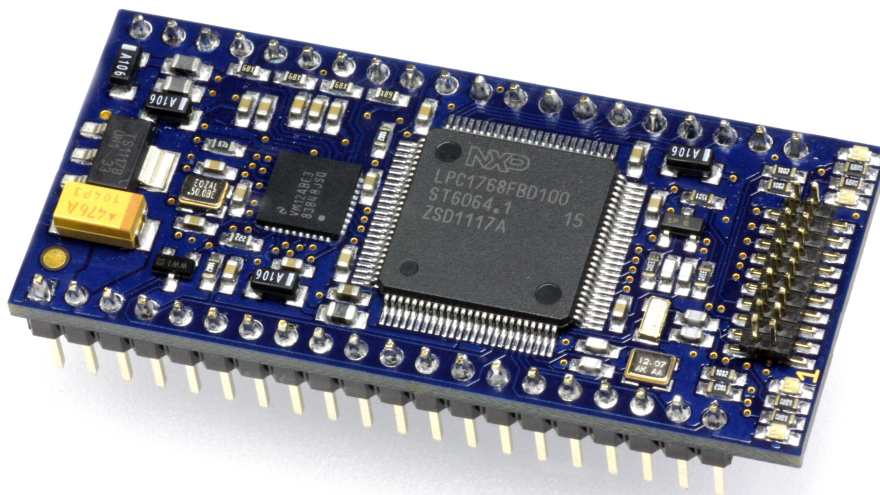


Chip1768

Hardware-Version V1.21



Mikrocontrollerboard mit NXP LPC1768 Cortex-M3

Benutzerhandbuch

Inhaltsverzeichnis

1	Überblick	4
1.1	Technische Daten	5
1.1.1	Chip1768-Modul	5
1.1.2	Mikrocontroller NXP LPC1768	5
2	Funktionen	6
2.1	Anschlussbelegung	7
2.2	Funktionstabelle	7
2.3	Unterschiede zum mbed	9
3	Schaltungsbeschreibung	10
3.1	Schaltplan	10
3.2	Stromversorgung	11
3.3	Debug- und Programmierschnittstelle	11
3.4	Takterzeugung	12
3.5	Ethernet	13
3.6	USB	14
3.7	PWM	15
3.8	Analog	15
3.9	Serielle Schnittstellen	15
3.10	RTC	16
4	Applikationshinweise	17
4.1	Der Bootloader	17
4.2	Anpassen von mbed-Programmen	18
4.2.1	Ethernetanwendungen	18
4.3	Verhalten nach Reset	18
4.4	SRAM	18
4.5	Nutzung in Verbindung mit KEIL MDK ARM	19
4.5.1	Voraussetzungen	19
4.5.2	Real-Time Trace	20
5	Anhang	22
5.1	Memory Map	22
5.2	Mechanische Abmessungen	23

Copyright (C)2011 by
ELMICRO Computer GmbH und Co. KG
Hohe Str. 9-13
D-04107 Leipzig
Telefon: +49-(0)341-9104810
Fax: +49-(0)341-9104818
Email: leipzig@elmicro.com
Web: <http://elmicro.com>

Dieses Handbuch wurde sorgfältig erstellt und geprüft. Trotzdem können Fehler und Irrtümer nicht ausgeschlossen werden. ELMICRO übernimmt keinerlei juristische Verantwortung für die uneingeschränkte Richtigkeit und Anwendbarkeit des Handbuchs und des beschriebenen Produktes. Die Eignung des Produktes für einen spezifischen Verwendungszweck wird nicht zugesichert. Die Haftung des Herstellers ist in jedem Fall auf den Kaufpreis des Produktes beschränkt. Eine Haftung für eventuelle Mangelfolgeschäden wird ausgeschlossen.

Produkt- und Preisänderungen bleiben, auch ohne vorherige Ankündigung, vorbehalten.

Die in diesem Handbuch erwähnten Software- und Hardwarebezeichnungen sind in den meisten Fällen auch eingetragene Warenzeichen und unterliegen als solche den gesetzlichen Bestimmungen. Es kann aus dem Fehlen einer besonderen Kennzeichnung nicht darauf geschlossen werden, daß die Bezeichnung ein freier Warenname ist. Gleiches gilt für Rechte aus Patenten und Gebrauchsmustern.

1 Überblick

Das Chip1768-Modul ermöglicht auf einfache Weise, von der Rechenleistung und Ausstattungsvielfalt des NXP LPC1768 Mikrocontrollers zu profitieren. Durch seine kompakte Bauform im 100mil-Standardraster kann das Modul schnell in eigene Projekte und Geräte eingesetzt werden. Durch seine Beschaltung erhält das Modul die nötige Robustheit und Flexibilität um einerseits als Rapid-Prototyping-Werkzeug eingesetzt zu werden und andererseits als endgültige Produktivhardware.

Beim verwendeten Mikrocontroller handelt es sich um einen NXP LPC1768 mit ARM Cortex-M3 (ARMv7M) Kern. Er verbindet die Vorteile der ARM Architektur wie weitreichende Debugging-Funktionen, Stromsparfeatures, Kompatibilität von Soft- und Hardwarewerkzeugen mit üppiger Peripherieausstattung aus dem Hause NXP.

Die Ähnlichkeit zum mbed1768-Modul von ARM/KEIL erklärt sich aus der Tatsache, dass Chip1768 vollständig¹ softwarekompatibel zum mbed1768 ist. Diese Tatsache eröffnet neue Möglichkeiten für den embedded Entwickler: Erste Funktionstests bzw. -demonstrationen („Proof-of-Concept“) werden mit dem mbed und dessen hervorragenden Bibliotheken im Onlinecompiler erstellt. Für das Produktdesign schließlich kann auf Chip1768 zurückgegriffen werden - wertvoller Programmcode kann also offline mit den herkömmlichen Werkzeugen (z.Bsp. Keils MDK ARM Entwicklungssoftware) geschrieben und dank Cortex-Debug-Schnittstelle auch höchst komfortabel debuggt werden!

Dieses Handbuch soll, neben der Erläuterung der Inbetriebnahme und Nutzung des Chip1768-Modules, einen Ausgangspunkt für die Softwareentwicklung mit ARMs 32 Bit Mikrocontrollern darstellen.

¹Eine kleine Änderung am Sourcecode für Ethernetanwendungen ist notwendig - siehe Kapitel 4.2.1 für nähere Erläuterungen.

1.1 Technische Daten

1.1.1 Chip1768-Modul

- Kompaktes Mikrocontrollermodul mit NXP LPC1768 MCU
- Cortex-Debug-Schnittstelle, 2x10 Stiftleiste mit 1,27mm Rastermaß, folgt ARMs Spezifikation des Cortex-Debug und -Trace Interface (Unterstützt JTAG, SWD, 4-Bit Trace)
- National Semiconductor DP83848J Ethernet Transceiver
- 12MHz und 32,786kHz Quarze
- 4 Signal-LEDs
- Spannungsbereich 4,5V .. 9V
- max. Stromaufnahme ca. 200mA (bei Versorgungsspannung von 5V und Nutzung der Ethernet-Schnittstelle)
- 40-polige DIP-Bauform
- 2,54mm Pinraster (100mil), 22,86mm Reihenabstand (900mil)
- Gesamtabmessungen: 56,50mm x 25,90mm

1.1.2 Mikrocontroller NXP LPC1768

- ARM Cortex-M3 (ARMv7) 32-Bit CPU
- bis 100MHz Haupttaktfrequenz
- 512kB Programmspeicher (Flash), 2x32kB RAM
- ARM Cortex Trace Unterstützung (Trace Port Unit)
- 10/100MBit Ethernet, RMI Interface, DMA-Controller
- 12-Bit Analog-Digital-Wandler mit 8 Kanälen
- 10-Bit Digital-Analog-Wandler (1 Kanal)
- 4 32-Bit breite Timer
- 6 PWM-Kanäle, 1x Motor Control PWM, 8 DMA-Kanäle
- USB 2.0 Interface mit integriertem Transceiver
- CAN2.0B mit 2 Kanälen
- 4x UART, 2x SSP, 1x SPI, 3x I²C, 1x I²S
- Interface für Quadratur-Encoder
- Low Power RTC, Unique ID
- interner 4MHz RC-Oszillator

2 Funktionen

Der NXP LPC1768 bietet im LQFP-Gehäuse 100 Anschlusspins. Davon sind 70 sog. „General Purpose Inputs/Outputs“ (GPIOs), also Funktionalpins. Durch die kompakte Modulbauform von Chip1768 mit seinen 40 Anschlussbeinchen musste eine Auswahl an herausgeführten GPIOs getroffen werden. Um kompatibel mit dem mbed1768 zu bleiben, wurden beim Chip1768 die meisten GPIOs genauso gewählt. Einige Änderungen wurden dennoch vorgenommen - sie finden sich im Unterkapitel 2.3.

