



Application Note

Version: 1.1
Datum: 25.11.2016
Autor: Achim Döbler



Inhaltsverzeichnis

1	Einleitung	2
2	Display	2
2.1	Pinbelegung	3
2.2	Schnittstelle zur Bildübertragung	4
2.2.1	Grundlagen	4
2.2.2	Timings	10
2.3	Schnittstelle zur Konfiguration	10
2.4	LED-Backlight	12
3	Touch	13
3.1	I ² C	13
4	Entwicklungsumgebung	16
4.1	Inbetriebnahme	16
4.1.1	Anschließen des Displays	16
4.1.2	Konfiguration der Displaygröße	17
4.1.3	Konfiguration von Touchscreens	18
4.1.4	Spannungsversorgung	18
4.2	Hardware	19
4.2.1	STM32-Mikrocontroller	19
4.2.2	Schaltplan	21
4.2.3	Pinbelegung der Stiftleisten	22
4.3	Software	24

1 Einleitung

In der Folge soll das von der ACTRON AG entwickelte **ACT I³**-Konzept für TFT-LCDs erläutert werden¹. Eine wesentliche Eigenschaft von **ACT I³** ist die einheitliche 50-polige Pinbelegung sowie Spannungsversorgung. Auf diese Weise können alle **ACT I³**-fähigen Displays direkt untereinander ausgetauscht werden, ohne dass eine Anpassung der Ansteuerhardware nötig wäre. Ein weiterer Punkt ist die gezielte EMV-Optimierung: Zur Minimierung der elektromagnetischen Störabstrahlung sind die Anschlussleitungen bei **ACT I³** vollständig mit mehreren niederohmigen und auf Masse liegenden Schichten geschirmt. Diese Maßnahme reduziert nicht nur die abgestrahlte Energie, sondern verringert auch die Störanfälligkeit der Displays maßgeblich. Um eine möglichst niederohmige Masseanbindung der Displays zu gewährleisten, besitzt **ACT I³** insgesamt 12 Masseleitungen. Sämtliche, für das TFT-LCD benötigte, Hilfsspannungen werden direkt auf dessen Flex-Tail erzeugt. Ein ebenfalls dort platzierter Backlight-Treiber stellt stets die korrekte Konstantstromversorgung der LEDs sicher. Die Realisierung einer Konstantstromquelle auf Applikationsseite entfällt daher vollständig. Um insbesondere den heutigen Industrieanforderungen gerecht zu werden, bieten alle Displays dieser Serie eine typische Leuchtdichte von 1000 cd/m² bei einer Lebensdauer von min. 50.000 Stunden. **ACT I³** besitzt daher wesentliche Vorteile gegenüber einer herkömmlichen Lösung:

- ✓ Einheitliches Pinout
- ✓ Zentrierte Anschlussleitung
- ✓ Einfache und einheitliche Versorgungsspannungen
- ✓ Niederohmige Masseanbindung
- ✓ EMV-optimiertes und EMV-abgeschirmtes Design
- ✓ Automatische Erkennung von insgesamt 16 Displayversionen möglich
- ✓ LED-Backlight-Treiber mit weitem Versorgungsbereich (5V-15V) integriert
- ✓ Einfache Regelung der Displayhelligkeit über PWM-Signal oder digitales Kommando
- ✓ Typische Helligkeit von 1000 cd/m² bei einer Lebensdauer von min. 50.000 Stunden

2 Display

Im folgenden Kapitel wird die Displayschnittstelle von **ACT I³** sowie deren zugehörige Timings näher beschrieben. Des Weiteren besitzen einige Displays dieser Serie zusätzlich eine Schnittstelle zur Konfiguration, auf die im weiteren Verlauf ebenfalls eingegangen wird.

¹In diese Arbeit sind Auszüge der Masterarbeit "Universeller Displaytester auf FPGA-Basis", die von dem Autor 2016 der Fakultät für Elektrotechnik und Informationstechnik an der Hochschule München vorgelegt wurde, eingegangen.

2.1 Pinbelegung

Tabelle 1 zeigt die Pinbelegung von **ACT I³**. Es werden lediglich zwei Spannungen benötigt: 3.3V zur Versorgung des Analog- und Digitalteils des Displays sowie 5V bis 15V für den Backlight-Treiber. Es wird empfohlen, auf Applikationsseite beide Spannungen hinreichend mit Keramik Kondensatoren nach Masse abzublocken².

Pin	Bezeichnung	Funktion
1	GND	Masse
2	VDD	Versorgungsspannung des Displays (3.3V)
3	VDD	Versorgungsspannung des Displays (3.3V)
4	VCC	Versorgungsspannung des Backlight-Treibers (5V - 15V)
5	VCC	Versorgungsspannung des Backlight-Treibers (5V - 15V)
6	PWM	PWM-Eingang des Backlight-Treibers (siehe Kapitel 2.4)
7	GND	Masse
8	R0	Datenbit 0, rot (LSB)
9	R1	Datenbit 1, rot
10	R2	Datenbit 2, rot
11	R3	Datenbit 3, rot
12	GND	Masse
13	R4	Datenbit 4, rot
14	R5	Datenbit 5, rot
15	R6	Datenbit 6, rot
16	R7	Datenbit 7, rot (MSB)
17	GND	Masse
18	G0	Datenbit 0, grün (LSB)
19	G1	Datenbit 1, grün
20	G2	Datenbit 2, grün
21	G3	Datenbit 3, grün
22	GND	Masse
23	G4	Datenbit 4, grün
24	G5	Datenbit 5, grün
25	G6	Datenbit 6, grün
26	G7	Datenbit 7, grün (MSB)
27	GND	Masse
28	B0	Datenbit 0, blau (LSB)
29	B1	Datenbit 1, blau
30	B2	Datenbit 2, blau
31	B3	Datenbit 3, blau
32	GND	Masse

²z.B. 2.2μF X7R

33	B4	Datenbit 4, blau
34	B5	Datenbit 5, blau
35	B6	Datenbit 6, blau
36	B7	Datenbit 7, blau (MSB)
37	GND	Masse
38	HS	Horizontale Synchronisation (Zeilensynchronisation)
39	VS	Vertikale Synchronisation (Bildsynchronisation)
40	GND	Masse
41	DE	Data-Enable-Signal
42	GND	Masse
43	DCLK	Pixelclock
44	GND	Masse
45	CS	SPI Chip-Select
46	SDIN	SPI MOSI
47	SCK	SPI Clock
48	DISPLAY CONTROL	Display-Enable
49	RESET	Display-Reset
50	GND	Masse

Tabelle 1: ACT I³-Pinbelegung

2.2 Schnittstelle zur Bildübertragung

2.2.1 Grundlagen

Bei dem von **ACT I³** verwendeten *digital pixel interface* (DPI), auch als TTL- oder RGB-Interface bezeichnet, werden die Daten unidirektional, parallel und synchron übertragen. DPI wird fast ausschließlich bei Displays verwendet, die keinen Bildspeicher besitzen und daher hohe Datenraten zur Ansteuerung benötigen. Die eigentliche Übertragung der Bilddaten erfolgt als ein linearer, serieller Datenstrom aus einzelnen Pixeln³. Jeder Pixel wird dabei parallel übertragen. Dies bedeutet, dass seine Farbinformation als paralleles Datenwort übergeben wird. Typische Farbtiefen sind 8 bis 24 Bit. Zur Übertragung der Bilddaten werden neben den Farbinformationen der Pixel auch einige Synchronisationsleitungen benötigt, über die sich das Display auf den Datenstrom synchronisieren kann. Hierzu zählen im Wesentlichen eine Synchronisationsleitung für neu zu übertragende Zeilen (horizontale Synchronisation) sowie eine zur Synchronisation auf jedes neue Bild (vertikale Synchronisation). Vor und nach den Bilddaten werden in jeweils horizontaler und vertikaler Richtung unsichtbare Bereiche mit ausschließlich schwarzen Pixeln übertragen. Diese Bereiche werden daher

³DPI ist das digitale Gegenstück zu der weit verbreiteten analogen VGA-Schnittstelle, bei der die Daten ebenfalls parallel, jedoch über einen analogen Pegel übertragen werden. Eine Wandlung von DPI in VGA ist daher vergleichsweise einfach und benötigt nur einen DAC. Eine Pixelclock wird bei VGA nicht benötigt, wodurch eine umgekehrte Wandlung erheblich aufwendiger zu realisieren ist.

auch als Schwarzschildern (engl. *porches*) bezeichnet und bewirken in der Übertragung somit eine schwarze Umrahmung, das sog. *Blanking*, der Bilddaten. Einige Displaycontroller benötigen diese Bereiche, um interne, analoge Komponenten regelmäßig und zu definierten Zeitpunkten mit einer Referenzspannung für schwarze Pixel abzugleichen bzw. zu versorgen. Die Synchronisationsimpulse für eine neue Zeile bzw. für ein neues Bild werden ebenfalls während des Blanking übertragen, so dass sich das Display bereits vor dem ersten sichtbaren Pixel synchronisieren kann. Eine weitere Signalleitung bei DPI gibt an, ob gerade sichtbare Bilddaten oder Blankingdaten übertragen werden (Tabelle 2).

