

Entwicklung und Optimierung von Display-Schnittstellen fuer embedded Linux Boards

-

Masterarbeit

im Fachgebiet Hard- und Softwareentwicklung



**GEORG-SIMON-OHM
HOCHSCHULE NÜRNBERG**

vorgelegt von: Armin Schlegel

Studiengebiet: Fakultaet EFI

Matrikelnummer: 2020863

Erstgutachter: Prof. Dr. Joerg Arndt

Zweitgutachter: Peter Meier

© 2014



Dieses Werk einschließlich seiner Teile ist **urheberrechtlich geschützt**. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Autors unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen sowie die Einspeicherung und Verarbeitung in elektronischen Systemen.



Zusammenfassung

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Quisque Natural-Programmierung blandit sed, hendrerit at, pharetra eget, dui. Sed lacus. Pellentesque malesuada. Cras gravida mi id sapien. Ut risus justo, fermentum non, scelerisque sit amet, lacinia in, erat. Proin nec lorem. Quisque porta, nisl at porta aliquam, felis libero consequat ipsum, vitae scelerisque dolor mi a odio. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Duis sollicitudin. Proin sollicitudin varius arcu. Morbi eleifend, metus sit amet placerat pharetra, dolor dui lobortis pede, vel imperdiet tellus eros imperdiet lorem. In hac habitasse platea dictumst. Curabitur elit mi, facilisis nec, ultricies id, aliquet et, magna. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam ac est. Mauris turpis enim, feugiat non, imperdiet congue, scelerisque non, purus. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Nullam dictum aliquet purus. Maecenas faucibus. Maecenas suscipit.

Abstract

Fusce neque est, tincidunt eu, nonummy nec, tempor iaculis, erat. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Vestibulum egestas, velit a rhoncus gravida, metus dolor pulvinar diam, sit amet placerat risus dolor sit amet elit. Maecenas eget purus ut est mattis porta. Suspendisse ut mi et mauris lobortis malesuada. Vestibulum dapibus. Duis hendrerit, elit eu venenatis eleifend, sapien ante volutpat odio, ac condimentum tellus massa ut massa. Etiam dapibus imperdiet metus. Sed sapien arcu, pulvinar quis, laoreet quis, venenatis non, justo. Aliquam est ante, pulvinar nec, accumsan sed, auctor sed, augue.

Ut adipiscing ligula. In mattis. Ut varius. In nec nulla at eros molestie viverra. Duis dolor risus, lobortis vel, dictum a, pellentesque id, lectus. Sed suscipit orci ac ligula venenatis condimentum. Maecenas et sem lacinia tortor cursus tempus. Mauris pellentesque risus at nulla. In arcu. Curabitur mattis mi quis dolor. In leo. Vivamus ut libero.



Inhaltsverzeichnis

Abbildungsverzeichnis	IV
Tabellenverzeichnis	V
1. Einleitung	1
1.1. Motivation	1
1.2. Ziel der Arbeit	1
1.3. Aufbau der Arbeit	2
1.4. Typographische Konventionen	2
2. Theoretische Grundlagen	3
2.1. Video-Schnittstellen	3
2.1.1. VGA	3
2.1.2. DVI	4
2.1.3. HDMI	5
2.1.4. RGB	5
2.1.5. LVDS	6
2.1.6. 8080-Interface	6
2.1.7. Bewertung der Video-Schnittstellen	8
2.2. Betrachtete Embedded Linux Boards	8
2.2.1. Raspberry Pi	9
2.2.2. GnuBlin Extended	9
2.2.3. Bewertung der Linux-Boards	9
3. Teil A	10
3.1. Untersuchte Displays mit 8080-Interface	10
3.1.1. 4.3“/5“ mit SSD1963	10
3.1.2. 3.2“ mit SSD1289	12
3.1.3. 5“ mit CPLD	12
3.2. 8080-Interface mittels SRAM-Interface	13
3.2.1. Konzept	14
3.2.2. MPMC - Multiport Memory Controller des NXP LPC313x .	15
3.2.3. Hardwareverbindung zwischen SRAM-Interface und Display (Adapterplatine)	17



Inhaltsverzeichnis

3.2.4. Software	17
3.2.4.1. Entwicklung eines Linux-Framebuffer-Treibers	17
3.2.4.1.1. Anpassungen für Display mit SSD1289 Controller	17
3.2.4.1.2. Anpassungen für Display mit SSD1963 Controller	17
3.2.4.1.3. Anpassungen für Display mit CPLD Controller	17
3.2.4.2. Entwicklung eines User-Space-Treibers	17
3.2.4.2.1. Anpassungen für Display mit SSD1289 Controller	17
3.2.4.2.2. Anpassungen für Display mit SSD1963 Controller	17
3.2.4.2.3. Anpassungen für Display mit CPLD Controller	17
3.2.4.3. Anpassung des APEX-Bootloaders zur Verwendung des Displays	17
3.2.4.4. Probleme bei der Entwicklung und Fehlersuche	17
3.2.4.4.1. Probleme mit SSD1963	17
3.2.4.4.1.1. Rolle des User-Space-Treibers	17
3.2.4.4.1.2. Debuggen mit Logik-Analyzer	17
3.3. Known Bugs	18
3.3.1. Known Bugs SSD1963	18
3.3.2. Known Bugs SSD1289	18
3.3.3. Known Bugs CPLD Display	18
4. Teil B	19
4.1. Konzept	19
4.2. Hardwareentwicklung	19
4.2.1. Spannungsversorgung	19
4.2.2. HDMI-RGB-Bridge	19
4.2.3. RGB-LVDS-Bridge	19
4.2.4. EDID-Daten	19
4.3. Software	19
4.3.1. EDID-Daten auf embedded Seite	19
4.3.1.1. Konzept	19
4.3.1.2. Low-Level-Treiber	19
4.3.1.2.1. UART-Treiber	19
4.3.1.2.2. I2C-Treiber	19
4.3.1.3. Programmablauf	20



Inhaltsverzeichnis

4.3.2. EDID-Daten auf PC Seite	20
4.3.2.1. Konzept	20
4.3.2.2. GTK GUI mit Glade	20
4.3.2.3. Programmablauf	20
4.4. Known Bugs	20
4.4.1. Hardware	20
4.4.1.1. HDMI-Stecker gekreuzt, CON2	20
4.4.1.2. LVDS-Steckerfootprint gespiegelt, CON6	20
4.4.1.3. +12V PWM Hintergrundbeleuchtung	20
4.4.1.4. +5V Kreis / Widerstand R13	20
4.4.1.5. USB D+/D- vertauscht R22, R23	20
4.4.2. Software	20
4.4.2.1. EDID Programmer	20
5. Zusammenfassung	21
Literaturverzeichnis	22
Eidesstattliche Erklärung	24
A. Anhang	i



Abbildungsverzeichnis

2.1.	VGA-Timing, Quelle: Valcarce [2011]	4
2.2.	RGB-Timing, Quelle: Texas Instruments [2011]	5
2.3.	8080-Timing des SSD1289, Quelle: Solomon Systech Limited [2007] .	7
3.1.	8080-Display Pinout, Quelle: Coldtears Electronics	11
3.2.	NXP LPC313x EBI, Quelle: NXP Semiconductors [2010]	14
3.3.	NXP LPC313x MPMC, Quelle:	16



Tabellenverzeichnis

2.1. Relevanz der Display-Schnittstellen für die Masterarbeit	8
3.1. Relevante Kommandos des SSD1963, Solomon Systech Limited [2008]	11
3.2. Relevante Kommandos des SSD1289, Solomon Systech Limited [2007]	12
3.3. Relevante Kommandos des MD050SD, ITEAD Studios [2013]	13
3.4. MPMC Register	15



-

1. Einleitung

1. Einleitung

1.1. Motivation

In der heutigen Zeit treten eingebettete Systeme (engl. embedded systems) immer stärker in den Vordergrund. Gerade in den Bereichen der Industrie, Telekommunikation oder Multimedia wächst der Bedarf an Lösungen die durch Zuverlässigkeit, Energiesparsamkeit und kompakter Bauform bestechen.