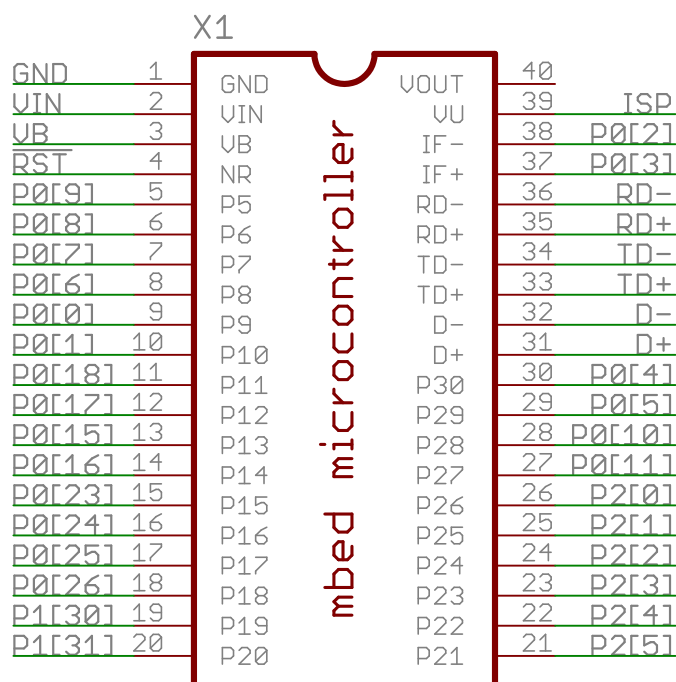
Die Möglichkeiten des Chip1768-Modules sind im Einzelnen:

- 28x GPIOs, davon 26 mit voller Interruptfähigkeit
- 2x I²C
- 1x duplexfähiges I²S
- 2x SSP, 1x SPI, 2x CAN
- 1x Full-Speed USB
- 1x 10/100MBit Ethernet
- 6x Match-Ausgänge für Timer (4x Timer2, 2x Timer3)
- 4x Capture-Eingänge für den Timer (je 2x für Timer 2 und 3)
- 4x UART (davon 1x mit der Möglichkeit, die Steuersignale DTR, DSR, DCD und CTS zu nutzen)
- 8x Eingänge für die Analog-Digital-Wandlung
- 1x Analogausgang des Digital-Analog-Wandlers
- 6x PWM-Ausgänge
- 1x Nichtmaskierbarer, externer Interrupteingang (NMI)
- Echtzeituhr (RTC) mit eigenem Spannungsversorgungseingang

Aufgrund der Funktionsvielfalt des verwendeten Mikrocontrollers kann in diesem Handbuch nicht auf die Programmierung der Peripheriebaugruppen eingegangen werden. Dem Entwickler und Nutzer von Chip1768 sei daher das Benutzerhandbuch² von NXP für den LPC1768 wärmstens ans Herz gelegt. Dieses stellt einen sehr verständlichen Begleiter für die Arbeit mit diesem Cortex-M3 Controller dar und erklärt detailliert und gut strukturiert die Inbetriebnahme der einzelnen Funktionen.

²Download unter: http://www.nxp.com/documents/user_manual/UM10360.pdf

2.1 Anschlussbelegung



Signal	Typ	Beschreibung
GND	-	Schaltungsmasse
VIN	Eingang	4,5 .. 9V
VB	Eingang	3V Batteriespannung für RTC
!RST	Eingang, low-aktiv	Haupt-Resetsignal für LPC1768
P5..P30	Ein-/Ausgang	Multifunktions-Ports, siehe Tabelle 2.2
D+	Ein-/Ausgang	USB-Datenleitung
D-	Ein-/Ausgang	USB-Datenleitung
TD+	Ausgang	Ethernet Sendeleitung, Positivsignal
TD-	Ausgang	Ethernet Sendeleitung, Negativsignal
RD+	Eingang	Ethernet Empfangsleitung, Positivsignal
RD-	Eingang	Ethernet Empfangsleitung, Negativsignal
IF+	Eingang	Empfangsleitung für Bootloader
IF-	Ausgang	Sendeleitung für Bootloader
ISP	Eingang	Steuereingang für den Bootloader-Modus

2.2 Funktionstabelle

In einem modernen Mikrocontroller wie dem LPC1768 herrscht, bedingt durch die immense Funktionsvielfalt, stets Gedränge an den Portpins. Die Lösung heißt Signal-Multiplexing. Den einzelnen GPIOs können also verschiedene Funktionen zugeordnet werden. Im Falle des vorliegenden Controller erlaubt diese Technik, jedem Pin eine von bis zu vier möglichen Funktionen zuzuordnen.

Die Funktion jedes einzelnen Pins kann über spezielle Register im Controller zur Programmlaufzeit oder bereits während der Initialisierung (im sog. „Startup-Script“) verändert werden.

Verantwortlich dafür sind die PINSEL_{xx}-Register der PINCON-Peripherie. Die Funktion jedes Pins wird durch jeweils zwei Bit im Register repräsentiert.

Die folgende Tabelle gibt Aufschluss darüber, welche Funktionen einzeln oder in Kombinationen nutzbar sind:

Pin	1. Funktion	2. Funktion	3. Funktion	4. Funktion	Pin am LPC1768
<i>P5</i>	P0[9]	I ² S (TX_SDA)	SSP1 (MOSI)	MAT2[3]	76
<i>P6</i>	P0[8]	I ² S (TX_WS)	SSP1 (MISO)	MAT2[2]	77
<i>P7</i>	P0[7]	I ² S (TX_CLK)	SSP1 (SCK)	MAT2[1]	78
<i>P8</i>	P0[6]	I ² S (RX_SDA)	SSP1 (SSEL)	MAT2[0]	79
<i>P9</i>	P0[0]	CAN1 (RD1)	UART3 (TX)	I ² C1 (SDA)	46
<i>P10</i>	P0[1]	CAN1 (TD1)	UART3(RX)	I ² C1 (SCL)	47
<i>P11</i>	P0[18]	UART1 (DCD)	SSP0 (MOSI)	SPI (MOSI)	60
<i>P12</i>	P0[17]	UART1 (CTS)	SSP0 (MISO)	SPI (MISO)	61
<i>P13</i>	P0[15]	UART1 (TXD)	SSP0 (SCK)	SPI (SCK)	62
<i>P14</i>	P0[16]	UART1 (RXD)	SSP0 (SSEL)	SPI (SSEL)	63
<i>P15</i>	P0[23]	A/D0[0]	I ² S (RX_CLK)	CAP3[0]	9
<i>P16</i>	P0[24]	A/D0[1]	I ² S (RX_WS)	CAP3[1]	8
<i>P17</i>	P0[25]	A/D0[2]	I ² S (RX_SDA)	UART3 (TXD)	7
<i>P18</i>	P0[26]	A/D0[3]	DAC out	UART3 (RXD)	6
<i>P19</i>	P1[30]	V _{BUS}	A/D0[4]		21
<i>P20</i>	P1[31]	SSP1 (SCK)	A/D0[5]		20
<i>P21</i>	P2[5]	PWM1[6]	UART1 (DTR)	TRACEDATA[0]	68
<i>P22</i>	P2[4]	PWM1[5]	UART1 (DSR)	TRACEDATA[1]	69
<i>P23</i>	P2[3]	PWM1[4]	UART1 (DCD)	TRACEDATA[2]	70
<i>P24</i>	P2[2]	PWM1[3]	UART1 (CTS)	TRACEDATA[3]	73
<i>P25</i>	P2[1]	PWM1[2]	UART1 (RXD)		74
<i>P26</i>	P2[0]	PWM1[1]	UART1 (TXD)		75
<i>P27</i>	P0[11]	UART2 (RXD)	I ² C2 (SCL)	MAT3[1]	49
<i>P28</i>	P0[10]	UART2 (TXD)	I ² C2 (SDA)	MAT3[0]	48
<i>P29</i>	P0[5]	I ² S (RX_WS)	CAN2 (TD)	CAP2[1]	80
<i>P30</i>	P0[4]	I ² S (RX_CLK)	CAN2 (RD)	CAP2[0]	81
<i>D+</i>	P0[29]	USB (D+)			29
<i>D-</i>	P0[30]	USB (D-)			30
<i>RX+</i> <i>RX-</i> <i>TX+</i> <i>TX-</i>	(Ethernet)				-
<i>IF+</i>	P0[2]	UART0 (TX)	A/D0[7]		98
<i>IF-</i>	P0[3]	UART0 (RX)	A/D0[6]		99
<i>ISP</i>	P2[10]	!EINT0	NMI		41