Datenleitung	Funktion	Bemerkung
CLK, PCLK, DCLK	Pixelclock	-
HS, H-Sync, Line-Sync	Horiz. Synchronisation	Beginn neue Zeile
VS, V-Sync, Frame-Sync	Vert. Synchronisation	Beginn neues Bild
DE, Data-Enable, Data-Valid	Daten oder Blanking	-
RGB[]	Parallele Datenleitungen	typ. 8,16,18 oder 24

Tabelle 2: Typische Datenleitungen bei DPI

Eine DPI-Übertragung wird durch die Anzahl der Datenleitungen, acht Zeitintervalle, die Pixelfrequenz und die Polarität aller beteiligten Signalleitungen definiert. Hierbei werden üblicherweise alle horizontalen Zeitintervalle (TH_x) als Vielfache der Periodendauer der Pixelfrequenz und alle vertikalen Zeitintervalle (TV_x) als Vielfache der Dauer einer Zeile angegeben (Tabelle 3).

Parameter	Funktion	Angabeformat
f_{CLK}	Pixelfrequenz	$f_{CLK} = 1/T_{CLK}$
TH	Dauer einer Zeile	$\sum TH_x$
THS	H. Synchronisationsdauer	$N \cdot T_{CLK}$
THB	H. Backporchdauer	$N \cdot T_{CLK}$
THD	H. Datendauer (Anzahl d. Pixel in X-Richtung)	$N \cdot T_{CLK}$
THF	H. Frontporchdauer	$N \cdot T_{CLK}$
TV	Dauer für ein Bild	$\sum TV_x$
TVS	V. Synchronisationsdauer	$N \cdot TH$
TVB	V. Backporchdauer	$N \cdot TH$
TVD	V. Datendauer (Anzahl d. Pixel in Y-Richtung)	$N \cdot TH$
TVF	V. Frontporchdauer	$N \cdot TH$

Tabelle 3: Timingparameter einer DPI-Schnittstelle

Die Abbildungen 1 - 3 zeigen die horizontalen und vertikalen Timings.