Obwohl eingebettete Systeme meist für den Anwender unsichtbar ihren Dienst verrichten, sind sie doch inzwischen allgegenwärtig. Im Bereich der Telekommunikation und Unterhaltungselektronik kommt ein solches System im Prinzip nicht mehr ohne ein Display aus. Die Möglichkeit zur Anzeige multimedialer Daten wird zur Kaufentscheidung. Auch hier gilt die Maxime: besser, schneller, größer.

Im Sektor der eingebetteten Systeme spielen Betriebssystem wie Linux neben diversen anderen Systemen wie beispielsweise RTOS, OSEK, QNX oder auch Windows eine sehr große Rolle. In Verbindung zeigen eingebettete Linuxsysteme mit Displays ein großes Potential. Mit der beliebten ARM-Architektur lassen sich so kostengünstige, leistungsstarke Systeme aufbauen, die die gestellten Aufgaben gut erfüllen kann. Sieht man sich allein den Marktanteil von Smartphones welche auf Android-Basis arbeiten an, wird der Trend klar, dass Hersteller eine offene Basis bevorzugen. [Brandt \[2013\]](#)

Es ist ersichtlich, dass auch in Zukunft Linux auf eingebetteten Systemen eine immer größere Rollen spielen wird.

1.2. Ziel der Arbeit

Das Ziel dieser Arbeit ist zu zeigen, dass die Verwendung von Displays mit eingebetteten Linux Systemen je nach Anforderung einfach oder über Umwege realisierbar ist.



-
1. Einleitung

1.3. Aufbau der Arbeit

Im ersten Teil der Arbeit werden theoretische Grundlagen gebildet, die für das Verständnis nötig sind. Hier werden Standards wie z.B. HDMI bzw. DVI, LVDS und RGB behandelt. Es wird ein Überblick über ausgewählte embedded Linux Boards gegeben und diese Klassifiziert mit welchen Displayschnittstellen diese ausgestattet sind bzw.. ausgestattet werden können. Der Zweite Teil behandelt das embedded Linux Board 'Gnublin', welches von Haus aus keine Displayschnittstelle vorgesehen hat. Hier werden zwei Varianten zur Ansteuerung von Displays erarbeitet. Die Ansteuerung wird hierbei vom Prozessor erledigt, da das 'Gnublin' keine dedizierten Grafikcontroller besitzt. Im dritten Teil wird für leistungsstärkere embedded Linux-Systeme mit HDMI-Schnittstelle eine Hardware entwickelt, RGB- oder LVDS-Panels anzuschließen. Um die Displays über die entwickelte Hardware anzusteuern, wird der dedizierte Grafikcontroller der Boards verwendet.

1.4. Typographische Konventionen



-

2. *Theoretische Grundlagen*

2. Theoretische Grundlagen

In diesem Kapitel werden Theoretische Grundlagen geschaffen, die zum weiteren Verständnis der Arbeit benötigt werden. Zuerst werden ausgewählte Video-Schnittstellen erläutert und verglichen und bewertet welchen praktischen Nutzen diese für handelsübliche embedded Linuxsysteme bietet. Im Weiteren werden zwei Linux Boards verglichen und bewertet sowie deren praktische Einsatzgebiete beispielhaft dargelegt.

2.1. Video-Schnittstellen

Unter Video-Schnittstellen kann man die Schnittstellen verstehen, die direkt zur Anzeige von Bilddaten dienen und physikalisch mit einer Anzeigeeinheit verbunden sind. Hier können sowohl Hardware- als auch Softwarekomponenten enthalten sein.

2.1.1. VGA

Unter VGA versteht man Video Graphics Array und wurde 1987 von IBM entwickelt. Der Stecker hat 15 Pins und liefert neben analogen Farbinformationen Horizontale und Vertikale Synchronisationssignale. Aufgrund der limitierten Spezifikationen ist die Schnittstelle eher antik und selbst Intel als Chipsethersteller will ab 2015 auf die Schnittstelle verzichten und digitalen Schnittstellen den Vorzug lassen ([Knuppfer \[2010\]](#)). Zwar ist die VGA-Schnittstelle noch nicht komplett obsolet, so wird sie den digitalen Schnittstellen trotzdem weichen müssen. Der Trend bei embedded Linuxsystemen ist zumindest der, dass handelsübliche Systeme direkt mit HDMI oder anderen digitalen Schnittstellen entwickelt werden. Die Funktionsweise der VGA-Schnittstelle ist in Abbildung 2.1 zu sehen. Es werden fünf analoge Leitungen benötigt: R, G, B, HSYNC¹ und VSYNC². Die ersten drei stellen die Farbwerte Rot, Grün und Blau dar. Je nach Intensität der Farbkanäle lassen sich aus einer Mischung jede Farbe darstellen. Zur Steuerung der Intensität können Pegel zwischen 0V (absolut dunkel) und +0.7V (absolut hell) pro Farbkanal angenommen werden. Die Signale HSYNC und VSYNC werden zur Steuerung der Zeilen und Spalten verwendet.

¹Hsync: Horizontale Synchronisation

²Vsync: Vertikale Synchronisation



2. Theoretische Grundlagen

Das Signal HSYNC zeigt an, wann eine Zeile vollständig ist. Während der HSYNC-Periode werden für jeden Pixel der Zeile zeitlich exakte Pulse auf den Farbleitungen angelegt. Sind alle Zeilen eines Bildes komplett, wird das VSYNC-Signal angestoßen, welches ein neues Bild von vorne beginnt ([Valcarce \[2011\]](#)).