2.3 Unterschiede zum mbed

USB

Mbed weist zwei USB-Anschlüsse auf: einen „sichtbaren“ mit bestückter Mini-USB Buchse sowie einen weiteren, der auf den Stiftleisten herausgeführt wird (die Signale D+ und D-). Der erste Anschluss wird für das Herunterladen von neuen Programmen (*.BIN-Dateien aus dem Online-Compiler) genutzt sowie für die Kommunikation via virtueller COM-Schnittstelle. Außerdem ist es durch diesen Anschluss möglich, das mbed mit der Betriebsspannung zu versorgen. Eine direkte Nutzung dieses USB-Anschlusses durch die USB-Hardware im LPC1768 ist nicht möglich. Für diesen Zweck steht die zweite USB-Schnittstelle zur Verfügung.

Chip1768 hingegen besitzt nur eine USB-Schnittstelle. Diese ist direkt mit der USB-Peripherie des Mikrocontrollers verbunden. Programmcode wird über das Cortex-Debug Interface (JTAG oder SWD) auf den Controller geladen. Ebenso verzichtet Chip1768 auf das Bereitstellen von „Zwischenspeicher“ (mbed erlaubt das Speichern von bis zu 2MB Daten und wird vom PC als Massenspeichergerät erkannt).

Ausgangsspannung V_{OUT}

Mbed stellt am Pin V_{OUT} eine Spannung von 3,3V bereit. Diese kann als Versorgungsspannung in eigenen Schaltungen verwendet werden. Chip1768 verzichtet auf die Bereitstellung dieser Spannung. Das Signal V_{OUT} ist unbeschalten.

Durch den Verzicht auf die USB-Komfortfunktion des mbed liegt am Pin VU (mbed) bzw. ISP (Chip1768) keine Spannung an. Mbed stellt hier die 5V des USB-Hosts zur Verfügung.

Bootloader

Chip1768 ermöglicht die Nutzung des integrierten Bootloaders und nutzt hierfür die Pins ISP, IF+ und IF-. Weiterführende Informationen zu diesem Thema gibt Kapitel 4.1 auf Seite 17.

Analog-Digital Wandler (ADC)

An den beiden Pins IF+ und IF- werden die LPC1768-Signale P0[2] und P0[3] herausgeführt. Eine Sonderfunktion dieser Signale stellt die Nutzung als Eingänge für den A/D-Wandler dar. Somit lassen sich sämtliche 8 Kanäle der A/D-Peripherie nutzen (beim mbed sind nur 6 Kanäle nutzbar).

UART0

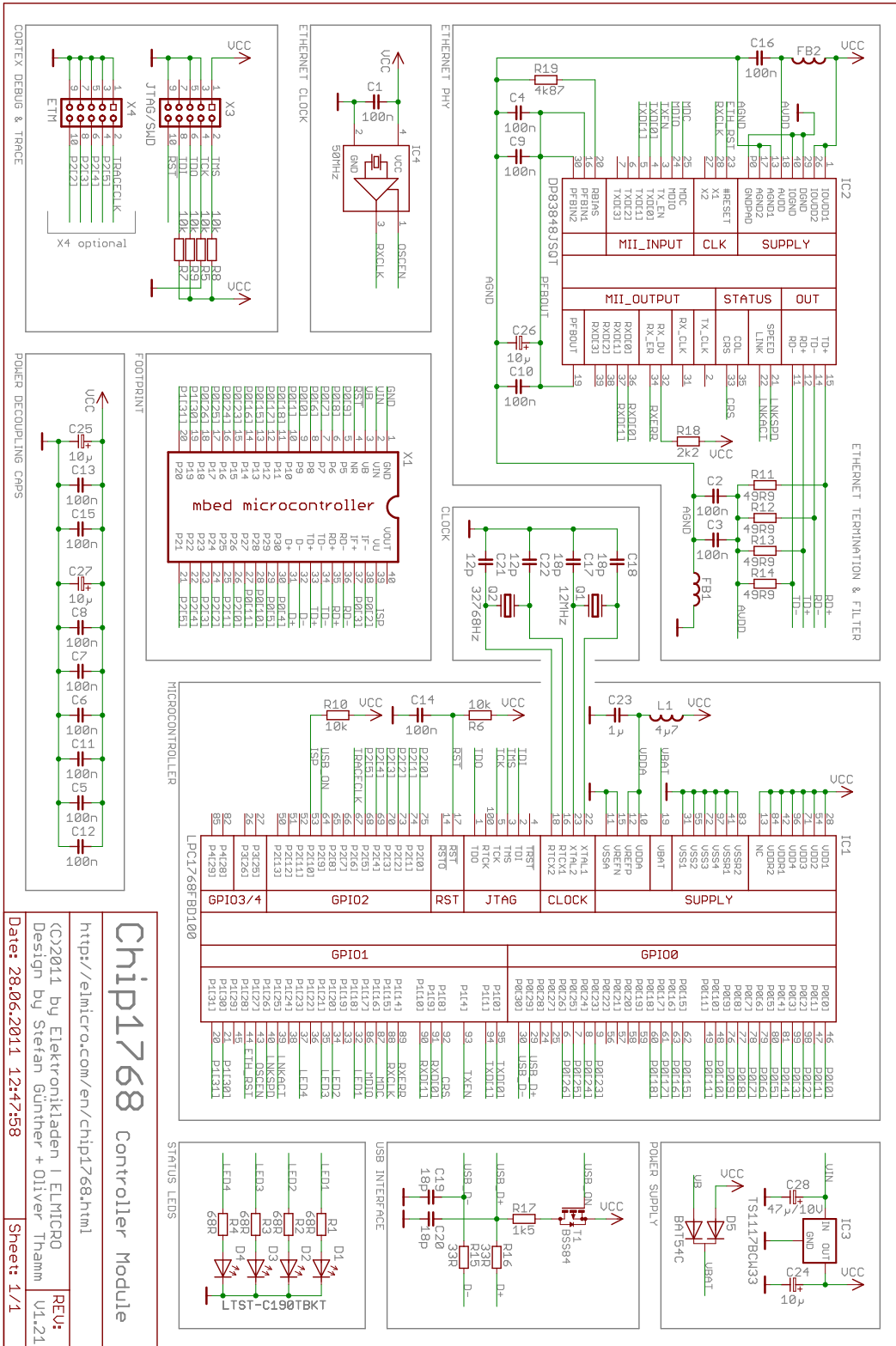
Eine weitere Sonderfunktion von IF+ und IF- ist die Nutzung als Datenleitungen der UART0-Peripherie des LPC1768. Damit lassen sich mit Chip1768 alle seriellen Schnittstellen (UART0 bis UART3) des LPC1768 nutzen. Mbed bietet hingegen nur die Nutzung von UART1 bis UART3.