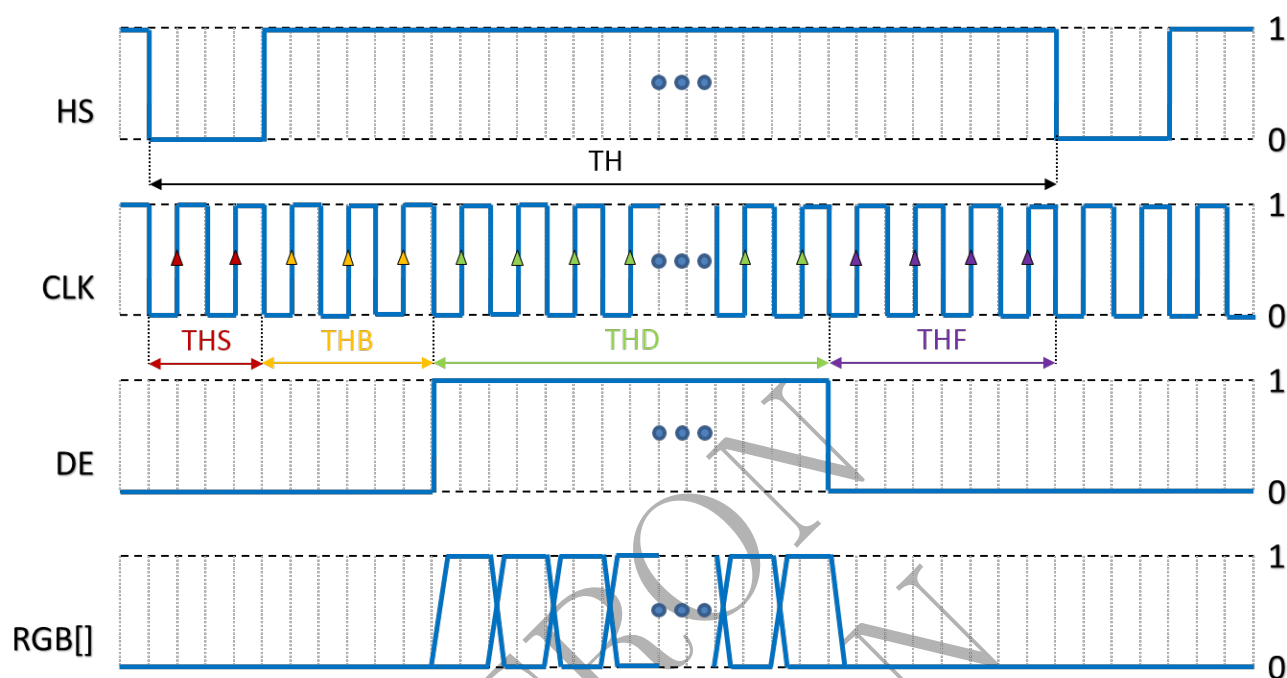


Abbildung 1: Horizontale Timings einer DPI-Schnittstelle

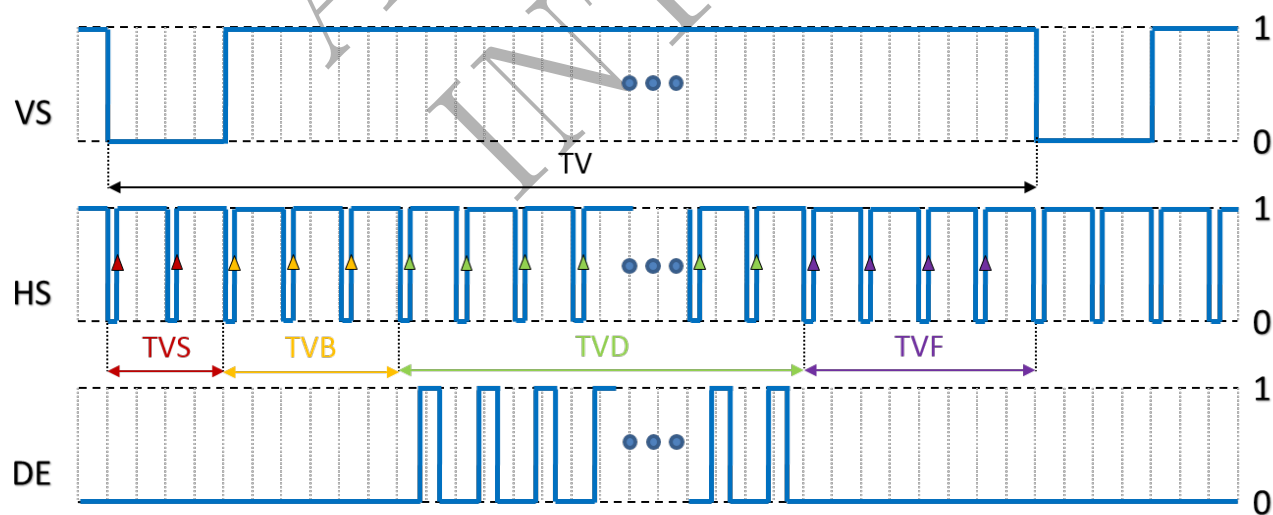


Abbildung 2: Vertikale Timings einer DPI-Schnittstelle

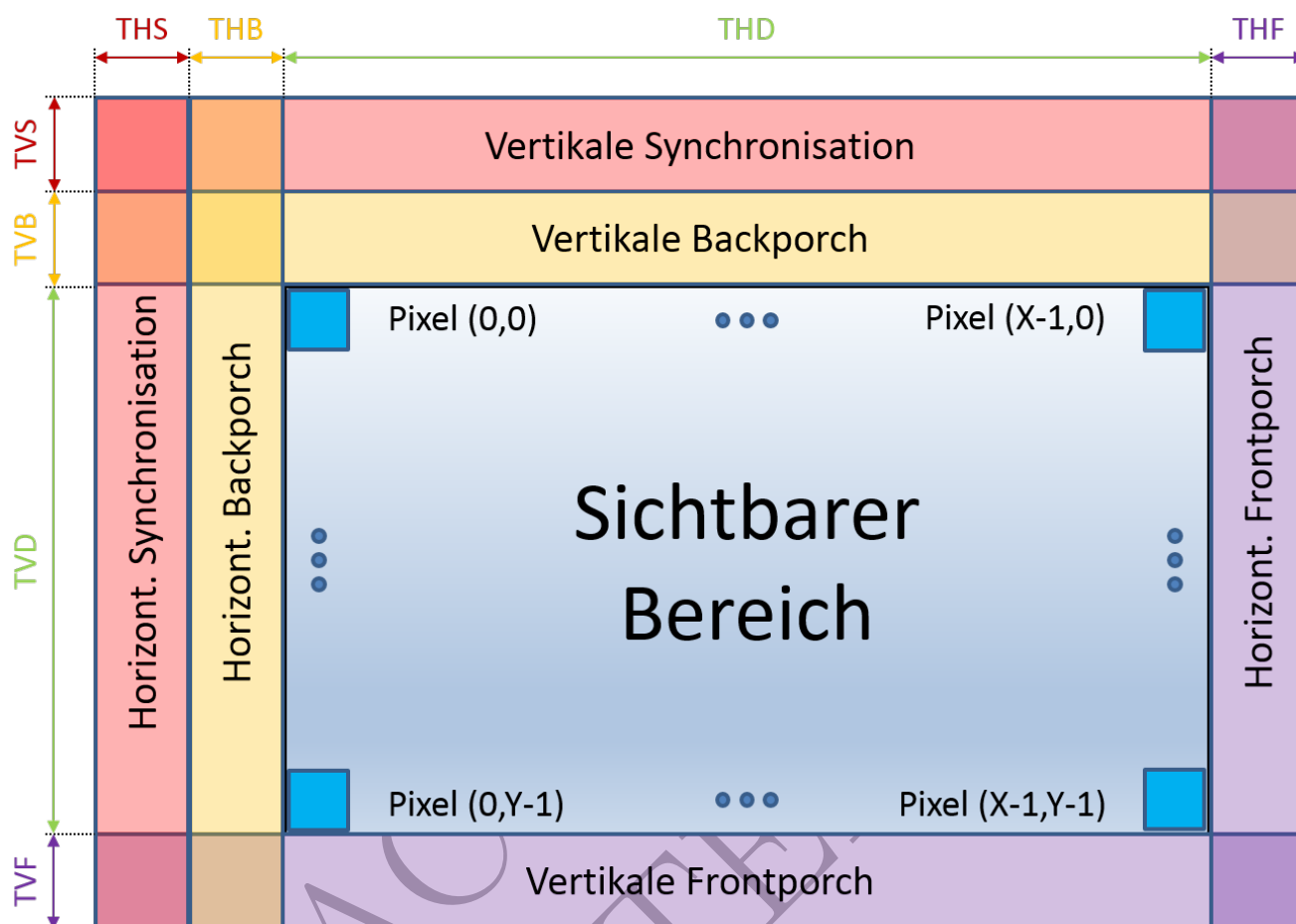


Abbildung 3: Gesamtübersicht der Timings bei DPI

In der Folge soll durch einige Berechnungen gezeigt werden, welche Datenraten sich beim Betrieb einer DPI-Schnittstelle ergeben bzw. welche Datenraten durch die Zielplattform bereitgestellt werden müssen. Als Beispiel werden die Timings eines 4.3" TFT-LCDs verwendet.

Die Periodendauer T_{PCLK} für einen Pixel berechnet sich zu:

Beispiel zur Berechnung der Periodendauer eines Pixels

$$T_{PCLK} := \frac{1}{f_{PCLK}} \quad (1)$$

$$T_{PCLK_{4.3" \text{ TFT-LCD}}} = \frac{1}{15 \text{ MHz}} = \underline{\underline{66.6 \text{ ns}}} \quad (2)$$

Hierüber kann die Gesamtdauer für eine Zeile (TH) als Summe aller horizontalen Zeitintervalle berechnet werden:

Gesamtdauer einer Zeile

$$TH := THS + THB + THD + THF \quad (3)$$

$$TH_{4.3'' \text{ TFT-LCD}} = (41 + 2 + 480 + 2) \cdot T_{PCLK} = \underline{\underline{35 \mu s}} \quad (4)$$

Nun kann die Dauer für ein vollständiges Bild (TV) und somit auch die Bildwiederholfrequenz, die sog. *Framerate*, wie folgt berechnet werden:

Gesamtdauer für ein Bild

$$TV := TVS + TVB + TVD + TVF \quad (5)$$

$$TV_{4.3'' \text{ TFT-LCD}} = (10 + 2 + 272 + 2) \cdot TH = \underline{\underline{10,01 \text{ ms}}} \quad (6)$$

$$Framerate_{4.3'' \text{ TFT-LCD}} := \frac{1}{TV} = \frac{1}{10,01 \text{ ms}} = 99,9 \text{ Hz} \approx \underline{\underline{100 \text{ Hz}}} \quad (7)$$

Für die oben berechnete Bildwiederholfrequenz von ca. 100 Hz und das genannte 4.3" Display kann zudem die benötigte Datenrate (inklusive Blanking) ermittelt werden. Diese Datenrate muss **kontinuierlich** bereitgestellt werden, selbst wenn der Bildinhalt konstant bleiben soll:

Benötigte Datenrate zur Bildübertragung

$$Datenrate_{DPI} := f_{PCLK} \cdot \dim\{RGB[]\} \cdot \text{Bit} \quad (8)$$

$$Datenrate_{4.3'' \text{ TFT-LCD}} = 15 \text{ MHz} \cdot 24 \text{ Bit} = \underline{\underline{45 \text{ MByte/s}}} \quad (9)$$

Bei einer unkomprimierten Übertragung der Bilddaten entspricht dies bereits in guter Näherung der maximalen Datenrate von USB 2.0⁴ und verdeutlicht daher den benötigten Hardwareaufwand von DPI. Während des Blankings muss die Zielplattform keine Daten aus dem Bildspeicher bereitstellen, sondern legt an alle Datenleitungen zur Farbübertragung eine logische 0 und überträgt somit ausschließlich schwarze Pixel. Hierdurch wird die effektive Datenrate, d.h. die gemittelte Datenrate des Bildspeichers, auch als *GRAM* bezeichnet, etwas reduziert. Die Größe η gibt dabei das Verhältnis zwischen dem rein sichtbaren Bereich und den gesamten Bilddaten inklusive Blanking an. Ihr Wertebereich liegt zwischen 0 und 1. Der jeweilige Wert hängt sowohl von dem

⁴Ausgegangen wird von einem durchschnittlichen Overhead bei USB 2.0 von 20% und einer Datenrate von 480 MBit/s. Somit ergibt sich eine maximale mittlere Nettodatenrate von ca. $0.8 \cdot 480 \text{ MBit/s} \cdot \frac{1 \text{ Byte}}{8 \text{ Bit}} = 48 \text{ MByte/s}$.

verwendeten Display als auch von dessen Konfiguration ab. Je kleiner der Wert, desto größer der Anteil des Blankings an den übertragenen Bilddaten, und um so niedriger die effektive Datenrate zur Bildübertragung. Ergibt sich ein Wert von exakt 1, so besitzt das Display keine Blanking-bereiche. Der theoretisch mögliche Wert 0 existiert in der Praxis nicht, da dies bedeuten würde, dass das Display entweder keinen sichtbaren Bereich besitzt oder ausschließlich ein schwarzes Bild darstellt.

Verhältnis zwischen sichtbarem Bereich und gesamten Bilddaten

$$\eta := \frac{f_{GRAM}}{f_{PCLK}} = \frac{THD}{TH} \cdot \frac{TV D}{TV} \quad (10)$$

$$\eta_{4.3'' \text{ TFT-LCD}} = \frac{480}{525} \cdot \frac{272}{286} = 0,87 = \underline{\underline{87\%}} \quad (11)$$

Da viele Displays einen weiten Bereich für das Blanking erlauben, kann auf diese Weise selbst bei einer vorgegebenen Pixelfrequenz die Datenrate zur Bildübertragung und damit der Mittelwert der Auslesefrequenz des Bildspeichers um ca. 20%-30% reduziert werden. Natürlich gilt dies nur für den primären Bildspeicher, z.B. das SDRAM, und nicht für den Cache. Das Blanking verursacht hier zwar eine Pause in der Übertragung, aber abgesehen davon muss der Cache die Bilddaten dennoch mit der vollen Pixelfrequenz bereitstellen.

