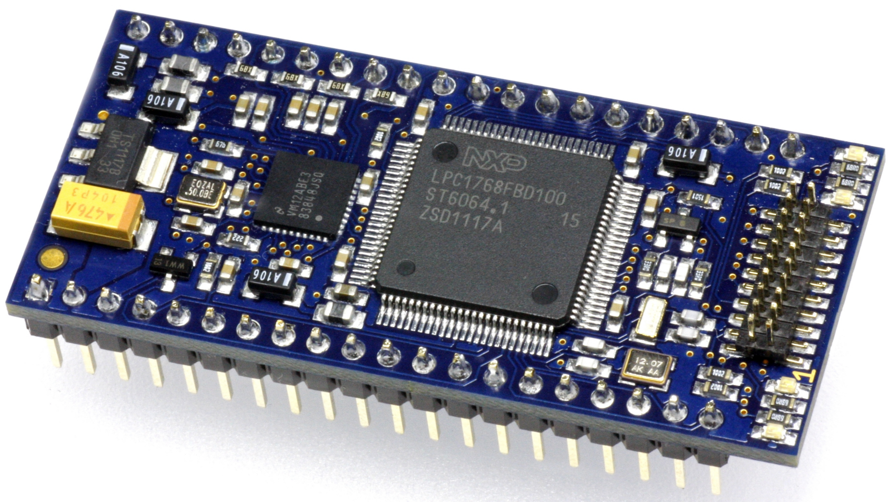
ELEKTRONIK LADEN

Chip1768

Hardware Version V1.21



ARM Cortex-M3 Microcontroller Board with NXP LPC1768

User's Manual

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

Contents

1 Overview

The Chip1768 module in an easy way allows to profit from the calculation power and richt equipment of the NXP LPC1768 microcontroller. With its compact form factor in 100mil standard grid the module can quickly be applied in new projects and existing units. Chip1768 implements the circuitry necessary for using it as a robust and flexible tool for a first Proof-of-Concept as well as Rapid-Prototyping and in the final stage as productive hardware.

The controller module is built around the NXP LPC1768 microcontroller. Featuring an ARM Cortex-M3 core (ARMv7M) architecture, this controller combines advanced debug features, energy saving facilities, rich peripheral equipment and compatibility of soft- and hardwaretools.

Chip1768's alikeness to ARM/KEIL's mbed1768 isn't pure coincidence as both are completely compatible in soft- and hardware[1]. This fact opens new possibilities to the embedded developer: First demonstrations and functional tests can be made with mbed1768 and its great online compiler and libraries. Developing commercial applications (where the sourcecode shouldn't exposed to the "‘cloud"’) is then done the "‘conventional way"’ with Chip1768 and an appropriate toolchain - on-chip-debugging included.

[1]However, a small change in sourcecode for applications dealing with ethernet is necessary - chapter 4.2.1 gives details about this

1.1 Technical Data

1.1.1 Chip1768 module

- Compact microcontroller module with NXP LPC1768 MCU
- Cortex Debug Interface, 2x10 pin header with 50mil grid, following ARMs specification for Trace Port Interface Unit ("‘TPIU"’)
- Supports JTAG, SWD, 4-Bit Trace
- National Semiconductor DP83848J Ethernet Transceiver
- 12MHz and 32,786kHz crystals
- 4 signalling LEDs
- Supply voltage 4,5V .. 9V
- max. current consumption app. 200mA (test case: 5V supply and actively transmitting data over ethernet)
- extra-wide DIP form factor, 2x20 pin headers
- 100mil pin grid, 900mil DIP with
- Overall dimensions: 2,22"’ x 1,02"’

1.1.2 NXP LPC1768 microcontroller

- ARM Cortex-M3 (ARMv7) 32-Bit CPU
- up to 100MHz Main (Core) frequency
- 512kB program memory (Flash), 2x32kB RAM
- Supports ARM Cortex ETM Trace
- 10/100MBit ethernet, RMI interface, DMA controller
- 12-Bit Analog-Digital-Converter, 8 channels
- 10-Bit Digital-Analog-Converter, 1 channel
- 4 32-Bit width timers
- 6 PWM channels, 1x Motor Control PWM, 8 DMA channels
- USB 2.0 interface with integrated transceiver
- CAN2.0B with 2 channels
- 4x UART, 2x SSP, 1x SPI, 3x I^2C, 1x I^2S
- Interface for quadrature encoder
- Low Power RTC, Unique ID
- internal 4MHz RC oscillator

2 Functions

NXP's LPC1768 offers 100 pin connectors in an LQFP package, 70 of them being GPIOs ("‘General Purpose Input/Output"’). The compact form factor of Chip1768 and its limited number of GPIOs made it necessary to make a selection, which of the LPC1768's GPIOs are used. Most of the mbed1768's GPIOs were kept unchanged in Chip1768, however, some enhancements were made. The differences between Chip1768 and mbed1768 were discussed in chapter 2.3.