NMI-Signal

Chip1768 bietet, im Gegensatz zum mbed, die Möglichkeit, den externen nichtmaskierbaren Interrupt (NMI) des LPC1768 zu verwenden. Das NMI-Signal wird an den Pin ISP herausgeführt.

3 Schaltungsbeschreibung

3.1 Schaltplan



Chip1768 Controller Module

<http://elmicro.com/en/chip1768.html>
(C)2011 by Elektronikladen | ELMICRO
Design by Stefan Günther + Oliver Thamm | V1.2.1
Date: 28.06.2011 12:47:58 Sheet: 1/1

3.2 Stromversorgung

Über den Eingangspin V_{IN} wird das Chip1768-Modul mit der notwendigen Betriebsspannung versorgt. Ein Linearregler sorgt für die Spannungswandlung nach 3,3V, die für den Betrieb des LPC1768 notwendig sind. Dieser Linearregler vom Typ TS1117B erlaubt eine Eingangsspannung von bis zu 12V³. Allerdings ist hier zu beachten, dass prinzipbedingt die entstehende Abwärme steigt, je höher die Eingangsspannung ist. Idealerweise wird V_{IN} mit 5V (Spannungsfest bis ca. 200mA) versorgt.

Die Eingangsspannung wird über einen Tantal-Kondensator mit 47 μ F gepuffert und speist dann den Linearregler. Dieser erzeugt die vom Chip1768 benötigte Versorgungsspannung von 3,3V. Im Schaltplan ist diese Spannungsdomäne mit VCC gekennzeichnet. An wichtigen Einspeisepunkten in die verschiedenen Bauelemente, vor allem an den ICs, wird VCC mit 10 μ F gepuffert. Somit wird einem Einbrechen der Versorgungsspannung bei plötzlichen Lastwechseln vorgebeugt. Zudem wird VCC mit 100nF-Kondensatoren an den entsprechenden Versorgungspins der ICs gegen Masse entkoppelt. Dies verringert evtl. vorhandene Störüberlagerungen in der Versorgungsspannung.

Achtung:

Die Eingangsspannung ist weder gegen Verpolung noch gegen Überspannung oder Kurzschluss abgesichert! Sie darf 10V auf keinen Fall überschreiten! Empfohlen wird eine Versorgungsspannung von 5V.

3.3 Debug- und Programmierschnittstelle

Das Programmieren und Debuggen der Applikation im Mikrocontroller kann wahlweise über die klassische JTAG-Verbindung oder über die „Serial Wire Debug“-Schnittstelle (SWD) erfolgen. Die JTAG-Unterstützung wurde von NXP allerdings lediglich aus Kompatibilitätsgründen beibehalten - es empfiehlt sich nach Möglichkeit, stets die SWD-Schnittstelle zu verwenden. Über SWD kann die „Data Trace“-Funktionalität des Cortex-M3 genutzt werden, was über JTAG nicht möglich ist.

Für zeit- und/oder ressourcenkritische Applikationen verfügt der LPC1768 zudem über ARMs „Embedded Trace Macrocell“-Technologie (ETM). Dies ermöglicht das sog. „Real-Time-Tracing“, bei dem jeder Prozessorschritt im Detail aufgezeichnet werden kann. Da der Prozessor dabei das Programm unterbrechungsfrei abarbeitet, erlaubt dieses Tracing sehr detaillierte Einblicke in den Programmablauf.

Für die Nutzung des Real-Time-Traces wird entsprechend ausgestattete Hard- und Software benötigt. Beispielhaft sei an dieser Stelle das Produkt von Keil erwähnt: Mit der Kombination aus ULINK Pro Adapter und der MDK-ARM Software ist es möglich, sämtliche erweiterten Debug-Features des LPC1768 (und die sämtlicher anderer ARM Cortex-M MCUs) zu benutzen.

Auf Chip1768 ist die „Cortex Debug+ETM“-Schnittstelle mittels der Stiftleiste X3/X4 zugänglich. Durch Vollbestückung nach dem ARM-Standard (2x10-polig) ist es möglich, von der „Cortex Trace“-Funktionalität zu profitieren.

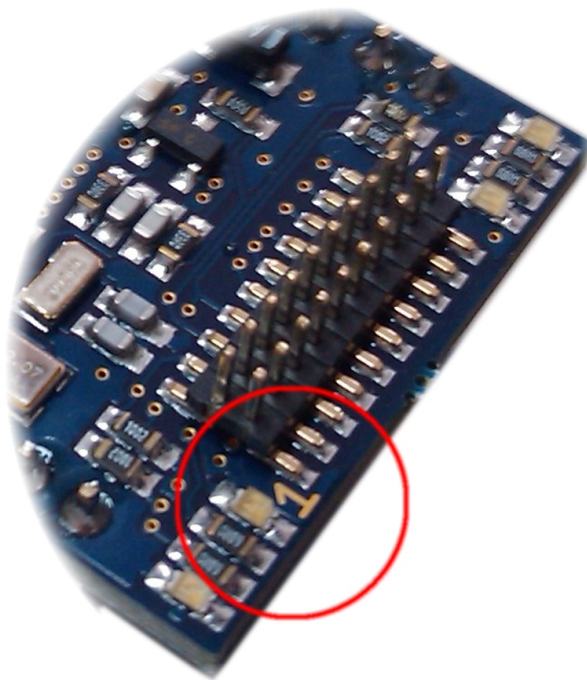
Das folgende Schema zeigt die von ARM spezifizierte Signaluordnung zur Debug-Schnittstelle:

³Der eingangsseitige Puffer-Elko C28 weist jedoch eine Spannungsfestigkeit von max. 10V auf!

Cortex-M ETM Interface
20-pin Connector

VTref	1	□ □	2	SDWIO / TMS
GND	3	□ □	4	SWDCLK / TCLK
GND	5	□ □	6	SWO / TDO
KEY	7	□	8	NC / TDI
GNDDetect	9	□ □	10	nRESET
GND/TgtPwr+Cap	11	□ □	12	TRACECLK
GND/TgtPwr+Cap	13	□ □	14	TRACEDATA[0]
GND	15	□ □	16	TRACEDATA[1]
GND	17	□ □	18	TRACEDATA[2]
GND	19	□ □	20	TRACEDATA[3]

Beim Verbinden des Debug-Adapters mit Chip1768 ist auf korrekte Polung wie in der folgenden Abbildung zu achten. Pin 1 ist mit einer goldenen „1“ gekennzeichnet:



Weiterhin ist zu beachten, dass Pin 7 als Stift vorhanden ist - beim Verbinden mit dem Debug-Adapter darf die entsprechende Buchse also nicht verschlossen („keyed“) sein. Kann auf die ETM-Funktion verzichtet werden, so werden nur die Pins 1 bis 10 benötigt.

3.4 Takterzeugung

Neben dem internen RC-Oszillator mit 4MHz Frequenz ist Chip1768 mit weiteren Taktquellen ausgestattet: Als Haupttaktgeber für den Mikrocontroller steht ein Quarz mit 12MHz zur Verfügung. Seine Gangabweichung beträgt $\pm 30\text{ppm}$ (bei 25°C).