Effektive Datenrate eines Displays

$$Datenrate_{GRAM} := Datenrate_{DPI} \cdot \eta \quad (12)$$

2.2.2 Timings

Tabelle 4 zeigt typische Timings der unterschiedlichen Displaygrößen bei **ACT I³**. Die Bedeutung der einzelnen Parameter ist im vorherigen Kapitel 2.2.1 beschrieben. Die zulässigen Bereiche sind den Datenblättern der jeweiligen Displays zu entnehmen.

Parameter	3.5"	4.3"	5"	7"	Angabeformat
f_{CLK}	6,5 MHz	9 MHz	33 MHz	33 MHz	$f_{CLK} = 1/T_{CLK}$
TH	408	525	1056	1056	$\sum THx$
THS	0	41	128	128	$N \cdot T_{CLK}$
THB	68	2	88	88	$N \cdot T_{CLK}$
THD	320	480	800	800	$N \cdot T_{CLK}$
THF	20	2	40	40	$N \cdot T_{CLK}$
TV	262	286	525	525	$\sum TVx$
TVS	0	10	2	2	$N \cdot TH$
TVB	18	2	33	33	$N \cdot TH$
TVD	240	272	480	480	$N \cdot TH$
TVF	4	2	10	10	$N \cdot TH$
Pol. PCLK	↑	↓	↓	↓	Übernahmeflanke
Pol. HS	AL	AL	AL	AL	AL=active low, AH=active high
Pol. VS	AL	AL	AL	AL	AL=active low, AH=active high
Pol. DE	AL	AH	AH	AH	AL=active low, AH=active high

Tabelle 4: Typische Displaytimings bei ACT I³

2.3 Schnittstelle zur Konfiguration

Bei dem von **ACT I³** als Konfigurationsschnittstelle eingesetzten *serial peripheral interface* (SPI) handelt es sich um eine synchrone, serielle Schnittstelle, die ursprünglich von Motorola entwickelt wurde⁵. Da SPI keine offizielle Spezifikation zu Grunde liegt, gibt es eine Vielzahl an Variationen dieser Schnittstelle. Im Allgemeinen wird die anzusprechende Baugruppe, in dieser Arbeit das Display, über eine sog. *Chip-Select-Leitung* (CS) oder *Slave-Select-Leitung* (SS) selektiert. Danach werden synchron zu einer Taktleitung (meist SCLK oder SCK) seriell Daten übertragen. Die Datenübertragung erfolgt meist in Vielfachen von 8 Bit. Je nachdem, ob die Übertragung uni- oder bidirektional erfolgt, werden eine bzw. zwei Datenleitungen benötigt⁶. Als Bezeichnungen für die Datenleitungen (Tabelle 5) verwendet man üblicherweise die Begriffe *serial data in* (SDI) und *serial data out* (SDO). Da diese leicht zu Verwechslungen in Bezug auf die Signalrichtung führen können, sind auch die Bezeichnungen *master out slave in* (MOSI) und *master in slave out* (MISO) immer häufiger anzutreffen, bei denen das Display als Slave zu betrachten ist. Ähnlich wie bei vielen ICs

⁵Quelle: https://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus

⁶In seltenen Fällen wird auch eine bidirektionale Datenleitung verwendet.

(z.B. ADCs) muss auch bei Displays zwischen der Übertragung von Nutzdaten (hier Bilddaten) und Kommandos (z.B. zur Initialisierung) unterschieden werden. Hierfür wird im Wesentlichen eine der folgenden Methoden verwendet:

- Einführung einer Präambel
- Verwendung einer zusätzlichen Datenleitung D/C zur Signalisierung von Daten oder Kommandos
- Kodierung der D/C-Information in einem zusätzlichen (z.B. neunten) Bit
- Feste Unterteilung des Datenrahmens: z.B. Verwendung von 1 Byte für Kommando und 3 Bytes für Daten

Displays mit SPI-Schnittstelle werden überwiegend unidirektional betrieben, da das Auslesen der Daten nur in seltenen Fällen benötigt wird (z.B. für XOR- oder Read-Modify-Write-Operationen). Daher benötigt ein solches Display zur Ansteuerung, je nachdem, ob eine D/C-Leitung verwendet wird, 3 oder 4 Datenleitungen. Häufig unterstützen Displays mehrere der oben genannten Methoden zur Unterscheidung von Kommandos und Daten, wodurch oft in Datenblättern von *3-wire* und *4-wire* SPI die Rede ist. Es soll an dieser Stelle besonders hervorgehoben werden, dass diese Bezeichnungen ausschließlich die Verwendung (*4-wire*) bzw. das Nichtvorhandensein (*3-wire*) einer D/C-Leitung signalisieren, und nicht, wie fälschlicherweise oft angenommen, ob ein uni- oder bidirektionales SPI verwendet wird.

Datenleitung	Funktion	Bemerkung
CS, nCS, $\overline{\text{CS}}$, SS	Baugruppe selektieren	wird nicht immer benötigt
SCLK, SCK	Clockleitung	-
SDO, MOSI	Datenausgang	-
SDI, MISO	Dateneingang	-
D/C	Kommando oder Daten senden	wird nicht immer benötigt

Tabelle 5: Typische Datenleitungen bei SPI

Prinzipieller Ablauf eines Schreibvorgangs (Abb. 4):

1. Zunächst wird über **D/C** signalisiert, ob ein Kommando oder Nutzdaten übertragen werden.
2. Es erfolgt die Selektierung des anzusprechenden Empfängers über **nCS**.
3. Das erste Datenbit wird angelegt.
4. Der Sender übergibt das Datenbit je nach Konfiguration mit einer steigenden oder fallenden Flanke an **SCK**. Danach wiederholen sich die Schritte 3 und 4 *n*-mal.
5. Abschließend wird der Teilnehmer deselektiert.

Da **ACT I³** SPI ausschließlich als Konfigurationsschnittstelle und nicht zur Übertragung von Bilddaten⁷ verwendet, wird keine D/C-Leitung benötigt. Schritt 1 entfällt somit.

⁷ACT I³-fähige Displays besitzen keinen internen Bildspeicher.

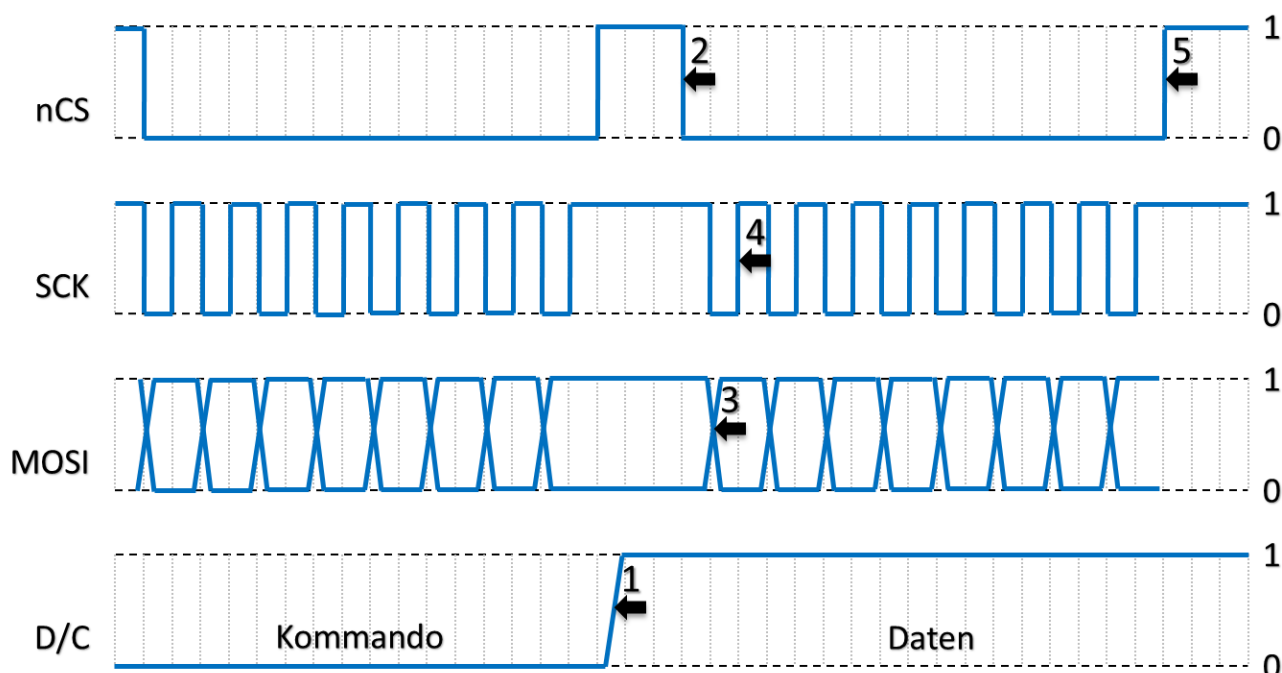


Abbildung 4: Beispiel für einen Schreibvorgang einer SPI-Schnittstelle

Vorteile:

- Sehr einfache, leicht mit jedem Mikrocontroller zu realisierende bzw. per Software zu emulierende Schnittstelle
- Nur wenige Datenleitungen
- Einfache Busanalyse möglich

Nachteile:

- Niedrige Datenraten (ca. 1 - 20 MBit/s)
- Overhead⁸ möglich

2.4 LED-Backlight

Alle Displays der **ACT I³**-Serie verfügen über einen integrierten LED-Backlight-Treiber. Dieser besitzt einen weiten Eingangsspannungsbereich von 5V bis 15V, wodurch die Ansteuerung der Displays vereinfacht wird. Da die Backlight-Versorgung direkt auf dem Display erfolgt, werden auch elektromagnetische Störungen, bedingt durch die Schaltfrequenz von LED-Treibern, auf ein Minimum reduziert. Als Backlight-Treiber verwendet **ACT I³** den IC TPS61165 der Firma Texas

⁸Als *Overhead* bezeichnet man Daten, die zusätzlich zu den Nutzdaten übertragen werden. Diese Daten werden von einigen Bussen zwingend benötigt und dienen z.B. der Datenflusssteuerung, der Adressierung oder zur Fehlerkorrektur.

Instruments. Hierbei handelt es sich um einen qualitativ hochwertigen Step-Up-Converter mit einem Wirkungsgrad von bis zu 90%.

3 Touch

3.1 I²C

Der I²C-Bus (engl. *inter-integrated circuit*) wurde ursprünglich von der Firma Philips im Jahr 1982 entwickelt. Dieser Bus unterliegt einer detaillierten Spezifikation, so dass keine grundlegenden Variationen, wie beispielsweise bei SPI, existieren⁹. Aus Lizenzgründen wird lediglich von einigen Herstellern die alternative Bezeichnung *two wire interface* (TWI) verwendet. Die Datenübertragung bei I²C erfolgt bidirektional, seriell und synchron. Es werden immer 2 Datenleitungen (Tabelle 6) benötigt: *serial clock* (SCL) und *serial data* (SDA).

Datenleitung	Funktion	Bemerkung
SCL	Clockleitung	Open-Collector-Ausgang
SDA	Datenleitung	Open-Collector-Ausgang

Tabelle 6: Typische Datenleitungen bei I²C

Beide Datenleitungen werden nie aktiv, sondern nur über Pullup-Widerstände nach high gezogen, wodurch sämtliche Busteilnehmer, die in Master und Slave unterteilt sind¹⁰, ausschließlich Open-Collector-Ausgänge verwenden dürfen. Dies schließt ein Gegeneinandertreiben der Teilnehmer aus. Zusätzlich arbeitet I²C nach dem Prinzip der sog. *Kollisionsvermeidung* (engl. *carrier sense multiple access/collision avoidance*, CSMA/CA). Hierbei prüft jeder Master, bevor er Daten sendet, ob der Bus gerade frei ist. Ist dies nicht der Fall, so bricht der Teilnehmer die Übertragung ab und versucht es zu einem späteren Zeitpunkt erneut. Die I²C-Kommunikation basiert im Wesentlichen auf folgenden Regeln:

- Jede Übertragung bei I²C besteht grundsätzlich aus Vielfachen von 9 Bit, da nach jedem übertragenen Byte ein sog. *Acknowledge-Bit* übertragen wird, über das der Empfänger den Erhalt der Daten bestätigt.
- Im Ruhezustand, auch als *Idle-State* bezeichnet, werden beide Leitungen über die Pullup-Widerstände auf eine logische 1 gezogen. Alle am Bus angeschlossenen Teilnehmer schalten ihre Open-Collector-Ausgänge hochohmig.
- Die Taktfrequenz der Clockleitung wird durch den Master bereitgestellt¹¹.
- Der Zustand der Datenleitung darf sich nur ändern, während SCL auf low liegt¹².

⁹Quelle: www.nxp.com/documents/user_manual/UM10204.pdf

¹⁰Jeder I²C-Bus benötigt mindestens einen Master und einen Slave. Meistens wird ein Master in Verbindung mit mehreren Slaves verwendet.

¹¹Der Slave kann durch sog. *clock stretching* eine Reduzierung der Datenrate erzwingen.

¹²Ausnahmen sind das Start- und Stop-Signal.

- Eine Datenübertragung kann nur durch einen Master initiiert werden.
- Liegt SCL auf high, so löst eine fallende Flanke an SDA das sog. *Start-Signal* und damit den Beginn einer Übertragung aus.
- Liegt SCL auf high, so löst eine steigende Flanke an SDA das sog. *Stop-Signal* und damit das Ende einer Übertragung aus.

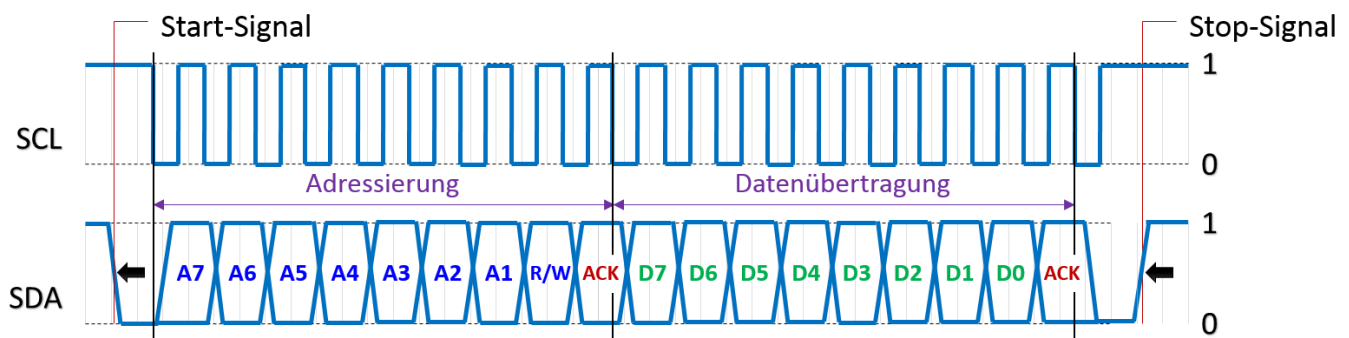


Abbildung 5: Timing einer I²C-Schnittstelle

Prinzipieller Ablauf einer Datenübertragung bei I²C (Abb. 6 und 7):

1. Der Master löst ein Start-Signal aus. Alle Teilnehmer erkennen dieses Signal und warten auf den Empfang der Adresse.
2. Der Master überträgt die 7-Bit-Adresse¹³ des anzusprechenden Slaves.
3. Anschließend sendet der Master das R/W-Bit, das angibt, ob er Daten senden oder empfangen möchte (0=senden, 1=empfangen).
4. Der angesprochene Slave quittiert den Erhalt seiner Adresse mit einem Acknowledge-Signal und zieht dazu SDA auf low.
5. Je nach übertragenem R/W-Bit sendet nun entweder der Master oder Slave ein oder mehrere Datenbytes.
6. Nach dem letzten Byte und dessen Acknowledge-Signal beendet der Master die Übertragung mit einem Stop-Signal und gibt damit den I²C-Bus wieder frei.

¹³I²C unterstützt auch die Verwendung einer 10-Bit Adresse, auf die in dieser Arbeit nicht näher eingegangen wird.

