Entwicklung und Optimierung von Display-Schnittstellen fuer embedded Linux Boards

-

Masterarbeit

im Fachgebiet Hard- und Softwareentwicklung

vorgelegt von:	Armin Schlegel
Studienbereich:	Fakultaet EFI
Matrikelnummer:	2020863
Erstgutachter:	Prof. Dr. Joerg Arndt
Zweitgutachter:	Peter Meier

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

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 eingebetten 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 eingebetten 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 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

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 Chiphersteller 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, HSYNC[1] und VSYNC[2]. 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.

[1] HSYNC: Horizontale Synchronisation
[2] VSYNC: Vertikale Synchronisation

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]).

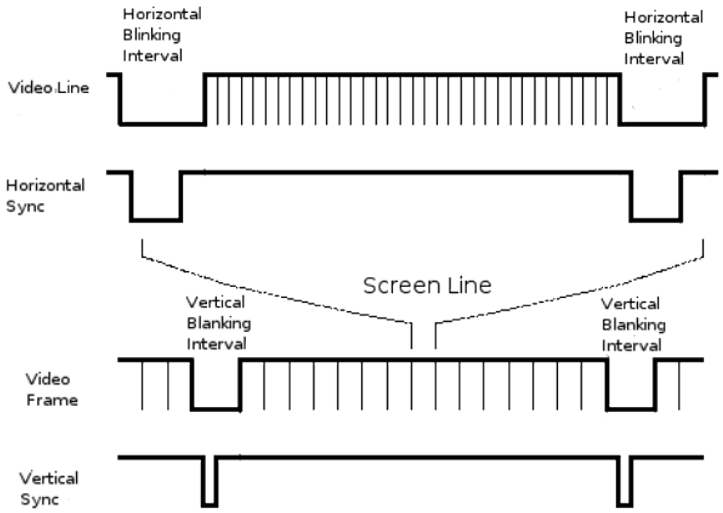

Abbildung 2.1.: VGA-Timing, Quelle: Valcarce [2011]

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 DDWG[3] verabschiedet, da der Wunsch nach Leistungsstärkeren Schnittstellen vorhanden war. QXGA-Auflösungen[4] 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 TMDS[5] Standard verwendet, welcher die 24 Bit Farbinformationen[6] 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/s[7] bzw. 7.44 GBit/s[8] übertragen. Um die Paare zuordnen zu können, wird ein weiteres Paar zur Synchronisation verwendet. Um die Übertragung

[3] Digital Display Working Group
[4] QXGA: 2048x1536
[5] Transition Minimized Differential Signaling - Differentielle Datenübertragung
[6] 24 Bit: je 8 Bit für Rot, Grün und Blau
[7] max. UXGA: 1600x1200@60Hz
[8] max. WUXGA: 1920x1200@60Hz

noch effizienter zu gestalten, gibt es die Möglichkeit bei High- sowie Low-Pegel des Taktsignals Daten zu übertragen[9] (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 16777216 [10] Farben zu erhalten. Dieser Farbmodus wird auch RGB888 genannt, da acht Bit für jede Far-

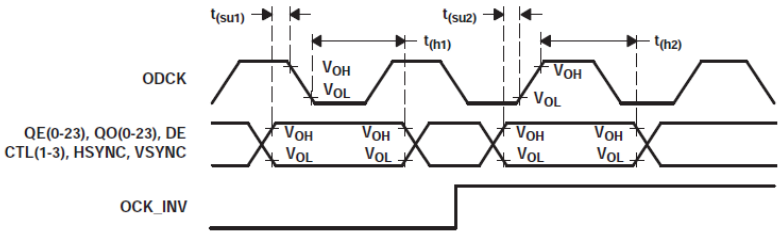

Abbildung 2.2.: RGB-Timing, Quelle: Instruments [2011]

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 65536[11] Farben. Da digital übertragen wird, ist eine Taktleitung notwendig um die Synchronizität zu ermöglichen. Aufgrund der Verbreitung und Mächtigkeit der Schnittstelle besitzen

[9] *Double Data Rate*
[10] 16777216 Farben = 2^24
[11] 65536 = 2^16

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. LVDS[12] zu übertragen. Die physikalische Funktionsweise der LVDS Leitung liegt darin, dass zweimal dasselbe 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

[12] LVDS: Low Voltage Differential Signaling

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 (Limited [2007]).