Als Low-Power Taktquelle fungiert ein 32,768kHz-Quarz. Er weist eine Ganggenauigkeit von

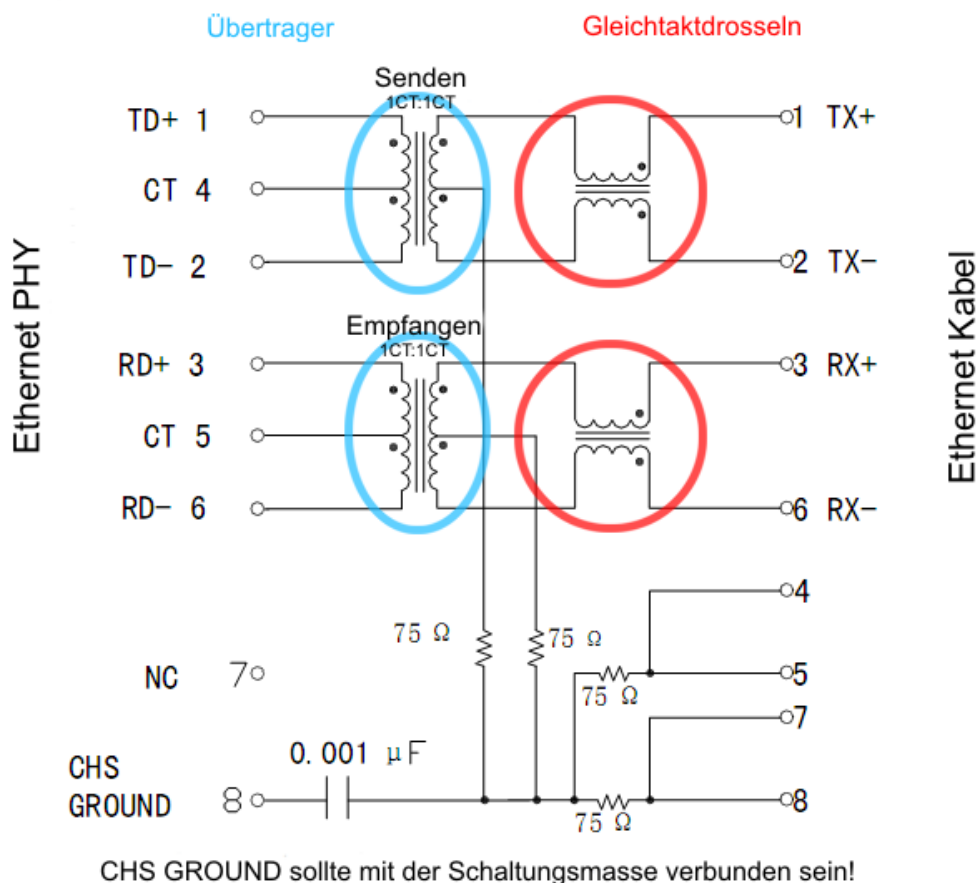
+20ppm (bei 25°C) auf.

Der Ethernet-Transceiver ist über ein RMI-Interface an die MCU gekoppelt, was eine Taktrate von 50MHz notwendig macht. Dieser Takt wird von einem integrierten Oszillator erzeugt und in den DP83848J sowie in den LPC1768 gespeist. Der Oszillator lässt sich mit dem Prozessor-Signal P1[27] ein- bzw. ausschalten. Ein High-Pegel aktiviert den Oszillator. Im Betrieb hat der Oszillator eine Stromaufnahme von max. 15mA. Die Frequenztoleranz beträgt +50ppm.

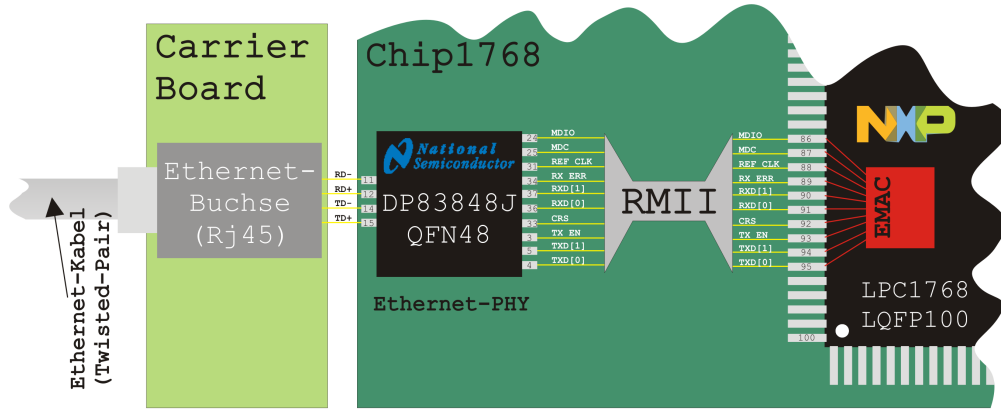
3.5 Ethernet

Die Ethernet-Funktionalität basiert auf dem Ethernet-Transceiver („PHY“) DP83848J der Firma National Semiconductors. In Kombination mit der 10/100MBit Ethernet Baugruppe („EMAC“ - Ethernet Media Access Controller) des LPC1768 ermöglicht dies die Verwendung von Chip1768 in Ethernetanwendungen. Einzig eine geeignete LAN-Buchse (normalerweise eine Buchse des Typs RJ45) muss noch zusätzlich verwendet werden.

Diese Buchse sollte mit einem Übertrager ausgestattet sein sowie möglichst auch Gleichtakt-drosseln aufweisen. Nachstehendes Schema verdeutlicht die Anforderungen an die LAN-Buchse.



Die Anbindung des DP83848J an den Mikrocontroller erfolgt über ein RMI-Interface (Reduced Media Independent Interface). Die folgende Abbildung zeigt die beteiligten Signale und ihre entsprechenden Controller-Pins.

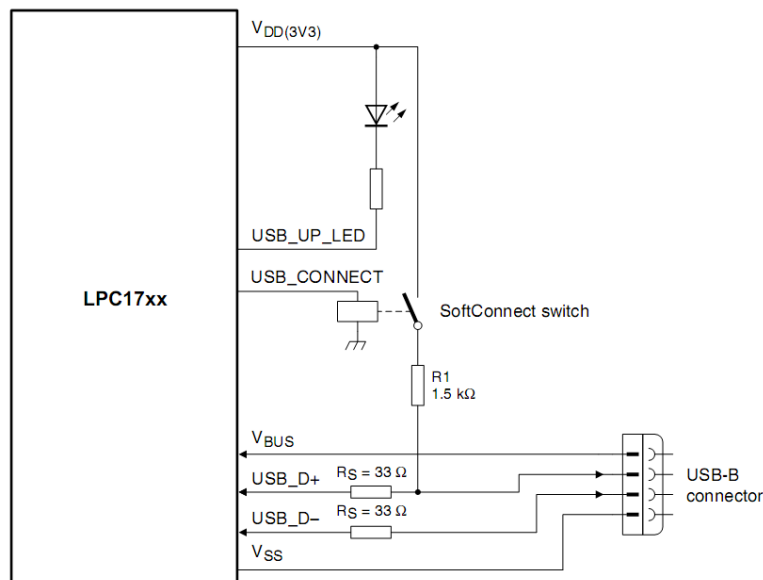


Über das Signal ETH_RST (GPIO P1[28]) kann mit einem LOW-Pegel ein Reset im DP83848J ausgelöst werden. Ein LOW-Pegel auf dem Signal OSCEN (GPIO P1[27]) schaltet den Taktgeber (ein 50MHz Oszillator) für den Ethernet Transceiver ab. Dies ist nützlich, um Anwendungen mit geringem Energiebedarf zu ermöglichen, die keine oder nur eine zeitweise Ethernetkonnktivität erfordern.

Die beiden Mittenabgriffe der Übertrager können für ein besseres Signalverhalten auf VCC und mit einem kleinen Kondensator gegen Masse gekoppelt werden.

3.6 USB

Das Chip1768-Modul kann als „USB-Device“ eingesetzt werden. Die notwendige Beschaltung des LPC1768 ist dafür bereits vorgesehen und richtet sich nach den Empfehlungen von NXP. Das folgende Schema erläutert die USB-Device Schaltung:



Der als „SoftConnect Switch“ bezeichnete, softwarekontrollierbare Schalter dient der Beaufschlagung der USB_D+-Leitung mit dem High-Pegel über den spezifikationsgerechten 1,5kΩ Widerstand R1. Dieser High-Pegel auf der Datenleitung dient der Signalisierung an den USB-Host, dass ein Full-Speed-Device angeschlossen ist. Da der USB Host beide Datenleitungen über einen 15kΩ-Widerstand auf Masse zieht (Pull-Down), wird der Pull-Down auf der USB_D+-Leitung

bei geschlossenem Schalter durch den Pull-Up über einen geringeren Widerstand (hier $1,5k\Omega$) „übergangen“.