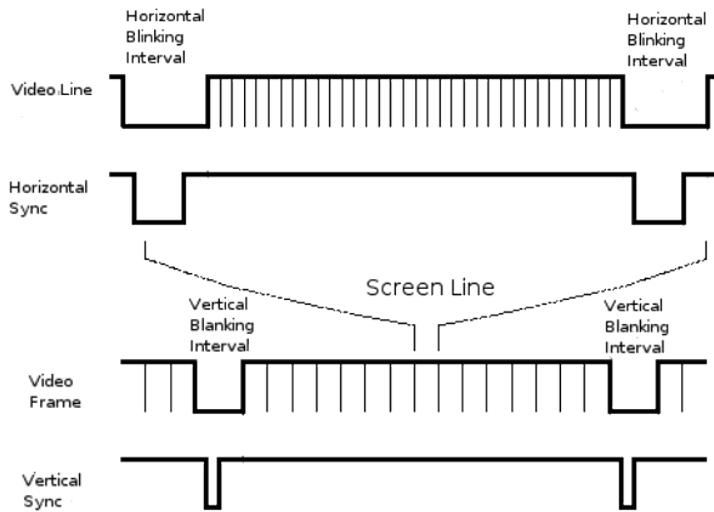


Abbildung 2.1.: VGA-Timing, Quelle: [Valcarce \[2011\]](#)

2.1.2. DVI

Hinter DVI steht der Begriff Digital Visual Interface und stellt ein digitale Schnittstelle zur Grafikanzeige dar. Der DVI Standard wurde 1999 von der DDWG³ verabschiedet, da der Wunsch nach Leistungsstärkeren Schnittstellen vorhanden war. QXGA-Auflösungen⁴ sind auf analogem Wege nicht mehr befriedigend erzielbar. Die DVI Schnittstelle beinhaltet neben den Digitalen Signalen zusätzlich analoge VGA Signale, was den Betrieb älterer Monitore und Displays zulässt. Zur digitalen Datenübertragung wird der TMDS⁵ Standard verwendet, welcher die 24 Bit Farbinformationen⁶ mittels eines Serializers in serielle Daten umwandelt. Je nach benötigter Bandbreite können drei oder sechs Aderpaare für Pixeldaten verwendet werden. Dies wird Single-Link bzw. Double-Link genannt und es lassen sich dabei max. 3.72 GBit/s⁷ bzw. 7.44 GBit/s⁸ übertragen. Um die Paare zuordnen zu können, wird ein weiteres Paar zur Synchronisation verwendet. Um die Übertragung

³Digital Display Working Group

⁴QXGA: 2048x1536

⁵Transition Minimized Differential Signaling - Differentielle Datenübertragung

⁶24 Bit: je 8 Bit für Rot, Grün und Blau

⁷max. UXGA: 1600x1200@60Hz

⁸max. WUXGA: 1920x1200@60Hz



2. Theoretische Grundlagen

noch effizienter zu gestalten, gibt es die Möglichkeit bei High- sowie Low-Pegel des Taktsignals Daten zu übertragen⁹ ([Leunig \[2002\]](#)).

2.1.3. HDMI

Gegeneber der DVI-Schnittstelle bietet die HDMI Schnittstelle dieselben Eigenschaften bezüglich der Videoübertragung verwendet zur ebenfalls TMDS. Hinzu kommt allerdings, dass sowohl Audio als auch Verschlüsselung unterstützt werden. Der Formfaktor der Stecker sind für den Hausgebrauch verkleinert worden. HDMI wurde als normierte Universallösung entwickelt und hat sich als solche etabliert ([Extron \[2014\]](#)). Nahezu jedes neu entwickelte Gerät mit Anzeigemöglichkeit, bietet eine HDMI-Schnittstelle - ebenso embedded Linux Boards wie z.B. bekannte Linux Boards wie Raspberry Pi oder Beagle Bone Black.

2.1.4. RGB

Der RGB-Bus, verwendet für kleine TFT-Panels bis ca. 7", funktioniert prinzipiell analog zur VGA-Schnittstelle, mit dem Unterschied, dass die Datenleitungen komplett digital sind. So werden die Signale für Rot, Grün und Blau nicht mehr analog im Bereich von 0V bis +0.7V dargestellt, sondern durch einen üblicherweise acht Bit breiten Bus pro Farbkanal. Die Auflösung pro Farbkanal ist mit 255 Intensitätsstufen gerechnet ausreichend um ein gesamtes Farbspektrum von 16777216¹⁰ Farben zu erhalten. Dieser Farbmodus wird auch RGB888 genannt, da acht Bit für jede Far-

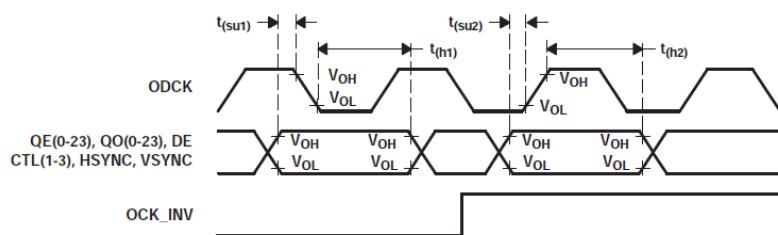


Abbildung 2.2.: RGB-Timing, Quelle: [Texas Instruments \[2011\]](#)

be zur Kodierung, insgesamt also 24 Bit, zur Verfügung stehen. Neben dem 24 Bit Modus ist RGB565 noch weit verbreitet, der je fünf Bit für Rot und Blau und sechs Bit für Grün verwendet. Hier ergibt sich ein Farbspektrum von 65536¹¹ Farben. Da digital übertragen wird, ist eine Takteleitung notwendig um die Synchronizität zu ermöglichen. Aufgrund der Verbreitung und Mächtigkeit der Schnittstelle besitzen

⁹Double Data Rate

¹⁰16777216 Farben = 2^{24}

¹¹65536 = 2^{16}



2. Theoretische Grundlagen

einige Prozessor, wie z.B. der Linuxfähige OMAP3530 von Texas Instruments, eine RGB-Schnittstelle. Abbildung 2.2 zeigt exemplarisch ein Timing-Diagramm der RGB-Schnittstelle des Bausteins TFP-401A von Texas Instruments.

2.1.5. LVDS

Um lange Strecken und große Bildformate übertragen zu können ist der parallele Datentransfer ungeeignet, da bei schnellem Takt z.B. das Übersprechen zu groß wird und das Signal schneller gestört wird. Deshalb ist die Praktik beliebt, große Datenmengen über eine differentielle Verbindung wie z.B. LVDS¹² zu übertragen. Die physikalische Funktionsweise der LVDS Leitung liegt darin, dass zweimal das selbe Signal übertragen wird - mit positiver Spannung und mit negativer Spannung. Wirkt nun von außen eine Störung auf die LVDS Leitung, werden beide Leitungen - positive wie auch die negative - gleichermaßen gestört. Durch das Zusammenführen beider Signale am Ende, kompensieren sich diese Störungen im Idealfall zu Null. Wie auch LVDS arbeitet das zuvor genannte TMDS ähnlich, da es sich hierbei auch um eine differentielle Übertragungsart handelt. Der Unterschied liegt in der Verwendung. TMDS wird oft eingesetzt, sobald das Signal das Gerät verlässt - z.B. Desktop-Bildschirm mit Anschlusskabel. Befindet sich das Anzeigegerät allerdings im selben Gehäuse, so wird oft LVDS eingesetzt. Neben Bilddaten ist es natürlich auch möglich andere Nutzdaten wie z.B. Sensordaten zu übertragen. Aufgrund der hohen Geschwindigkeit und geringen Fehlerrate werden differentiellen Übertragungen werden gerne für Displays angewendet.