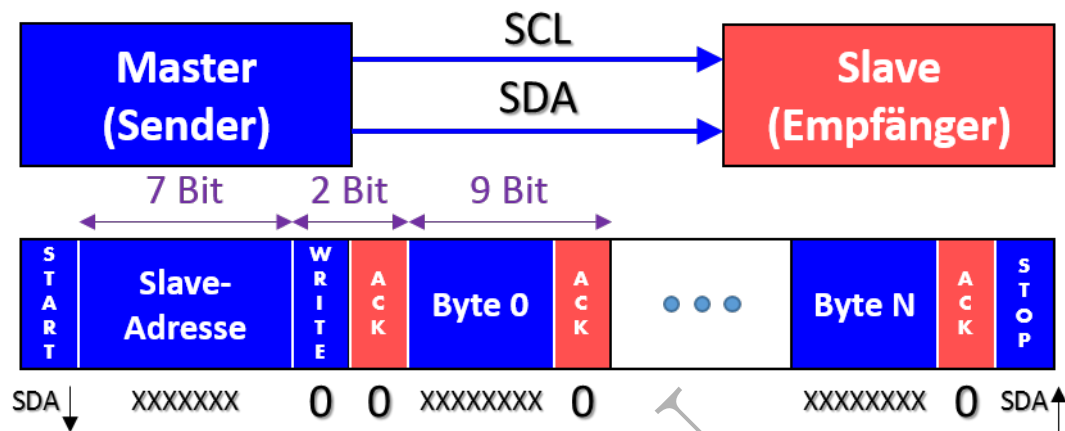


Abbildung 6: Ablauf einer I²C-Übertragung von Master an Slave

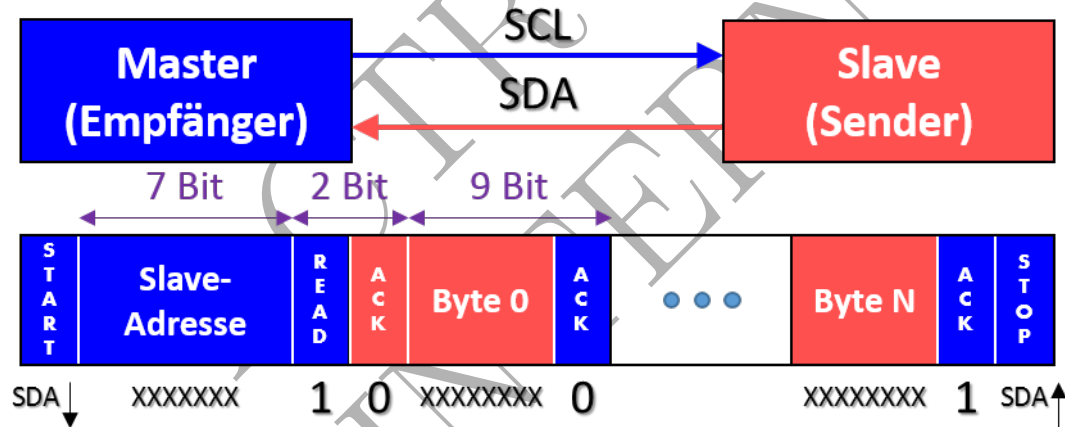


Abbildung 7: Ablauf einer I²C-Übertragung von Slave an Master

Vorteile:

- Detailliert spezifizierter Bus
- Unabhängig von der Anzahl der Teilnehmer nur zwei Datenleitungen
- Einfache Busanalyse möglich

Nachteile:

- Sehr niedrige Datenraten (ca. 0.1 - 1 MBit/s)
- Zwingend von Bus benötigter Overhead

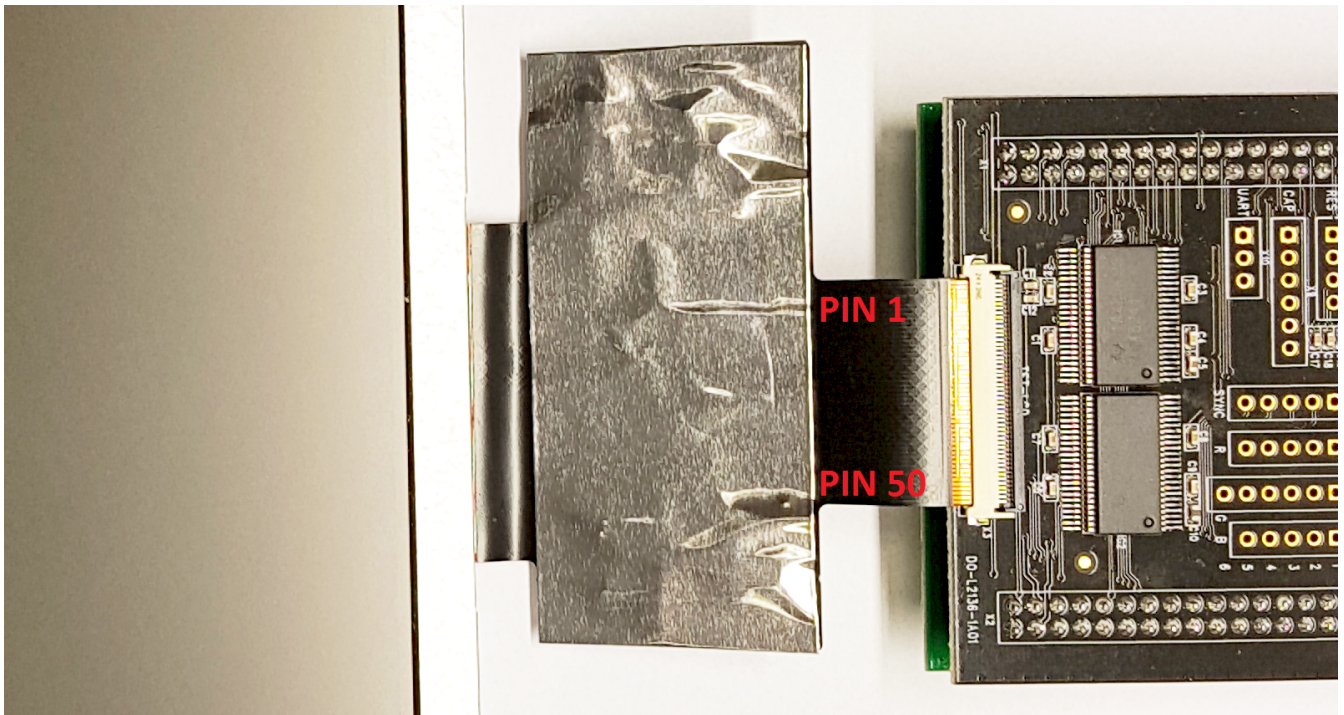


Abbildung 9: Anschließen des Displays an die ACT I³-Entwicklungsumgebung

4.1.2 Konfiguration der Displaygröße

Zur Konfiguration der angeschlossenen Displaygröße werden die Schalter **S1** und **S2** des Mäuseklaviers (Abb. 10) benötigt. Die möglichen Konfigurationen sind in Tabelle 7 dargestellt.

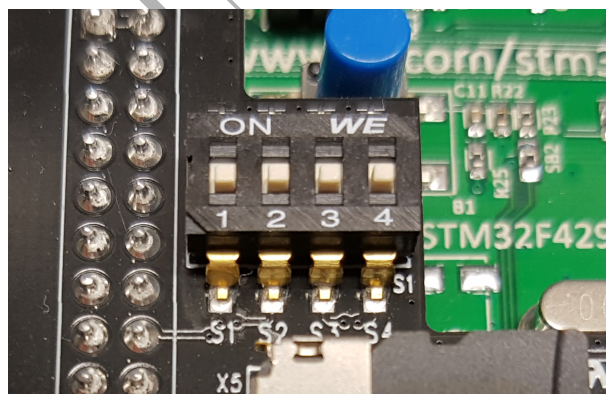


Abbildung 10: Mäuseklavier der ACT I³-Entwicklungsumgebung

S1	S2	Displaygröße
OFF	OFF	3.5"
ON	OFF	4.3"
OFF	ON	5"
ON	ON	7"

Tabelle 7: Konfiguration der Displaygröße über das Mäuseklavier

4.1.3 Konfiguration von Touchscreens

Die **ACT I³**-Entwicklungsumgebung erkennt über I²C angeschlossene kapazitive Touchscreens automatisch. Zur Inbetriebnahme des analog-resistiven Touchcontrollers STMPE811 wird **S4** des Mäuseklaviers verwendet (Tabelle 8).

S4	Touchcontroller STMPE811
OFF	deaktiviert
ON	aktiviert

Tabelle 8: Konfiguration des analog-resistiven Touchcontrollers STMPE811

4.1.4 Spannungsversorgung

Zur Versorgung der **ACT I³**-Entwicklungsumgebung dient die Mini-USB-Buchse des STM32F429-Discovery-Kits (Abb. 11). Bitte beachten Sie, dass insbesondere die Displaygrößen 5" und 7" eine USB-Stromaufnahme von **über 500 mA** verursachen und somit die USB-Spezifikation verletzen. Es wird daher dringend empfohlen, die **ACT I³**-Entwicklungsumgebung ausschließlich über einen gepufferten USB-Hub oder ein USB-Netzteil mit einem Nennstrom von mindestens 1.3A zu versorgen. Sollte die **ACT I³**-Entwicklungsumgebung an einer unzureichenden Quelle betrieben werden, so kann die 5V-Versorgung auf einige Volt zusammenbrechen. Hierdurch steigt der Strom des Backlight-Treibers an, was schließlich zu dessen thermischen Zerstörung führen kann.