Das USB_CONNECT-Signal aus obiger Abbildung heißt im Schaltplan des Chip1768 USB_ON und ist mit dem Controller-Pin 64 (P2[9]) verbunden.

Wenn Chip1768 als sog. „Bus-powered“ USB Device eingesetzt werden soll, sind zusätzliche Maßnahmen zu ergreifen: Es muss beachtet werden, dass die USB-Spezifikationen eine Stromaufnahme von max. 100mA (Low Power) bzw. 500mA (High Power) erlaubt. Ein Strombedarf von über 100mA muss jedoch vorher vom Device beim USB Host „beantragt“ werden. Zudem sollte eine Sicherung in der Schaltung vorgesehen werden, die den Strom auf diese zulässigen Werte beschränkt. In die Planung sollte auch einfließen, dass auf der USB +5V-Leitung hochfrequente Einstörungen auftreten können, die mit geeigneten Mitteln (z.Bsp. Ferritelemente) unterdrückt werden sollten. Ebenso kann es relativ große Abweichungen von den angestrebten +5V geben - die Schaltung muss entsprechend tolerant ausgelegt werden (Toleranzbereich: 4,4V..5,25V).

3.7 PWM

Der NXP LPC1768 bietet Pulsweitenmodulation für bis zu sechs Kanäle. Chip1768 stellt hierfür die Signale P21 bis P26 zur Verfügung. P26 ist dabei mit dem Controller-Pin P2[0], P21 mit P2[5] verbunden. Über die PWM-Ausgänge können beispielsweise Dimmvorgänge bei Beleuchtungsanwendungen, Drehzahlsteuerungen für Motoren oder ähnliche Anwendungen realisiert werden.

3.8 Analog

Chip178 erlaubt die Verwendung von bis zu acht Analogeingängen. Beaufschlagt mit Spannungen von 0..3,3V berechnet der Analog-Digital-Wandler (ADC) im LPC1768 12Bit breite Digitalwerte der einzelnen Eingangssignale. Der ADC arbeitet mit einer Abtastrate von bis zu 200kHz. Die Vergleichsspannungen V_{REFP} und V_{REFN} liegen fest auf 3,3V bzw. 0V/GND.

Achtung:

Es muss sichergestellt werden, dass die Signalspannungen an den Analogeingängen auf keinen Fall 3,3V überschreiten! Andernfalls kann es zu Beschädigungen der ADC-Baugruppe bzw. des gesamten Mikrocontrollers auf Chip1768 führen!

Ein einkanaliger Analogausgang erlaubt das Erzeugen analoger Spannungen. Chip1768 führt diese Funktion auf P18 heraus. Der Digital-Analog-Wandler (DAC) im LPC1768 arbeitet mit einer Auflösung von 10Bit und einer Updaterate von bis zu 1MHz. Ebenso wie beim ADC befinden sich die Referenzspannungen bei 0V und 3,3V.

3.9 Serielle Schnittstellen

Chip1768 bietet die Möglichkeit, bis zu vier universelle asynchrone serielle Empfangs- und Sendebaugruppen (UARTs) zu nutzen. UART1 weist dabei eine Vollbeschriftung mit den vier Flow-Control Signalen CTS, DCD, DSR und DTR auf. Die anderen UARTs verfügen jeweils über eine Sende- und eine Empfangsleitung.

An den Signalen P9 und P10 lässt sich die Funktion für den CAN-Bus einstellen.

Für die Nutzung von Erweiterungshardware für die Soundein- und -ausgabe lässt sich die I²S-Schnittstelle aktivieren.

Chip1768 kann ebenfalls mittels I²C-Bus kommunizieren. Hierfür stehen zwei Schnittstellen zur Verfügung.

Die Anbindung von SPI-Hardware kann über zwei SSP-Schnittstellen (diese beinhalten einen ressourcenschonenden DMA-Controller) oder eine SPI-Verbindung erfolgen.

3.10 RTC

Die „Real Time Clock“-Peripherie (RTC) im NXP LPC1768 kann über eine Stützbatterie selbst dann noch die Zeit weiterzählen, wenn die Hauptspannungsversorgung unterbrochen wird. Dafür muss eine Sekundärspannungsquelle an den VBAT-Eingangspin von Chip1768 mit einer Spannung von ca. 3V angeschlossen werden. Gute Dienste leistet hier eine handelsübliche CR2302-Knopfzelle.

4 Applikationshinweise

4.1 Der Bootloader

Der NXP LPC1768 ist mit einem Bootloader ausgestattet, der das Programmieren des Programmspeichers (Flash) ohne Cortex-Debug (also JTAG oder SWD) Adapter erlaubt. NXP unterstützt das für nicht-kommerzielle Nutzung kostenfreie Programm FlashMagic, mit dem sich der Bootloader-Modus komfortabel nutzen lässt.

Der LPC1768 prüft ca. 5ms nach der Low/High-Flanke des Reset-Signales, welcher Pegel am ISP-Signal anliegt. Im Normalfall sorgt ein Pull-Up-Widerstand für das Starten des Nutzerprogrammes aus dem Programmspeicher des Mikrocontrollers. Liegt allerdings ein Low-Pegel an, wird der Bootloader gestartet.

Um den Bootloader zu starten, muss das Signal ISP am Chip1768 auf GND/Masse gelegt sowie ein Hauptreset ausgelöst werden. Die Kommunikation kann über die Signale IF+ sowie IF- erfolgen.

Das Starten des Bootloaders lässt sich mit geringem Schaltungsaufwand komfortabler gestalten. Durch die Nutzung der „Hardware Handshake“ bzw. „Flow-Control“-Signale DTR oder CTR der seriellen Schnittstelle kann ein automatischer Reset sowie ein Low-Pegel am ISP-Signal erzeugt werden. FlashMagic unterstützt diese Komfortvariante.⁴

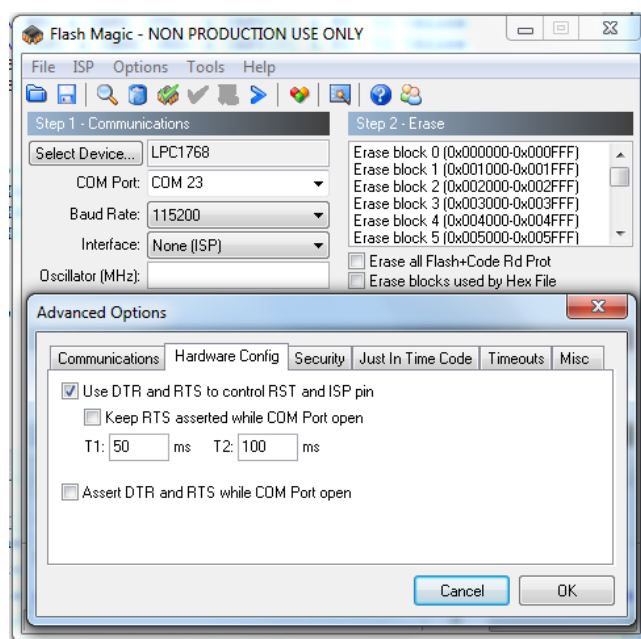


Abbildung 1: FlashMagic für die komfortable Nutzung des Bootloaders

⁴Bei den meisten seriell-zu-USB Konvertern (u.a. von FTDI angeboten) steht leider nur eines dieser Signale zur Verfügung. Somit kann mit diesen Adaptern entweder das Reset- oder das ISP-Signal automatisch generiert werden.

4.2 Anpassen von mbed-Programmen

4.2.1 Ethernetanwendungen

Chip1768 verfügt über keinen zusätzlichen Steuer-Prozessor wie ihn mbed aufweist. Dieser wird beim mbed u.a. für die Bereitstellung einer IPv4-MAC-Adresse genutzt. Programme, die diesen Steuer-Prozessor nach dieser Nummer abfragen, können mit dem Chip1768 nicht ohne Anpassungen genutzt werden. Dazu zählt die EthernetNetIf-Bibliothek des mbed Online-Compilers.

