, wie dies zum Aufbauen der LED-Anzeige erforderlich ist. Es genügt eine +1-Adreßzählung, um die Bytes auszulesen. Der Datenstrom muß in dieser Form angeliefert werden. Die Umsetzung eines XY-Bildpuffers (als zweidimensionales Array) in einen solchen Datenstrom muß in der ansteuernden Einrichtung erfolgen (PC, Kommandogerät o. dergl.). Im 8-Bit-Betrieb wird jedes Byte mit zwei Hexadezimalzeichen übertragen. Die höherwertige Tetrade (Bits 7...4) kommt zuerst.

Die LED-Anzeige besteht aus Punktmatrixanzeigen (Dot Matrix Displays) der Organsiationsform 5 • 7. Es sind zwei Reihen zu je 5 Anzeigen vorgesehen, eine obere (Upper) und eine untere (Lower). Jede Reihe ist an einen LED-Treiberschaltkreis MM5451 angeschlossen.

Jede LED-Spalte entspricht zwei Bytes. Jedes Byte nimmt die 7 Pixel einer Spalte einer Punktmatrixanzeige auf. Die Spaltenbytes werden so hintereinander gespeichert, wie sie zu den LED-Treiberschaltkreisen ausgegeben werden, erst das Byte für die oberen LEDs, dann das für die unteren. Bit 0 ist immer jeweils ganz oben.

X-Koordinaten* und Byte-Offsets im Bildpuffer:

X-Koordinate	Byteoffset	X-Koordinate	Byteoffset
0	0	13	34
1	10	14	44
2	20	15	6
3	30	16	16
4	40	17	26
5	2	18	36
6	12	19	46
7	22	20	8
8	32	21	18
9	42	22	28
10	4	23	38
11	14	24	48
12	24		

*: Für jede X-Koordinate werden 2 Bytes gespeichert, um die14 Pixel in Y-Richtung unterzubringen.

Y-Koordinaten und Bytes:

Bytes	2.								1.							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Y-Koordinaten		13	12	11	10	9	8	7		6	5	4	3	2	1	0
HEX-Zeichen	3.				4.				1.				2.			

Bytes im Bildpuffer:

	Y-Koordinaten, ungerade Byte-Offsets*								**Y-Koordinaten, gerade Byte-Offsets***							
	–	13	12	11	10	9	8	7	–	6	5	4	3	2	1	0
X = 0	1								0							
X = 5	3								2							
X =10	5								4							
X = 15	7								6							
X = 20	9								8							
X = 1	11								10							
X = 6	13								12							
X = 11	15								14							
X = 16	17								16							
X = 21	19								18							
X = 2	21								20							
X = 7	23								22							
X = 12	25								24							
X = 17	27								26							
X = 22	29								28							
X = 3	31								30							
X = 8	33								32							
X = 13	35								34							
X = 18	37								38							
X = 23	39								40							
X = 4	41								42							
X = 9	43								44							
X = 14	45								46							
X = 19	47								48							
X = 24	49								48							

*: Bezogen auf die Anfangsadresse des Bildpuffers

Der hexadezimale Datenstrom (1. bis 100. Byte):

	Y-Koordinaten, ungerade Byte-Offsets*								**Y-Koordinaten, gerade Byte-Offsets***							
	–	13	12	11	10	9	8	7	–	6	5	4	3	2	1	0
X = 0	3.				4.				1.				2.			
X = 5	7.				8.				5.				6.			
X =10	11.				12.				9.				10.			
X = 15	15.				16.				13.				14.			
X = 20	19.				20.				17.				18.			
X = 1	23.				24.				21.				22.			
X = 6	27.				28.				25.				26.			
X = 11	31.				32.				29.				30.			
X = 16	35.				36.				33.				34.			
X = 21	39.				40.				37.				38.			
X = 2	43.				44.				41.				42.			
X = 7	47.				48.				45.				46.			
X = 12	51.				52.				49.				50.			
X = 17	55.				56.				53.				54.			
X = 22	59.				60.				57.				58.			
X = 3	63.				64.				61.				62.			
X = 8	67.				68.				65.				66.			
X = 13	71.				72.				69.				70.			
X = 18	75.				76.				73.				74.			
X = 23	79.				80.				77.				78.			
X = 4	83.				84.				81.				82.			
X = 9	87.				88.				85.				86.			
X = 14	91.				92.				89.				90.			
X = 19	95.				96.				93.				94.			
X = 24	99.				100.				97.				98.			

*: Bezogen auf die Anfangsadresse des Bildpuffers

Anhang

Die ASCII Steuerzeichen:

Dez.	HEX	Zeichen	Bedeutung	Dez.	HEX	Zeichen	Bedeutung
0	0	NUL	Null	16	10	DLE	Data Link Escape
1	1	SOH	Start of Heading	17	11	DC1	Device Control 1
2	2	STX	Start of Text	18	12	DC2	Device Control 2
3	3	ETX	End of Text	19	13	DC3	Device Control 3
4	4	EOT	Endo of Transmission	20	14	DC4	Device Control 4
5	5	ENQ	Enquiry	21	15	NAK	Negative Acknowledge
6	6	ACK	Acknowledge	22	16	SYN	Synchronous Idle
7	7	BEL	Bell (Klingel)	23	17	ETB	End of Transmission Block
8	8	BS	Backspace (Rückschritt)	24	18	CAN	Cancel (ungültig machen)
9	9	TAB	Horizontal Tabulation	25	19	EM	End of Medium
10	A	LF	Line Feed (Zeilenvorschub)	26	1A	SUB	Substitute Character
11	B	VT	Vertical Tabulation	27	1B	ESC	Escape (Umschaltung)
12	C	FF	Form Feed (New Page; neue Seite)	28	1C	FS	File Spearator
13	D	CR	Carriage Return (Zeilenrücklauf)	29	1D	GS	Group Separator
14	E	SO	Shift Out (Dauerumschaltung)	30	1E	RS	Record Separator
15	F	SI	Shift In (Rückschaltung)	31	1F	US	Unit Separator
				127	7F	DEL	Delete