ACHTUNG

Vergewissern Sie sich, dass **D2** und **D3** des Discovery-Kits überbrückt sind. Hierdurch wird u.a. die 5V-USB-Versorgung direkt mit der 5V-Versorgungsspannung der Platine verbunden. Beachten Sie daher unbedingt, dass eine fehlerhafte Einspeisung einer externen Spannung, bei gleichzeitiger Versorgung über die USB-Mini-Buchse, den angeschlossenen PC beschädigen kann!



Abbildung 11: Mini-USB-Buchse des STM32F429-Discovery-Kits

4.2 Hardware

Im folgenden Kapitel werden die Hardwarekomponenten sowie die Schaltpläne der **ACT I³**-Entwicklungsumgebung vorgestellt.

4.2.1 STM32-Mikrocontroller

Bei der STM32F4x9-Serie der Firma STMicroelectronics handelt es sich um eine äußerst leistungsfähige Mikrocontroller-Familie. Neben den üblichen Peripheriekomponenten, wie beispielsweise Timern und ADCs, enthalten diese Mikrocontroller eine spezielle Hardware zur direkten Ansteuerung von TFT-LCDs (Abb. 12¹⁴). Dieser sog. *LCD-TFT Display Controller* (LTDC) verfügt zudem über einen ebenfalls integrierten, speziell für die Ansteuerung von Displays entwickelten DMA, den sog. *Chrom-ART Accelerator* (DMA2D), wodurch eine Hardwarebeschleunigung für die Darstellung von aufwendigen Grafiken ermöglicht wird.

¹⁴Quelle: www.st.com

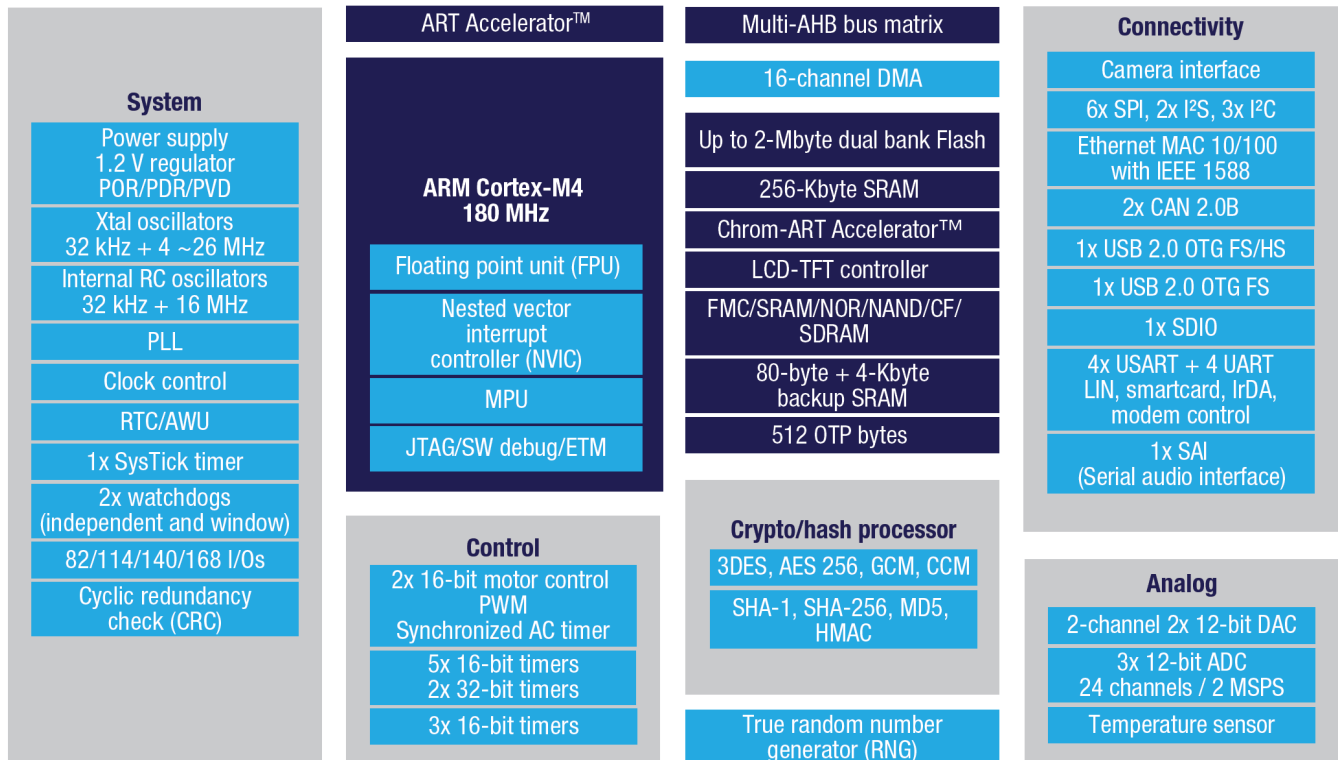


Abbildung 12: Kernkomponenten der STM32F4x9-Serie

Zur Anbindung des STM32F429-Discovery-Kits an **ACT I³** wurde eine Adapterplatine entwickelt, welche die entsprechenden Signale des Mikrocontrollers mit denen der Displayschnittstelle verbindet. Zusätzlich bietet diese Adapterplatine u.a. folgende Komponenten:

- Buchse für Micro-SD-Karte
- Mäuseklavier, 4-polig
- RGB-LED
- Analog-resistiven Touchcontroller STMPE811
- Mehrere Stiftheuten, um Zugriff auf Schnittstellen des STM32 zu erhalten

4.2.2 Schaltplan

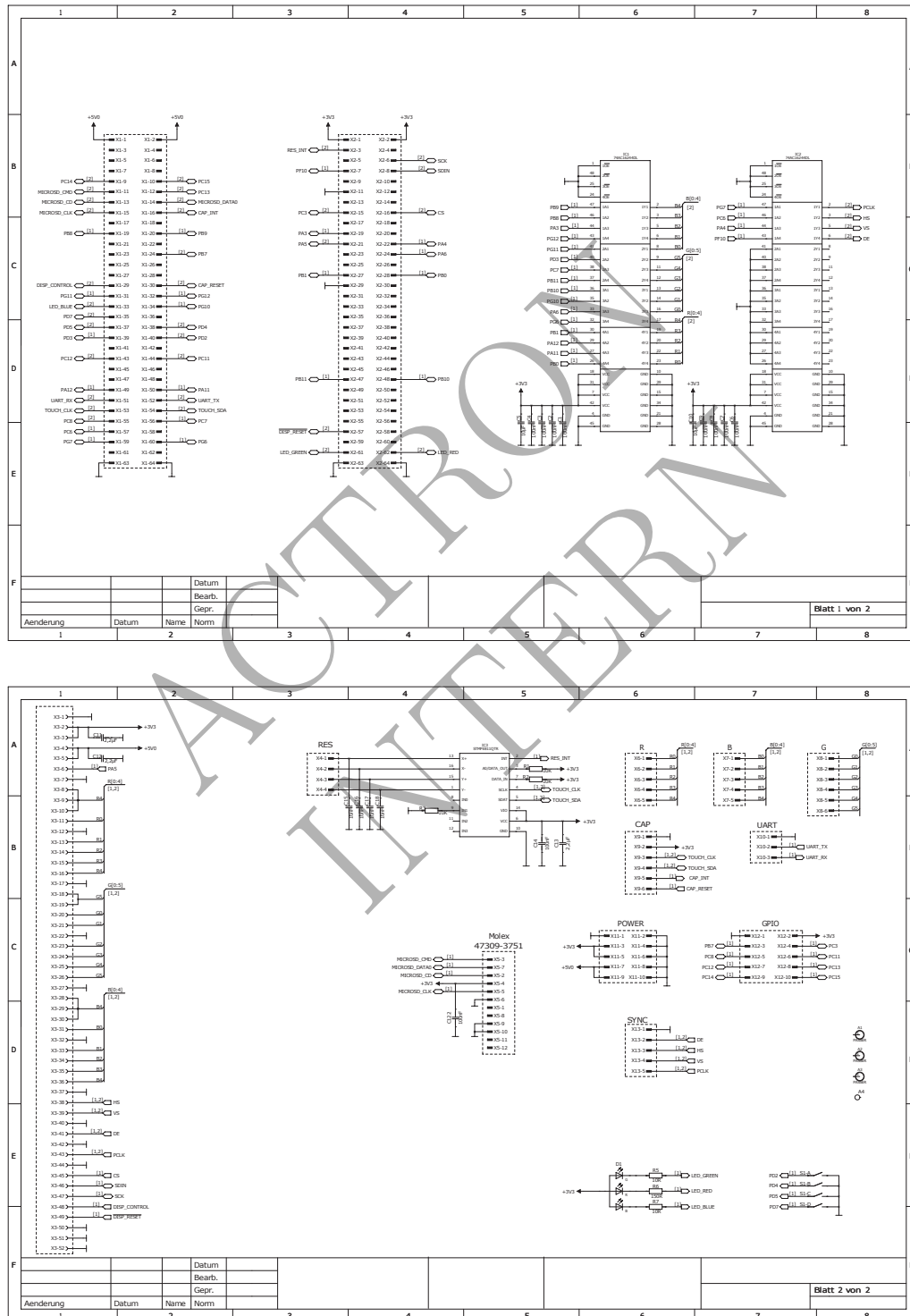


Abbildung 13: Schaltplan der ACT I³-Adapterplatte für das STM32F429-Discovery-Kit

4.2.3 Pinbelegung der Stiftleisten

Die **ACT I³**-Entwicklungsumgebung bietet neben der Displayschnittstelle eine Vielzahl weiterer Anschlussmöglichkeiten. Diese werden über Stiftleisten (nicht bestückt) herausgeführt. Analog-resistive Touchscreens können über die Stiftleiste **RES** (Tabelle 9) mit dem bereitgestellten Touchcontroller STMPE811 verbunden werden.