Um Programme, die auf dieser Bibliothek aufbauen, nutzen zu können, muss die entsprechende Funktion neu deklariert („überschrieben“) werden. Ein sehr einfacher Lösungsansatz besteht darin, eine MAC-Adresse fest in den Programmcode zu schreiben. Denkbar wäre natürlich ebenfalls, die MAC-Adresse aus der Unique-ID des LPC1768 berechnen zu lassen. Alternativ kann auch ein externer nichtflüchtiger Speicher für die Speicherung der MAC-Adresse verwendet werden.

Folgende Funktion hat sich als einfache Lösung bewährt:

```
extern "C" void mbed_mac_address(char *s)
{
    char mac[6];
    mac[0]=0x0A;
    mac[1]=0xC1;
    mac[2]=0x10;
    mac[3]=0x51;
    mac[4]=0x0B;
    mac[5]=0xCC;
    memcpy(s, mac, 6);
}
```

Das Einbauen dieser Funktionsdeklaration führt zum Überschreiben der bestehenden Funktion `mbed_mac_address`, die bei der EthernetNetIf-Bibliothek für das Erstellen der MAC-Adresse zuständig ist. Durch diese Codeanpassung wird es möglich, mbed-Applikationen, die Netzwerkverbindungen benötigen, auch auf dem Chip1768-Modul zu nutzen.

4.3 Verhalten nach Reset

- Nach ca. fünf Sekunden prüft der Controller, ob der Bootloader gestartet werden muss (Low-Pegel am Pin ISP)
- Allen GPIOs ist ihre Primärfunktion (digitaler Eingang) zugeordnet
- Der interne Oszillator schwingt an und der Controller arbeitet mit ca. 4MHz Systemtakt
- Aus Sicht des Nutzerprogrammes stellt sich nach dem Reset die Speicherverteilung wie auf Seite 22 dar.
- Der „Watchdog Timer“ ist ausgeschaltet

4.4 SRAM

NXPs LPC1768 verfügt über 64kB SRAM. Jedoch ist zu beachten, dass dieser in zwei Blöcken zu je 32kB vorliegt. Dies gilt es im Blick zu behalten, wenn mit großen Variablen umgegangen wird - falls der Speicherbedarf 32kB übersteigt, muss geteilt werden.

4.5 Nutzung in Verbindung mit KEIL MDK ARM

Mit der „MDK-ARM“ Entwicklungsumgebung („IDE“ - Integrated Development Environment) bietet der Hersteller Keil eine sehr komfortable Software für die Programmentwicklung für ARM-Controller an. Die IDE vereint eine Projektverwaltung, einen C-Compiler sowie einen Debugging-Teil in einer Windows-Anwendung. Vorallem die Debug-Funktionalität stellt dabei einen herausragenden Komfortgewinn für den Firmwareentwickler dar.

Für kleinere Projekte sowie zur Evaluierung vor der Kaufentscheidung steht eine auf 32kB Codegröße limitierte Version des Softwarepaketes für den Download kostenfrei zur Verfügung.

4.5.1 Voraussetzungen

Da es sich um eine Windows-Software handelt, ist eine Installation des Microsoft-Betriebssystems natürlich zwingende Voraussetzung. Zudem wird ein geeigneter Adapter benötigt, der die elektrische Verbindung mit Chip1768 herstellt. Über diesen Adapter gelangt Programmcode auf den Controller und das Debuggen wird ermöglicht.

Folgende Adapter werden in der aktuellen Version von MDK-ARM (μ vision Version 4.20) unterstützt:

- Keil ULINK Pro, ULINK2, ULINK ME
- RDI Interface Driver
- Altera Blaster Cortex Debugger
- Stellaris ICDI
- Signum Systems JTAGjet
- Cortex-M/R J-LINK/J-Trace
- ST-Link Debugger
- NULink Debugger

4.5.2 Real-Time Trace

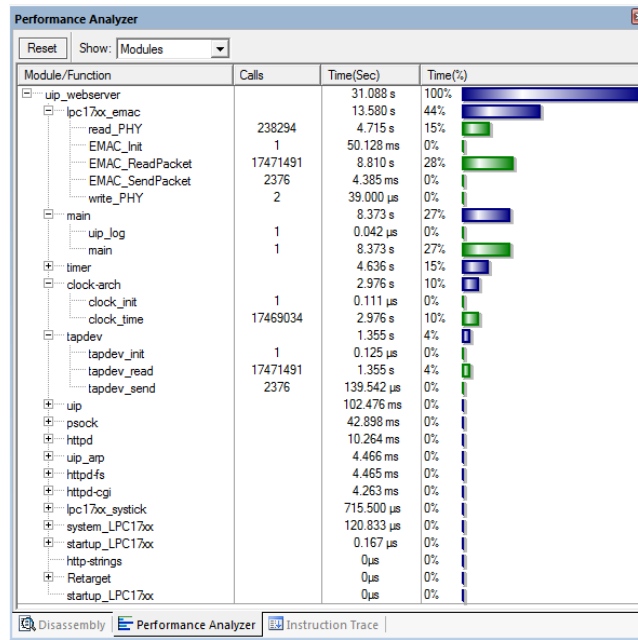
Der verwendete Mikrocontroller von NXP beinhaltet ARMs Cortex ETM⁵ Trace Peripherie. Diese ermöglicht es, jeden einzelnen Schritt der Programmabarbeitung aufzuzeichnen - während das Programm in Echtzeit abläuft. Erweiterten Analysen der Controllerapplikation steht damit nichts mehr im Wege.

MDK-ARM ermöglicht u.a. gefilterte, chronologisch sortierte Auflistungen aller Prozessorschritte:

#	Type	Flag	Num	PC	Opcode	Instruction	Source Code	Address	Data	Cycles	Time[s]
103810	ETM			0x00001E2	F4C	LDR r0, #0x00000000	LPC_SC->SCS = ~BIT4; //wait OSC				
103819	ETM			0x0000ACE	6000	LDR r0, #0x00000000					
103820	ETM			0x00000A3	F4C	BIC r0, #0x00000000					
103821	ETM			0x0000A04	43A	LDR r1, #0x232; @0x00000000...					
103822	ETM			0x0000A06	F8C10A0	STR r0, #0x1A0					
103823	ETM			0x0000A06	408	MOV r0, #1	403: LPC_SC->SCS = BITS;				
103824	ETM			0x0000A0C	F8D00A0	LDR r0, #0x1A0					
103825	ETM			0x0000A03	F460000	ORR r0, #0x20					
103826	ETM			0x0000A04	F8C10A0	STR r0, #0x1A0					
103827	ETM			0x0000A0B	BF00	NOP	404: while(LPC_SC->SCS&BIT6) == 4; //wait for oscillator...				
103828	ETM			0x0000A04	4045	LDR r0, #0x270; @0x00000000...					
103829	ETM			0x0000A0C	6000	LDR r0, #0x00000000					
103830	ETM			0x0000A0E	F0000A0	AND r0, #0x40				1697	0.00001697
103831	ETM			0x0000A05	2004	CMG r0, #0x04				1701	0.00001701
103832	ETM			0x0000A0F	D0F9	BEQ 0x0000A0EA					
103833	ETM			0x0000A0F	43A3	LDR r0, #0x200; @0x00000000...	405: LPC_SC->PLLCON = ~BIT1; //disconnect P...				
103834	ETM			0x0000A0B	6000	LDR r0, #0x00000000					
103835	ETM			0x0000A0A	F000002	BIC r0, #0x02					
103836	ETM			0x0000A0E	430	LDR r1, #0x100; @0x00000000...					
103837	ETM			0x0000B00	F8C10B0	STR r0, #0x00	406: LPC_SC->PLLFEED = 0xA;				
103838	ETM			0x0000B0A	F46000A	MOV r0, #0xA					
103839	ETM			0x0000B08	43F	LDR r1, #0x252; @0x00000000...					
103840	ETM			0x0000B0A	6008	STR r0, #0x00	407: LPC_SC->PLLFEED = 0x5;				
103841	ETM			0x0000B0C	F040005	MOV r0, #0x55					
103842	ETM			0x0000B10	432B	LDR r1, #0x172; @0x00000000...					
103843	ETM			0x0000B12	F8C100C	STR r0, #0x00					
103844	ETM			0x0000B16	408	MOV r0, #1	408: LPC_SC->PLLCON = ~BIT0; //PLL disable				
103845	ETM			0x0000B18	F8D00B0	LDR r0, #0x00					
103846	ETM			0x0000B1C	F000001	BIC r0, #0x01					
103847	ETM			0x0000B20	4338	LDR r1, #0x224; @0x00000000...				1833	0.00001833
103848	ETM			0x0000B22	6008	STR r0, #0x00					
103849	ETM			0x0000B24	F04000A	MOV r0, #0xA	409: LPC_SC->PLLFEED = 0xA;				
103850	ETM			0x0000B28	4307	LDR r1, #0x200; @0x00000000...					
103851	ETM			0x0000B2A	6008	STR r0, #0x00					
103852	ETM			0x0000B2C	F040005	MOV r0, #0x55	410: LPC_SC->PLLFEED = 0x5;				
103853	ETM			0x0000B30	6008	STR r0, #0x00					
103854	ETM			0x0000B32	F040001	MOV r0, #0x01	411: LPC_SC->CLKSRCSEL = 1; //select the main oscillator as a...				
103855	ETM			0x0000B36	4322	LDR r1, #0x130; @0x00000000...					
103856	ETM			0x0000B38	F8C100C	STR r0, #0x100					
103857	ETM			0x0000B3C	408	MOV r0, #1	412: LPC_SC->PLLCFG = freqM; //sets M				
103858	ETM			0x0000B3E	F8D00B4	LDR r0, #0x04					
103859	ETM			0x0000B42	F040010	ORR r0, #0x10					
103860	ETM			0x0000B46	F8C100A	STR r0, #0x04	413: LPC_SC->PLLCFG = freqN<10; //sets N			1969	0.00001969
103861	ETM			0x0000B4A	408	MOV r0, #1					
103862	ETM			0x0000B4C	F8D00B4	LDR r0, #0x04					
103863	ETM			0x0000B50	F440000	ORR r0, #0x10000					
103864	ETM			0x0000B54	432D	LDR r1, #0x100; @0x00000000...					
103865	ETM			0x0000B58	6008	STR r0, #0x00					
103866	ETM			0x0000B5A	F04000A	MOV r0, #0xA	414: LPC_SC->PLLFEED = 0xA;				
103867	ETM			0x0000B5C	430A	LDR r1, #0x100; @0x00000000...					
103868	ETM			0x0000B5E	6008	STR r0, #0x00					
103869	ETM			0x0000B60	F040005	MOV r0, #0x55					
103870	ETM			0x0000B64	4316	LDR r1, #0x08; @0x00000000...	415: LPC_SC->PLLFEED = 0x5;				
103871	ETM			0x0000B68	F8C100C	STR r0, #0x00					
103872	ETM			0x0000B6A	408	MOV r0, #1					
103873	ETM			0x0000B6C	F8D00B0	LDR r0, #0x00	416: LPC_SC->PLLCON = 1; //enable PLL				
103874	ETM			0x0000B70	F040001	ORR r0, #0x01					
103875	ETM			0x0000B74	432	LDR r1, #0x140; @0x00000000...					

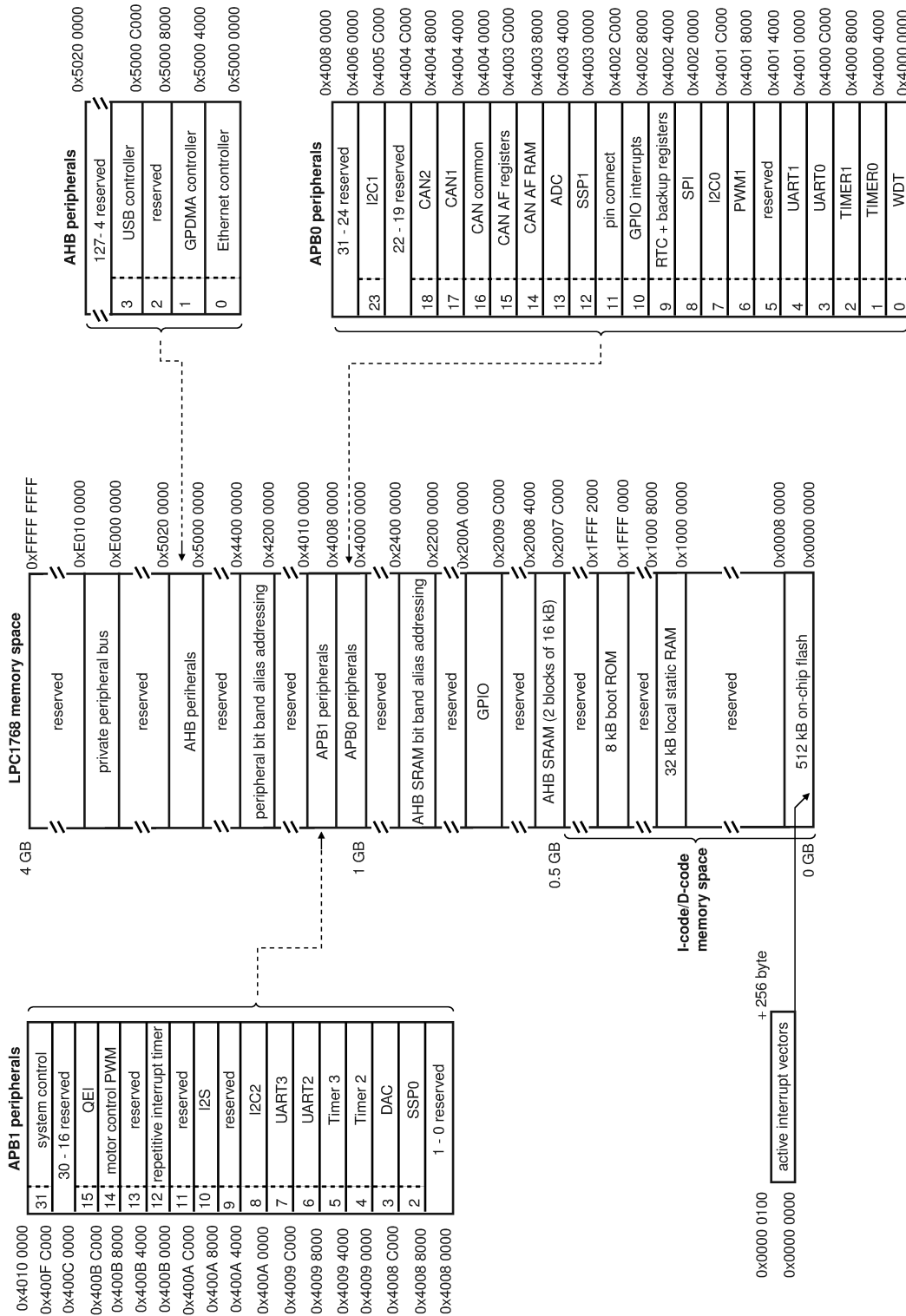
Ebenso kann mittels der Real-Time Trace Funktionalität von Chip1768 auf die Performance der Applikation bzw. der Controllerauslastung geschlossen werden. Innerhalb weniger Sekunden lassen sich somit Rückschlüsse auf „Problemstellen“ der zu entwickelnden Software ziehen:

⁵ETM - Embedded Trace Macrocell; eine optionale Komponente, die von Chipfertigern in ihre ARM Cortex-Derivaten vorgesehen werden kann. Nur mit der ETM ist die Nutzung von Real-Time Trace möglich.



5 Anhang

5.1 Memory Map



Quelle: Handbuch zum NXP LPC1768, Seite 13

5.2 Mechanische Abmessungen