2.1.6. 8080-Interface

Das 8080-Interface ist eine antike Schnittstelle ursprünglich von Intel 8080 Prozessor. Sie wird bis heute verwendet, um Speicher, kleine TFT-Displays oder andere Bausteine mit einem Mikrocontroller zu betreiben. Eckdaten des 8080-Interface sind sowohl der Datenbus selbst als auch der Adressbus mit z.B. acht, 16 oder 32 Bit, je eine eine Leitung für Read-Enable, Write-Enable und Chip-Select. Durch die Verwendung der Chip-Select Leitungen ist es möglich mehrere Teilnehmer am selben Bus zu betreiben. Alle Teilnehmer, deren Chip-Leitung nicht aktiv ist, verhalten sich für andere Busteilnehmer unsichtbar. Erst mit Zuweisung der Chip-Selects werden diese sichtbar und übernehmen den Bus. Ein Hostsystem steuert als sog. Master die am Bus hängenden Slaves. Moechte das Hostsystem von einem Slave Daten lesen, wird ein Lesezyklus initiiert, der die Chip-Select Leitung aktiviert, die gewünschte

¹²LVDS: Low Voltage Differential Signaling



2. Theoretische Grundlagen

Adresse an den Bus anlegt, die Read-Enable Leitung aktiviert und nach einer festgelegten Zeit diese wieder deaktiviert. Der Slave legt die gewünschten Daten auf den Datenbus und der Host kann diese Daten korrekt lesen. Analog dazu funktioniert der Schreibzyklus ähnlich. Abbildung 2.3 zeigt das Timing Diagramm eines Schreib- und Lesezyklus des Displaycontrollers SSD1289. Das Signal D/C wird verwendet um zu unterscheiden, ob ein Daten oder ein Kommando auf dem Bus anliegen. Dazu kann beispielsweise eine Adressleitung des 8080-Bus verwendet werden. CS stellt das Chip-Select dar. WR und RD beziehen sich auf Write- bzw. Read-Enable. D0-D17 sind 18 Datenbits des Bus ([Solomon Systech Limited \[2007\]](#)).

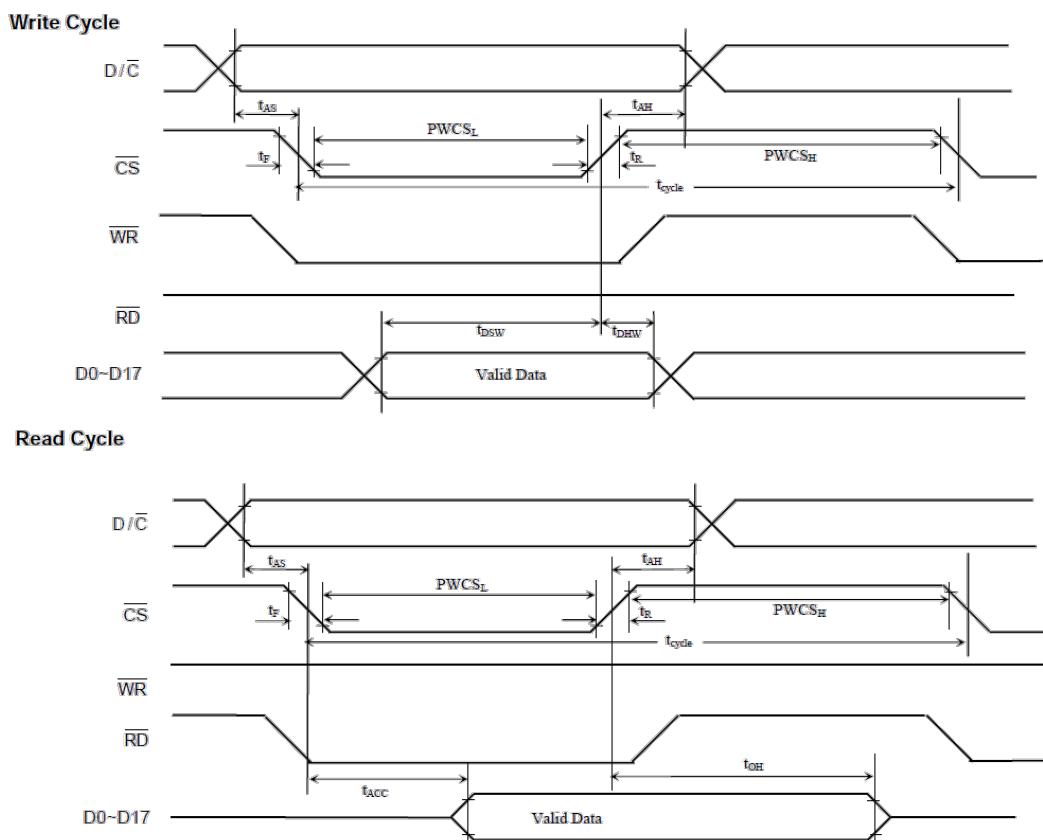


Abbildung 2.3.: 8080-Timing des SSD1289, Quelle: [Solomon Systech Limited \[2007\]](#)

Viele Mikrocontroller besitzen bereits ein 8080-Interface dediziert in Hardware - allerdings nicht alle. Als Ersatz kann das Protokoll mit GPIO¹³ in Software implementiert werden. Dies ist allerdings wesentlich langsamer als eine Lösung, die bereits in Hardware läuft, da GPIO-Pins nicht dafür geschaffen sind, sich mit schneller Frequenz schalten zu lassen.

¹³GPIO: General Purpose In/Output



2. Theoretische Grundlagen

2.1.7. Bewertung der Video-Schnittstellen

Nachdem nun die wichtigsten Schnittstellen dargestellt wurden, werden diese im Folgenden mit dem Fokus auf die Masterarbeit hinsichtlich der Relevanz bewertet. Da die VGA-Schnittstelle antik und obsolet ist, spielt sie heutzutage nur noch eine geringe Rolle. Insbesondere im Bereich der embedded Systeme wird sie kaum verwendet. Für die Masterarbeit ist die VGA-Schnittstelle uninteressant, da diese von keiner, in der Masterarbeit behandelten, Hardware verwendet wird. Jedoch wurde diese eingangs behandelt, da diese den Übergang zum digitalen RGB-Bus schafft. Die DVI- und HDMI-Schnittstellen, welche für den Bereich der Videoanzeige praktisch identisch sind, nehmen jedoch einen hohen Stellenwert in der Masterarbeit ein. Im zweiten Teil der Arbeit wird eine Hardware entwickelt, welche als Eingangssignale die TMDS der DVI-/HDMI-Schnittstelle nutzt. Ebenso spielen die RGB-Schnittstelle und LVDS eine große Rolle, da an diesen Schnittstellen der entwickelten Hardware TFT-Panels angeschlossen werden.

