

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

Studienbereich: Fakultät 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, consectetur 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, consectetur 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, consectetur 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	3
2.1.3. HDMI	5
2.1.4. TV-OUT	5
2.1.5. LVDS	5
2.1.6. 8080-Interface	5
2.1.7. Verwendung der SPI-Schnittstelle	5
2.1.8. Bewertung der Video-Schnittstellen	5
2.2. Betrachtete Embedded Linux Boards	5
2.2.1. Raspberry Pi	5
2.2.2. GnuBLIN Extended	5
2.2.3. Bewertung der Linux-Boards	5
3. Teil A	6
3.1. Display mit 8080-Interface	6
3.2. 8080-Interface mittels GPIO-Pins	6
3.2.1. Konzept	6
3.2.2. Hardwareverbindung zwischen GPIO-Pins und Display	6
3.2.3. User-Space-Treiber	6
3.2.3.1. Low-Level-Treiber	6
3.2.3.1.1. GPIO-Pin Frequenz erhöhen	6
3.2.3.1.2. GPIO-Treiber	6



3.2.3.1.3.	Displaytreiber für SSD1963	6
3.2.4.	Ansteuerung des Displays	7
3.3.	8080-Interface mittels SRAM-Interface	7
3.3.1.	Konzept	7
3.3.2.	MPMC - Multiport Memory Controller des NXP LPC313x	7
3.3.3.	Hardwareverbindung zwischen SRAM-Interface und Display (Adapterplatine)	7
3.3.4.	Untersuchte Displays	7
3.3.4.1.	3.2"mit SSD1289	7
3.3.4.2.	4.3"/5"mit SSD1963	7
3.3.4.3.	5"mit CPLD	7
3.3.5.	Software	7
3.3.5.1.	Entwicklung eines Linux-Framebuffer-Treibers	7
3.3.5.1.1.	Anpassungen für Display mit SSD1289 Con- troller	7
3.3.5.1.2.	Anpassungen für Display mit SSD1963 Con- troller	7
3.3.5.1.3.	Anpassungen für Display mit CPLD Con- troller	7
3.3.5.2.	Entwicklung eines User-Space-Treibers	7
3.3.5.2.1.	Anpassungen für Display mit SSD1289 Con- troller	7
3.3.5.2.2.	Anpassungen für Display mit SSD1963 Con- troller	7
3.3.5.2.3.	Anpassungen für Display mit CPLD Con- troller	7
3.3.5.3.	Anpassung des APEX-Bootloaders zur Verwendung des Displays	8
3.3.5.4.	Probleme bei der Entwicklung und Fehlersuche	8
3.3.5.4.1.	Probleme mit SSD1963	8
3.3.5.4.1.1.	Rolle des User-Space-Treibers	8
3.3.5.4.1.2.	Debuggen mit Logik-Analyzer	8
3.3.5.4.1.3.	Lösung des Problems	8
3.4.	Vor- und Nachteile	8
4.	Teil B	9
4.1.	Konzept	9
4.2.	Hardwareentwicklung	9
4.2.1.	Spannungsversorgung	9
4.2.2.	HDMI-RGB-Bridge	9



Inhaltsverzeichnis

4.2.3. RGB-LVDS-Bridge	9
4.2.4. EDID-Daten	9
4.3. Software	9
4.3.1. EDID-Daten auf embedded Seite	9
4.3.1.1. Konzept	9
4.3.1.2. Low-Level-Treiber	9
4.3.1.2.1. UART-Treiber	9
4.3.1.2.2. I2C-Treiber	9
4.3.1.3. Programmablauf	10
4.3.2. EDID-Daten auf PC Seite	10
4.3.2.1. Konzept	10
4.3.2.2. GTK GUI mit Glade	10
4.3.2.3. Programmablauf	10
5. Zusammenfassung	11
Literaturverzeichnis	12
Eidesstattliche Erklärung	13
A. Anhang	i



Abbildungsverzeichnis

2.1. Quelle: DVI-Timing, Wikipedia [2010]	4
---	---



Tabellenverzeichnis



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 Betriebssysteme 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.



1.3. 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 leistungstärkere 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

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 dargestellt.

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 Chiphersteller will ab 2015 auf die Schnittstelle verzichten [Knuppfer \[2010\]](#) und digitalen Schnittstellen den Vorzug lassen. Zwar ist die VGA-Schnittstelle noch nicht komplett wegrationalisiert, 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.

2.1.2. DVI

Hinter DVI steht der Begriff Digital Visual Interface und stellt eine digitale Schnittstelle zur Grafikanzeige dar. Der DVI Standard wurde 1999 von der DDWG¹ verabschiedet, da der Wunsch nach leistungstärkeren Schnittstellen vorhanden war.