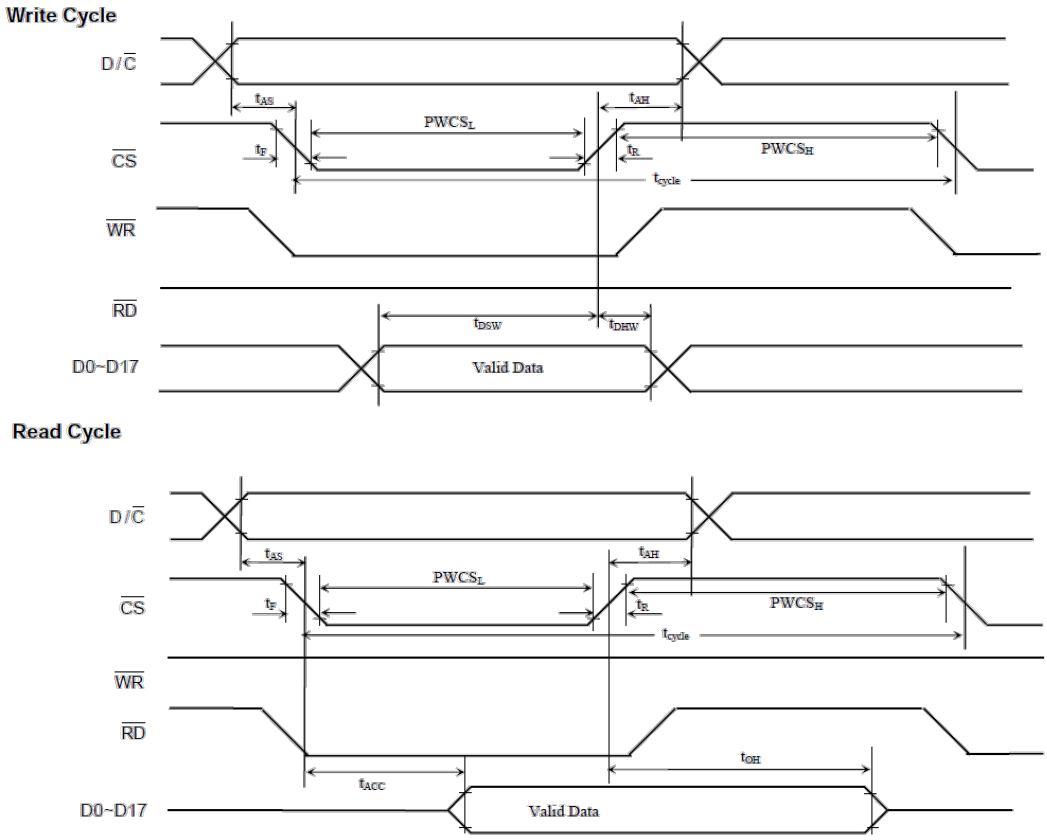

Abbildung 2.3.: 8080-Timing des SSD1289, Quelle: Limited [2007]

Viele Mikrocontroller besitzen bereits ein 8080-Interface dediziert in Hardware - allerdings nicht alle. Als Ersatz kann das Protokoll mit GPIO[13] 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.

[13]GPIO: General Purpose In/Output

2.1.7. Verwendung der SPI-Schnittstelle

2.1.8. Bewertung der Video-Schnittstellen

Die VGA-Schnittstelle ist für diese Masterarbeit uninterressant und wird daher nicht näher behandelt. Mittels Adapter lässt sich der DVI-Steckernach HDMI wandeln, weshalb die DVI-Schnittstelle für diese Masterarbeit Relevanz besitzt. Die RGB-Schnittstelle besitzt eine hohe Relevanz für diese Masterarbeit, da im Weiteren eine Platine entwickelt wird, welche diese Schnittstelle nutzt.

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

3.3.5.3. 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

4.3.1.3. 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, Mathtias: *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

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

Instruments 2011

Instruments, Texas: *TI PanelBus™ DIGITAL RECEIVER - TFP401A-EP*. www.ti.com/litv/slds160a. Version: 2011, Abruf: 22.05.2014 (document), 2.2

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

Limited 2007

Limited, Solomon S.: *SSD1289 Datasheet*. http://www.kosmodrom.com.ua/el/STM32-TFT/SSD1289.pdf. Version: 2007, Abruf: 22.05.2014 (document), 2.1.6, 2.3

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

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 22. Mai 2014

Armin Schlegel

A. Anhang