Neben den reinen Video-Schnittstellen weist das beschriebene 8080-Interface, das ursprünglich nicht zur Bildübertragung gedacht war, ein hohes Potential auf und besitzt für den ersten Teil der Masterarbeit hohen Stellenwert. Gerade im embedded Bereich besitzt diese Schnittstelle nach wie vor einen hohen Stellenwert, da vor allem kleine Displays damit hinreichend schnell und effizient betrieben werden können. Tabelle 2.1 zeigt nochmals eine kurze Übersicht der Relevanz der einzelnen Schnittstellen für die Masterarbeit.

große Rolle: umschreiben

Schnittstelle	Relevanz für Masterarbeit	Verwendung in der Masterarbeit
VGA	keine	-
DVI	mittel	Teil B
HDMI	hoch	Teil B
RGB	hoch	Teil B
LVDS	hoch	Teil B
8080-Interface	hoch	Teil A

Tabelle 2.1.: Relevanz der Display-Schnittstellen für die Masterarbeit

2.2. Betrachtete Embedded Linux Boards

In diesem Abschnitt werden die verwendeten Linux-Boards dargestellt, verglichen und hinsichtlich der Verwendbarkeit in der Masterarbeit bewertet. Da sich die Masterarbeit in zwei Teile gliedert, wird für beide Anwendungsfälle ein typisches Linux-Board hergehoben, welches den Anforderungen gerecht werden muss, eine billige und effiziente Anzeige zu gestatten.



-

2. Theoretische Grundlagen

2.2.1. Raspberry Pi

Am wohl bekanntesten und mit einer riesigen Community hinter dem Projekt ist der Raspberry Pi von der Raspberry Pi Foundation ¹⁴. Um die wichtigsten Eckdaten des Einplatinenrechners im Checkkartenformat zu nennen, besitzt er in der Ausfuehrung Model B einen ARM11-Core (Broadcom BCM2835), 512 Megabyte SDRAM, eine Broadcom VideoCore IV GPU sowie diverse Schnittstellen wie HDMI, USB 2.0, UART¹⁵, SPI¹⁶, I2C¹⁷ sowie GPIO-Pins¹⁸.

Der erschwingliche Preis macht den Raspberry Pi attraktiv und zieht die Community an, da man für rund 40 Euro einen kompletten Rechner bekommt.

Lehrstuhl	Professor
BWL	Maier
MB	Müller
Jura	Schmidt

2.2.2. Gnublin Extended

2.2.3. Bewertung der Linux-Boards

8080-
Interface
in SRAM
erwäh-
nen!

¹⁴<http://www.raspberrypi.org>

¹⁵UART: Universal Asynchronous Receiver Transmitter - RS2323

¹⁶SPI: Serial Peripheral Interface - 4 Draht Bus

¹⁷I2C: Inter Integrated Circuit - 2 Draht Bus

¹⁸GPIO: General Purpose Input Output



3. Teil A

Im Folgenden Kapitel wird Teil A dieser Arbeit behandelt. Es wird die Ansteuerung von TFT Displays über den 8080-Bus auf Basis des GnuBlin Linuxboards realisiert. Hierzu wurden verschiedene große LCD Displays mit unterschiedlichen Controllern unter Verwendung des 8080-Interface und untersucht.

3.1. Untersuchte Displays mit 8080-Interface

Dieser Abschnitt behandelt die untersuchten Displays. Es wurden drei Displays aus China untersucht. Der Fokus bei der Bestellung lag vor allem darauf, dass die Pinbelegung der jeweiligen Displays übereinstimmen. So ist die Entwicklung von nur einer Adapterplatine zwischen GnuBlin und Display nötig. Alle verwendeten Displays werden im 16 Bit Farbmodus betrieben. Die hieraus resultierende Farbtiefe beträgt 65.535 Farben.

Alle verwendeten Displays arbeiten dahingehend gleich, dass sie Kommandos und Daten auf dem Datenbus anliegen, diese jedoch durch eine gesonderte Leitung unterscheiden werden. Soll dem Display also etwas mitgeteilt werden, so muss zuerst ein entsprechendes Kommando und im Anschluss die Nutzdaten gesendet werden. Um Pixeldaten an das Display zu senden, hat sich die Vorgehensweise etabliert, eine Rechteckige Region im RAM des Displays zu reservieren, das durch die 4 Eckpunkte des Rechtecks definiert sind. Werden im Anschluss Pixeldaten gesendet, inkrementiert der Controller die Adresse automatisch und springt bei einem Zeilenumbruch automatisch an die richtige Stelle im RAM. Der maximale Speicher im Controller beschränkt die maximale Auflösung der ansteuerbaren TFT-Panel. Trotz der Tatsache, dass sich die Displays auf elektrischer Seite nicht unterscheiden, so müssen diese allerdings alle speziell softwareseitig behandelt werden.

3.1.1. 4.3“/5“ mit SSD1963

Die Wahl des Controllers SSD1963 von Solomon Systech liegt nahe, da dieser bereits mit einem 4.3“ Panel in einer vorausgehende Arbeit verwendet wird. Dort ist das Display mittels GPIO-Pins am Raspberry Pi angeschlossen. Die Software bezüglich der reinen Displayansteuerung ist somit bereits vorhanden (siehe Schlegel [2013]).



3. Teil A

Aufgrund eines Problems, das in Abschnitt 3.3.1 näher beschrieben ist, wird für diese Arbeit zusätzlich ein anderes Display mit 5“ Panel aber selbem Controller untersucht. Abbildung 3.1 zeigt das Pinout der verwendeten Displays. Die Displays

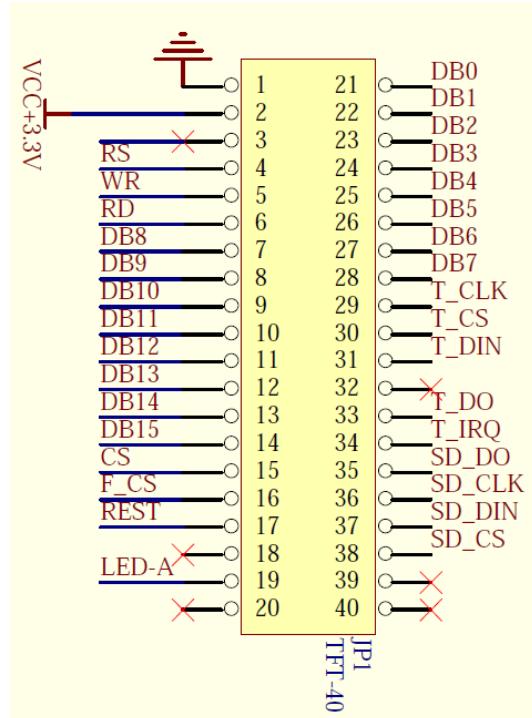


Abbildung 3.1.: 8080-Display Pinout, Quelle: [Coldtears Electronics](#)

besitzen bei 4.3“ eine Auflösung von 480x272 beziehungsweise bei 5“ 800x480 Pixeln. Neben den für die Initialisierung nötigen Kommandos besitzt der Controller folgende wichtigen Kommandos. Diese sind in Tabelle 3.1 beschrieben. Die zur Initialisierung notwendigen Kommandos werden hier nicht erläutert, da diese aus dem Datenblatt entnehmbar sind.