Pin	Bezeichnung	Funktion
1	X+	AR-Touch rechts (R)
2	X-	AR-Touch links (L)
3	Y+	AR-Touch oben (U,T)
4	Y-	AR-Touch unten (D,B)

Tabelle 9: Pinbelegung der Stiftleiste RES

Zur Ansteuerung von kapazitiven Touchscreens steht die Stiftleiste **CAP** (Tabelle 10) bereit. Sie führt die I²C-Schnittstelle des STM32F429 heraus und kann daher auch für andere I²C-fähige Baugruppen verwendet werden. Die Interruptleitung kann über die Software sowohl für steigende als auch fallende Flanken konfiguriert werden.

Pin	Bezeichnung	Funktion	STM32-Port
1	GND	Masse	-
2	3V3	Versorgungsspannung für Touchcontroller	-
3	TOUCH_CLK	I ² C-Clockleitung	PA8
4	TOUCH_SDA	I ² C-Datenleitung	PC9
5	CAP_INT	Interrupt-Leitung des Touchcontrollers	PE3
6	CAP_RESET	Reset-Leitung des Touchcontrollers	PG14

Tabelle 10: Pinbelegung der Stiftleiste CAP

Die Stiftleiste **UART** bietet eine UART-Schnittstelle z.B. zur Kommunikation mit einem PC (Tabelle 11).

Pin	Bezeichnung	Funktion	STM32-Port
1	GND	Masse	-
2	UART_TX	UART-Ausgang des STM32	PA9
3	UART_RX	UART-Eingang des STM32	PA10

Tabelle 11: Pinbelegung der Stiftleiste UART

Um in der Lage zu sein, neben Displays und Touchscreens auch andere Baugruppen anzusteuern, stellt die Stiftleiste **GPIO** mehrere universelle GPIOs des STM32F429 zur Verfügung. Hierüber können u.a. serielle Busse, wie beispielsweise SPI oder I²C, bereitgestellt werden.

Pin	Bezeichnung	Funktion	STM32-Port
1	GND	Masse	-
2	3V3	3.3V	-
3	PB7	Universal GPIO	PB7
4	PC3	Universal GPIO	PC3
5	PC8	Universal GPIO	PC8
6	PC11	Universal GPIO	PC11
7	PC12	Universal GPIO	PC12
8	PC13	Universal GPIO	PC13
9	PC14	Universal GPIO	PC14
10	PC15	Universal GPIO	PC15

Tabelle 12: Pinbelegung der Stiftleiste GPIO

Sämtliche parallele Farbinformationen der DPI-Schnittstelle liegen an den Stiftleisten **R**, **G** und **B** (Tabelle 13 - 15). Die zugehörigen Synchronisationssignale stehen an **SYNC** bereit (Tabelle 16). Dies ermöglicht beispielsweise, die schnellen Signale an einem digitalen Speicheroszilloskop näher zu untersuchen. Um eine hohe Flankensteilheit sowie Signalintegrität zu gewährleisten, werden alle DPI-Signale des STM32F429 über Leitungstreiber ausgegeben. Diese können selbstverständlich in der späteren Applikation entfallen.

Pin	Bezeichnung	Funktion	STM32-Port
1	R0	Datenbit 0, rot	PB0
2	R1	Datenbit 1, rot	PA11
3	R2	Datenbit 2, rot	PA12
4	R3	Datenbit 3, rot	PB1
5	R4	Datenbit 4, rot	PG6

Tabelle 13: Pinbelegung der Stiftleiste R

Pin	Bezeichnung	Funktion	STM32-Port
1	G0	Datenbit 0, grün	PA6
2	G1	Datenbit 1, grün	PG10
3	G2	Datenbit 2, grün	PB10
4	G3	Datenbit 3, grün	PB11
5	G4	Datenbit 4, grün	PC7
6	G5	Datenbit 5, grün	PD3

Tabelle 14: Pinbelegung der Stiftleiste G

Pin	Bezeichnung	Funktion	STM32-Port
1	B0	Datenbit 0, blau	PG11
2	B1	Datenbit 1, blau	PG12
3	B2	Datenbit 2, blau	PA3
4	B3	Datenbit 3, blau	PB8
5	B4	Datenbit 4, blau	PB9

Tabelle 15: Pinbelegung der Stiftleiste B

Pin	Bezeichnung	Funktion	STM32-Port
1	GND	Masse	-
2	DE	Data Enable	PF10
3	HS	Horizontale Synchronisation	PC6
4	VS	Vertikale Synchronisation	PA4
5	PCLK	Pixelclock	PG7

Tabelle 16: Pinbelegung der Stiftleiste SYNC

Die Versorgung externer Baugruppen kann über die Stiftleiste **POWER** erfolgen (Tabelle 17). Ebenso kann hierüber auch die Spannungsversorgung der **ACT I³**-Entwicklungsumgebung eingespeist werden. Bitte beachten Sie hierzu unbedingt die Hinweise in Kapitel 4.1.4.

Pin	Bezeichnung	Funktion
1	GND	Masse
2	GND	Masse
3	3V3	3.3V
4	GND	Masse
5	3V3	3.3V
6	GND	Masse
7	5V0	5.0V
8	GND	Masse
9	5V0	5.0V
10	GND	Masse

Tabelle 17: Pinbelegung der Stiftleiste POWER

4.3 Software

Wird noch ergänzt.

Abbildungsverzeichnis

1	Horizontale Timings einer DPI-Schnittstelle	6
2	Vertikale Timings einer DPI-Schnittstelle	6
3	Gesamtübersicht der Timings bei DPI	7
4	Beispiel für einen Schreibvorgang einer SPI-Schnittstelle	12
5	Timing einer I ² C-Schnittstelle	14
6	Ablauf einer I ² C-Übertragung von Master an Slave	15
7	Ablauf einer I ² C-Übertragung von Slave an Master	15
8	ACT I ³ -Entwicklungsumgebung	16
9	Anschließen des Displays an die ACT I ³ -Entwicklungsumgebung	17
10	Mäuseklavier der ACT I ³ -Entwicklungsumgebung	17
11	Mini-USB-Buchse des STM32F429-Discovery-Kits	19
12	Kernkomponenten der STM32F4x9-Serie	20
13	Schaltplan der ACT I ³ -Adapterplatine für das STM32F429-Discovery-Kit	21

Tabellenverzeichnis

1	ACT I ³ -Pinbelegung	4
2	Typische Datenleitungen bei DPI	5
3	Timingparameter einer DPI-Schnittstelle	5
4	Typische Displaytimings bei ACT I ³	10
5	Typische Datenleitungen bei SPI	11
6	Typische Datenleitungen bei I ² C	13
7	Konfiguration der Displaygröße über das Mäuseklavier	18
8	Konfiguration des analog-resistiven Touchcontrollers STMPE811	18
9	Pinbelegung der Stiftleiste RES	22
10	Pinbelegung der Stiftleiste CAP	22
11	Pinbelegung der Stiftleiste UART	22
12	Pinbelegung der Stiftleiste GPIO	23
13	Pinbelegung der Stiftleiste R	23
14	Pinbelegung der Stiftleiste G	23
15	Pinbelegung der Stiftleiste B	24
16	Pinbelegung der Stiftleiste SYNC	24
17	Pinbelegung der Stiftleiste POWER	24

Version	Datum	Beschreibung
v1.1	25.11.2016	▪ Korrektur einiger Fehler.
v1.0	24.11.2016	▪ Erste Veröffentlichung. ▪ Dokumentation erstellt mit \LaTeX .

ACTRON
INTERN