¹Digital Display Working Group



2. Theoretische Grundlagen

QXGA-Auflösungen² 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 TMDS³ Standard verwendet, welcher die 24 Bit Farbinformationen⁴ 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/s⁵ bzw. 7.44 GBit/s⁶ übertragen. Um die Paare zuordnen zu können, wird ein weiteres Paar zur Synchronisation verwendet. Um die Übertragung noch effizienter zu gestalten, gibt es die Möglichkeit bei High- sowie Low-Pegel des Taktsignals Daten zu übertragen. Dieser Modus heisst Double Data Rate [Leunig \[2002\]](#).

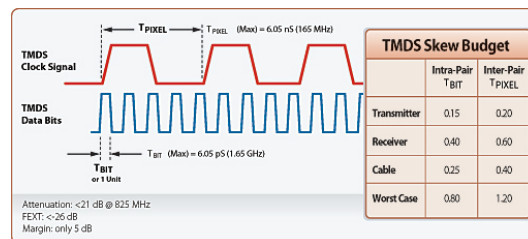


Abbildung 2.1.: Quelle: DVI-Timing, [Wikipedia \[2010\]](#)

zeigt ein Timing Diagramm, welches die Datenübertragung darstellt. Es sind

²QXGA: 2048x1536

³Transition Minimized Differential Signaling - Differentielle Datenübertragung

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

⁵max. UXGA: 1600x1200@60Hz

⁶max. WUXGA: 1920x1200@60Hz



2.1.3. HDMI

2.1.4. TV-OUT

2.1.5. LVDS

2.1.6. 8080-Interface

2.1.7. Verwendung der SPI-Schnittstelle

2.1.8. Bewertung der Video-Schnittstellen

2.2. Betrachtete Embedded Linux Boards

2.2.1. Raspberry Pi

2.2.2. Gnublin Extended

2.2.3. Bewertung der Linux-Boards



3. Teil A

3.1. Display mit 8080-Interface

3.2. 8080-Interface mittels GPIO-Pins

3.2.1. Konzept

3.2.2. Hardwareverbindung zwischen GPIO-Pins und Display

3.2.3. User-Space-Treiber

3.2.3.1. Low-Level-Treiber

3.2.3.1.1. GPIO-Pin Frequenz erhöhen

3.2.3.1.2. GPIO-Treiber

3.2.3.1.3. Displaytreiber für SSD1963



3.2.4. Ansteuerung des Displays

3.3. 8080-Interface mittels SRAM-Interface

3.3.1. Konzept

3.3.2. MPMC - Multiport Memory Controller des NXP LPC313x

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

3.3.4. Untersuchte Displays

3.3.4.1. 3.2" mit SSD1289

3.3.4.2. 4.3"/5" mit SSD1963

3.3.4.3. 5" mit CPLD

3.3.5. Software

3.3.5.1. Entwicklung eines Linux-Framebuffer-Treibers

3.3.5.1.1. Anpassungen für Display mit SSD1289 Controller

3.3.5.1.2. Anpassungen für Display mit SSD1963 Controller

3.3.5.1.3. Anpassungen für Display mit CPLD Controller

3.3.5.2. Entwicklung eines User-Space-Treibers

3.3.5.2.1. Anpassungen für Display mit SSD1289 Controller

3.3.5.2.2. Anpassungen für Display mit SSD1963 Controller

3.3.5.2.3. Anpassungen für Display mit CPLD Controller



3. Teil A

3.3.5.3. Anpassung des APEX-Bootloaders zur Verwendung des Displays

3.3.5.4. Probleme bei der Entwicklung und Fehlersuche

3.3.5.4.1. Probleme mit SSD1963

3.3.5.4.1.1. Rolle des User-Space-Treibers

3.3.5.4.1.2. Debuggen mit Logik-Analyzer

3.3.5.4.1.3. Lösung des Problems

3.4. Vor- und Nachteile



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



5. Zusammenfassung



Literaturverzeichnis

Brandt 2013

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

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](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](http://www.leunig.de/_pro/downloads/DVI_WhitePaper.pdf). Version: 2002, Abruf: 20.05.2014 2.1.2

Wikipedia 2010

WIKIPEDIA: *Timing Diagramm des DVI Standards.* [http://commons.
wikimedia.org/wiki/File:Dvi-timing-serializer-function.png](http://commons.wikimedia.org/wiki/File:Dvi-timing-serializer-function.png).
Version: 2010, Abruf: 20.05.2014 (document), 2.1



Eidesstattliche Erklärung

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

*Entwicklung und Optimierung von Display-Schnittstellen für 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 20. Mai 2014

ARMIN SCHLEGEL



A. Anhang