Kommando	Hex-Code	Kommentar
Set Column Address	0x2A	Eckpunkte des RAM-Fensters in X-Richtung
Set Page Address	0x2B	Eckpunkte des RAM-Fensters in Y-Richtung
Write Memory Start	0x2C	Alle Folgenden Pixeldaten werden im RAM-Fenster platziert

Tabelle 3.1.: Relevante Kommandos des SSD1963, [Solomon Systech Limited \[2008\]](#)



3. Teil A

3.1.2. 3.2“ mit SSD1289

Das 3.2“ Display von Sainsmart wird mit einen SSD1289 von Solomon Systech betrieben. Dieses Display hat eine Aufloesung ovn 320x240 Farbpunkten. Das Pinout ist dasselbe, das in Abbildung 3.1MPMCStatic zu sehen ist. Analog zu Tabelle 3.1 besitzt der SSD1289 seine eigenen wichtigen Kommandos. Diese sind in Tabelle 3.2 erlaeutert. Die zur Initialisierung notwendigen Kommandos werden hier nicht erläutert, da diese aus dem Datenblatt entnehmbar sind.

Kommando	Hex-Code	Kommentar
Horizontal RAM address position	0x44	Eckpunkte des RAM-Fensters in X-Richtung
Vertical RAM address start position	0x45	Startpunkt des RAM-Fensters in Y-Richtung
Horizontal RAM address stop position	0x46	Endpunkt des RAM-Fensters in Y-Richtung
Set GDDRAM X address counter	0x4E	Zeiger im RAM-Fenster in X-Richtung
Set GDDRAM Y address counter	0x4F	Zeiger im RAM-Fenster in Y-Richtung
RAM Write Register	0x22	Alle Folgenden Pixeldaten werden im RAM-Fenster platziert

Tabelle 3.2.: Relevante Kommandos des SSD1289, [Solomon Systech Limited \[2007\]](#)

3.1.3. 5“ mit CPLD

Als drittes Display mit 8080-Interface kommt eine 5“ Display mit einer Aufloesung von 800x480 Bildpunkten zum Einsatz, dass keinen univerell einsetzbaren Controller für variable Displaypanels im klassischen Sinn besitzt, sondern ein CPLD ¹⁹ als Controller mit zugeschnittenen Timings für das verwendete TFT-Panel. Der Vorteil eines solchen Displays ist, dass keine Initialisierungsroutine benötigt wird, um die Timings für das Panel einzustellen. Ein Reset setzt das Display betriebsbereit. Nachteilig stellt sich der Umstand ein, dass nur TFT-Panels exakter Größe und mit exakten Timings verwendet werden können. Für diese Arbeit ist allerdings die Verwendung von anderen Panels belanglos. Auch hier ist das Pinout des Displays analog zu dem Gezeigten in Abbildung 3.1.

Wichtige Kommandos zum Betrieb des Displays sind in Tabelle 3.3 einsehbar.

¹⁹CPLD: Complex Programmable Logic Device



-
3. Teil A

Kommando	Hex-Code	Kommentar
Beginning Row Address	0x02	Startpunkt des RAM-Fensters in X-Richtung
Ending Row Address	0x06	Endpunkt des RAM-Fensters in X-Richtung
Beginning Column Address	0x03	Startpunkt des RAM-Fensters in Y-Richtung
Ending Column Address	0x07	Endpunkt des RAM-Fensters in Y-Richtung
Writing Page Register	0x05	Alle Folgenden Pixeldaten werden im RAM-Fenster platziert

Tabelle 3.3.: Relevante Kommandos des MD050SD, [ITEAD Studios \[2013\]](#)

3.2. 8080-Interface mittels SRAM-Interface

Wie bereits in Abschnitt [2.2.2](#) erwähnt, besitzt der Prozessor des Gnublin bereits ein externes 8080-Interface, auf welches zugegriffen wird. Im Folgenden wird auf das Konzept, die Idee und die Realisierung auf Hardware- und Softwareseite eingegangen.



3. Teil A

3.2.1. Konzept

Im Gnublin stellt ein NXP LPC313x die zentrale Recheneinheit dar. Dieser besitzt ein sogenanntes EBI²⁰, worüber externe Bausteine wie Speicher, Ethernetcontroller oder ähnliche Bausteine angesprochen werden können.

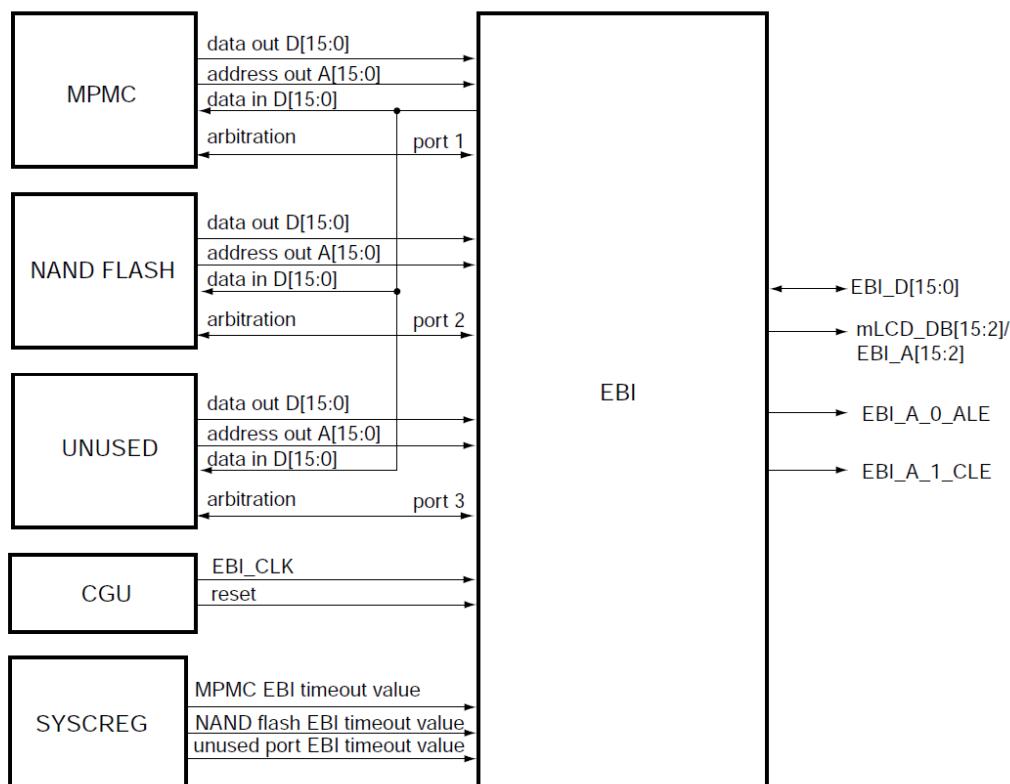


Abbildung 3.2.: NXP LPC313x EBI, Quelle: [NXP Semiconductors \[2010\]](#)

In Abbildung 3.2 ist ein Blockschaltbild des EBI zu sehen, bei welchem neben CGU²¹ und SYSCREG²², MPMC²³ sowie das NAND Flash an den Eingängen des EBI angeschlossen sind. An den Ausgängen des EBI sind Adress- und Datenbus zum Anschluss an externe Bausteine herausgeführt. Damit verschiedenartigen Bausteine an denselben Adress- und Datenpins angeschlossen werden kann, ist eine Priorisierung notwendig. Die Höchste Priorität besitzt der MPMC, gefolgt vom NAND Flash. Die Grundidee ist, das Display über den MPMC anzuschließen, da er so konfiguriert werden kann, dass er sich 8080-konform verhält.

²⁰EBI: External Bus Interface

²¹CGU: Clock Generation Unit, Takterzeugung

²²SYSCREG: System Control Register, Steuerregister

²³MPMC: Multiport Memory Controller



-
3. Teil A

3.2.2. MPMC - Multiport Memory Controller des NXP LPC313x

Der MPMC stellt die Möglichkeit zur Verfügung Bausteine wie dynamisches und statisches RAM anzubinden. Die Refresh-Zyklen werden bei Verwendung von dynamischen RAMs automatisch vollzogen. Das SDRAM-Interface bietet von Haus aus die Möglichkeit Displays mit 8080-Interface zu betreiben. Dies schließt allerdings die Verwendung von dynamischen RAMs aus. Soll ein Betriebssystem wie Linux laufen, ist allerdings die Verwendung von dynamischem RAM unerlässlich. Im Folgenden wird die Schnittstelle für das statische RAM SRAM-Interface benannt. Es besteht die Möglichkeit das Interface des statischen RAM zu verwenden, um ein Display zu betreiben, da es sich so konfigurieren lässt, dass es sich wie ein 8080-Interface verhält. Damit sich die verschiedenen Slaves an Adress- und Datenbus nicht überschneiden, regelt das EBI den Zugriff auf die Busse über Chip-Select Leitungen. Am Gublin ist eine dieser Chip-Select-Leitungen fuer das SRAM-Interface nach außen gelegt. Die restlichen Anschlüsse wie Write-Enable, Read-Enable sind ebenfalls herausgeführt [NXP Semiconductors \[2010\]](#).



3. Teil A

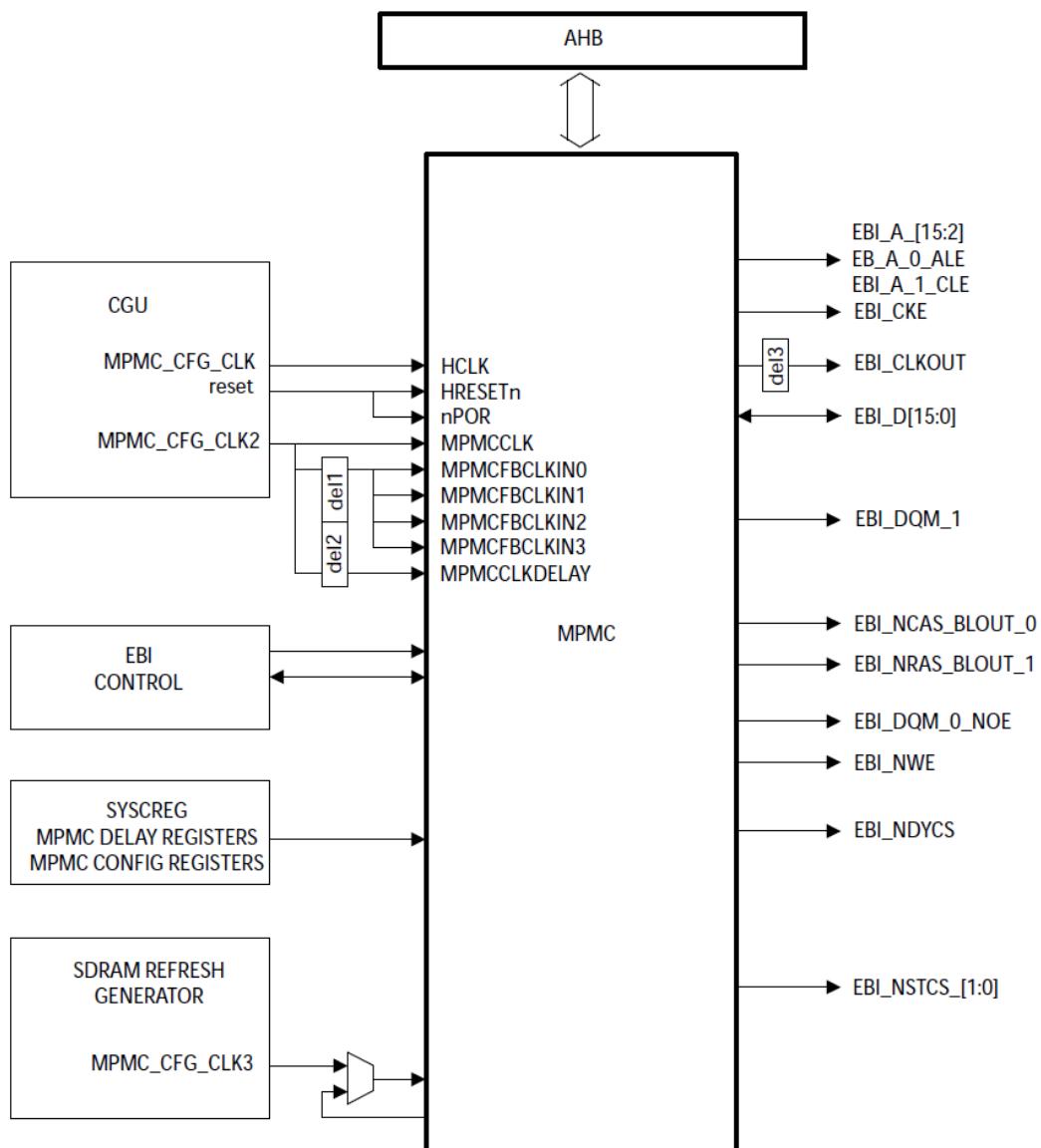


Abbildung 3.3.: NXP LPC313x MPMC, Quelle:



3. Teil A

Die Register des MPMC werden so konfiguriert, dass die Schnittstelle kompatibel zum Display und dessen Timings wird. Entsprechend dem verwendeten Chip-Select-Signal werden die Register MPMCStaticConfig0, MPMCStaticWaitWen0, MPMCStaticWaitOen0, MPMCStaticRd0, MPMCStaticPage0, MPMCStaticWr0 und MPMCStaticWaitTurn0 entsprechend Tabelle 3.4 konfiguriert. Die Basisadresse des MPMC ist 0x1700 8000.

Register	Offset	Wert	Beschreibung
MPMCStaticConfig0	0x200	0x81	16 Bit Modus, Aktiviert die Nutzung von EBI_nWE
MPMCStaticWaitWen0	0x204	13	$13 + 1 = 14$ Wartezyklen ab Chip-Select bis Write-Enable
MPMCStaticWaitOen0	0x208	0	$0 + 1 = 1$ Wartezyklus ab Chip-Select bis Output-Enable
MPMCStaticRd0	0x20C	0	$0 + 1 = 1$ Wartezyklus ab Chip-Select bis Read-Enable
MPMCStaticPage0	0x210	0	$0 + 1 = 1$ Wartezyklus für sequential Page Mode Access
MPMCStaticWr0	0x214	15	$15 + 2 = 17$ Wartezyklen bis Write-Access
MPMCStaticWaitTurn0	0x218	0	$0 + 1 = 1$ Turnaround Cycles

Tabelle 3.4.: MPMC Register

3.2.3. Hardwareverbindung zwischen SRAM-Interface und Display (Adapterplatine)

3.2.4. Software

3.2.4.1. Entwicklung eines Linux-Framebuffer-Treibers

3.2.4.1.1. Anpassungen für Display mit SSD1289 Controller

3.2.4.1.2. Anpassungen für Display mit SSD1963 Controller

3.2.4.1.3. Anpassungen für Display mit CPLD Controller

3.2.4.2. Entwicklung eines User-Space-Treibers

3.2.4.2.1. Anpassungen für Display mit SSD1289 Controller

3.2.4.2.2. Anpassungen für Display mit SSD1963 Controller



-
3. Teil A

3.2.4.2.3. Anpassungen für Display mit CPLD Controller

3.2.4.3. Anpassung des APEX-Bootloaders zur Verwendung des Displays

3.2.4.4. Probleme bei der Entwicklung und Fehlersuche

3.2.4.4.1. Probleme mit SSD1963

3.2.4.4.1.1. Rolle des User-Space-Treibers

3.2.4.4.1.2. Debuggen mit Logik-Analyzer

3.3. Known Bugs

3.3.1. Known Bugs SSD1963

3.3.2. Known Bugs SSD1289

3.3.3. Known Bugs CPLD Display



4. Teil B

4.1. Konzept

4.2. Hardwareentwicklung

4.2.1. Spannungsversorgung

4.2.2. HDMI-RGB-Bridge

4.2.3. RGB-LVDS-Bridge

4.2.4. EDID-Daten

4.3. Software

4.3.1. EDID-Daten auf embedded Seite

4.3.1.1. Konzept

4.3.1.2. Low-Level-Treiber

4.3.1.2.1. UART-Treiber

4.3.1.2.2. I2C-Treiber



-
4. Teil B

4.3.1.3. Programmablauf

4.3.2. EDID-Daten auf PC Seite

4.3.2.1. Konzept

4.3.2.2. GTK GUI mit Glade

4.3.2.3. Programmablauf

4.4. Known Bugs

4.4.1. Hardware

4.4.1.1. HDMI-Stecker gekreuzt, CON2

4.4.1.2. LVDS-Steckerfootprint gespiegelt, CON6

4.4.1.3. +12V PWM Hintergrundbeleuchtung

4.4.1.4. +5V Kreis / Widerstand R13

4.4.1.5. USB D+/D- vertauscht R22, R23

4.4.2. Software

4.4.2.1. EDID Programmer



-
5. Zusammenfassung

5. Zusammenfassung



Literaturverzeichnis

Literaturverzeichnis

Brandt 2013

BRANDT, Matthias: *Fast 80 Prozent Marktanteil für Android.* <http://de.statista.com/themen/581/smartphones/infografik/1326/smartphone-absatz-weltweit>. Version: 2013, Abruf: 20.05.2014 **1.1**

Coldtears Electronics

COLDTEARS ELECTRONICS, CE: *Schematic for 5"Display with SSD1963.* https://github.com/siredmar/master/raw/master/Recherche/Displays/8080/5Inch_SSD1963/schematic.pdf, Abruf: 28.07.2014 (**document**), **3.1**

Extron 2014

EXTRON: *DVI and HDMI: The Short and the Long of It.* http://www.extron.com/company/article.aspx?id=dvihdmi_ts. Version: 2014, Abruf: 20.05.2014 **2.1.3**

ITEAD Studios 2013

ITEAD STUDIOS, ITEAD: *MD070SD Datasheet.* https://github.com/siredmar/master/raw/master/Recherche/Displays/8080/5Inch_MD050SD/DS_IM130820001.pdf. Version: 2013, Abruf: 22.05.2014 (**document**), **3.3**

Knuppfer 2010

KNUPPFER, Nick: *Leading PC Companies Move to All Digital Display Technology, Phasing out Analog.* http://newsroom.intel.com/community/intel_newsroom/blog/2010/12/08/leading-pc-companies-move-to-all-digital-display-technology-phasing-out-analog?cid=rss-258152-c1-262653. Version: 2010, Abruf: 20.05.2014 **2.1.1**

Leunig 2002

LEUNIG, Peter H.: *Der DVI-Standard - ein Überblick.* http://www.leunig.de/_pro/downloads/DVI_WhitePaper.pdf. Version: 2002, Abruf: 20.05.2014 **2.1.2**

NXP Semiconductors 2010

NXP SEMICONDUCTORS, NXP: *LPC3130/31 User Manual.* <https://github.com/siredmar/master/raw/master/Recherche/Gnublin/datasheets/user.manual.lpc3130.lpc3131.pdf>. Version: 2010, Abruf: 28.07.2014 (**document**), **3.2, 3.2.2**



Literaturverzeichnis

Schlegel 2013

SCHLEGEL, Armin: *Ansteuerung eines TFT-Displays mit dem Raspberry Pi über die GPIO-Pins.* https://github.com/siredmar/siredmar_projects/raw/master/embedded/RPi/RPI_SSD1963/doc/RPI_SSD1963_24_11_2013.pdf.

Version: 2013, Abruf: 28.07.2014 [3.1.1](#)

Solomon Systech Limited 2007

SOLOMON SYSTECH LIMITED, SSD: *SSD1289 Datasheet.* <http://www.kosmodrom.com.ua/el/STM32-TFT/SSD1289.pdf>. Version: 2007, Abruf: 22.05.2014 ([document](#)), [2.1.6](#), [2.3](#), [3.2](#)

Solomon Systech Limited 2008

SOLOMON SYSTECH LIMITED, SSD: *SSD1963 Datasheet.* https://github.com/siredmar/master/raw/master/Recherche/Displays/8080/5Inch_SSD1963/SSD1963datasheet.pdf. Version: 2008, Abruf: 22.05.2014 ([document](#)), [3.1](#)

Texas Instruments 2011

TEXAS INSTRUMENTS, TI: *TI PanelBus™ DIGITAL RECEIVER - TFP401A-EP.* www.ti.com/litv/slds160a. Version: 2011, Abruf: 22.05.2014 ([document](#)), [2.2](#)

Valcarce 2011

VALCARCE, Javier: *VGA Video Signal Format and Timing Specifications.* http://www.javiervalcarce.eu/wiki/VGA_Video_Signal_Format_and_Timing_Specifications. Version: 2011, Abruf: 22.05.2014 ([document](#)), [2.1.1](#), [2.1](#)



Eidesstattliche Erklärung

Eidesstattliche Erklärung

Ich, Armin Schlegel, Matrikel-Nr. 2020863, versichere hiermit, dass ich meine Masterarbeit mit dem Thema

Entwicklung und Optimierung von Display-Schnittstellen fuer embedded Linux Boards -

selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, wobei ich alle wörtlichen und sinngemäßen Zitate als solche gekennzeichnet habe. Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Mir ist bekannt, dass ich meine Masterarbeit zusammen mit dieser Erklärung fristgemäß nach Vergabe des Themas in dreifacher Ausfertigung und gebunden im Prüfungsamt der Ohm-Hochschule abzugeben oder spätestens mit dem Poststempel des Tages, an dem die Frist abläuft, zu senden habe.

Nuernberg, den 28. Juli 2014

ARMIN SCHLEGEL



A. Anhang