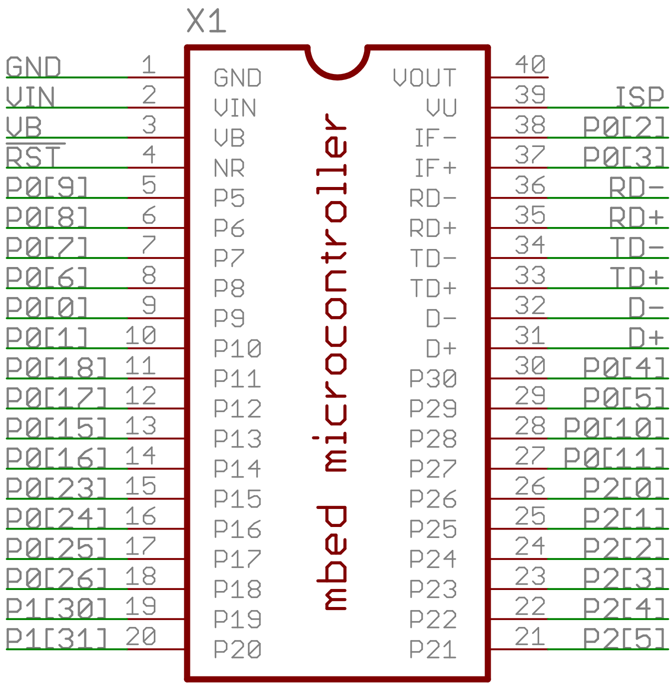
Following the possibilities of the Chip1768 module:

- 28x GPIOs, 26 of these with ability to generate interrupts
- 2x I^2C
- 1x I^2S (duplex capable)
- 2x SSP, 1x SPI, 2x CAN
- 1x Full-Speed USB
- 1x 10/100MBit Ethernet
- 6x Match outputs for timers (4x Timer2, 2x Timer3)
- 4x Capture inputs for timers (each 2x for Timer2 and 3)
- 4x UART (from which 1 with optional handshake signals DTR, DSR, DCD and CTS)
- 8x inputs for Analog-Digital-Converter
- 1x output for Digital-Analog-Converter
- 6x PWM outputs
- 1x external, non-maskable interrupt input (NMI)
- Real-Time-Clock (RTC) with separate power supply input

As for the sheer amount of built-in functions of the used microcontroller this manual can't focus on how to use these peripherals in applications. For developing with Chip1768, a continuous reference to NXP's manual for the LPC1768[2] will be a good start. Each peripheral function is explained in detail and getting the "‘easier"’ peripherals, like the ADC or UARTs, to work is done in a straight-forward way (step-by-step instructions). This user manual will guide the starter as well as the experienced developer through the process of writing an application for the LPC1768.

[2]Download at: http://www.nxp.com/documents/user_manual/UM10360.pdf

2.1 Pinout



Signal	Type	Description
GND	-	circuit ground
VIN	Input	4,5 .. 9V
VB	Input	3V power supply for RTC
!RST	Input, low-active	main reset for LPC1768
P5..P30	Inputs/Outputs	multifunctional ports, table 2.2 has details
D+	Input/Output	USB data line
D-	Input/Output	USB data line
TD+	Output	Ethernet transmit line, positive
TD-	Output	Ethernet transmit line, negative
RD+	Input	Ethernet receive line, positive
RD-	Input	Ethernet receive line, negative
IF+	Input	receive line for Bootloader mode
IF-	Output	transmit line for Bootloader mode
ISP	Input	receive line for Bootloader mode

2.2 Table of Functions

Due to the many functions built into modern microcontrollers, there wouldn't be enough port-pins (GPIOs) if each of them has its own, unique function. To solve this problem, signal multiplexing allows each single GPIO to get one of several possible functions assigned. In the case of LPC1768, each GPIO can have up to four different functions.

To select a GPIO's function, special registers are used in the controller. These registers can be programmed at program run-time or at startup (via a "'Startup-Script"'). In the LPC1768, the registers named `PINSELxx` in the `PINCON` peripheral are responsible for the signal multiplexing. In these registers, each GPIO has two bits representing its current selected function.

The following table offers valuable clues to the usable combination of functions:

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

2.3 Differences between Chip1768 and mbed1768

USB

Mbed1768 implements two USB interfaces: a "‘visible"’ one with mounted mini-USB connector and a second one which is simply brought out through the pin headers (signals `D+` and `D-`). The first interface is used to program mbed1768 (*.BIN files from the online-compiler) as well as for communication via virtual COM port. Also, mbed1768 can get powered using the mounted USB connector. The user program can't use this USB interface, though. It is hidden behind the "‘black box"’ which handles all the mbed1768's magic. Applications which need USB connectivity have to use the second interface.

Chip1768 in contrast only has one USB interface, directly connected to the controller's USB peripheral. Program code is transferred to the on-chip flash of LPC1768 using the Cortex Debug Interface (with JTAG or Serial Wire). Likewise Chip1768 passes on mbed's "‘magic"’ like the USB flash disk function (mbed offers 2MB storage space for program code (BIN-files), websites, captured data,...

Power supply output V$_{OUT}$

Mbed can be used to power an external circuit as it offers a voltage of 3,3V at a maximum current of 800mA. On Chip1768, the pin V_{OUT} is left unconnected.
The mbed's `VU` pin delivers the host's USB supply voltage (somewhat around 5V) - Chip1768 doesn't have the primary USB connector, so this voltage can't be used.

Bootloader

Chip1768 enables the user to take advantage of the LPC1768's builtin bootloader mode. This is accomplished by breaking out the necessary signals to the pins `ISP, IF+` und `IF-.` Further information on how to use the integrated bootloader can be found in chapter4.1 at page 17.

Analog-Digital Converter (ADC)

The pins named `IF+` and `IF-` are associated to the LPC1768's signals `P0[2]` and `P0[3]`. One of their respective functions are inputs for the ADC. Therefore, on Chip1768 all 8 channels of the ADC peripheral are useable instead of only 6 with mbed1768.

UART0

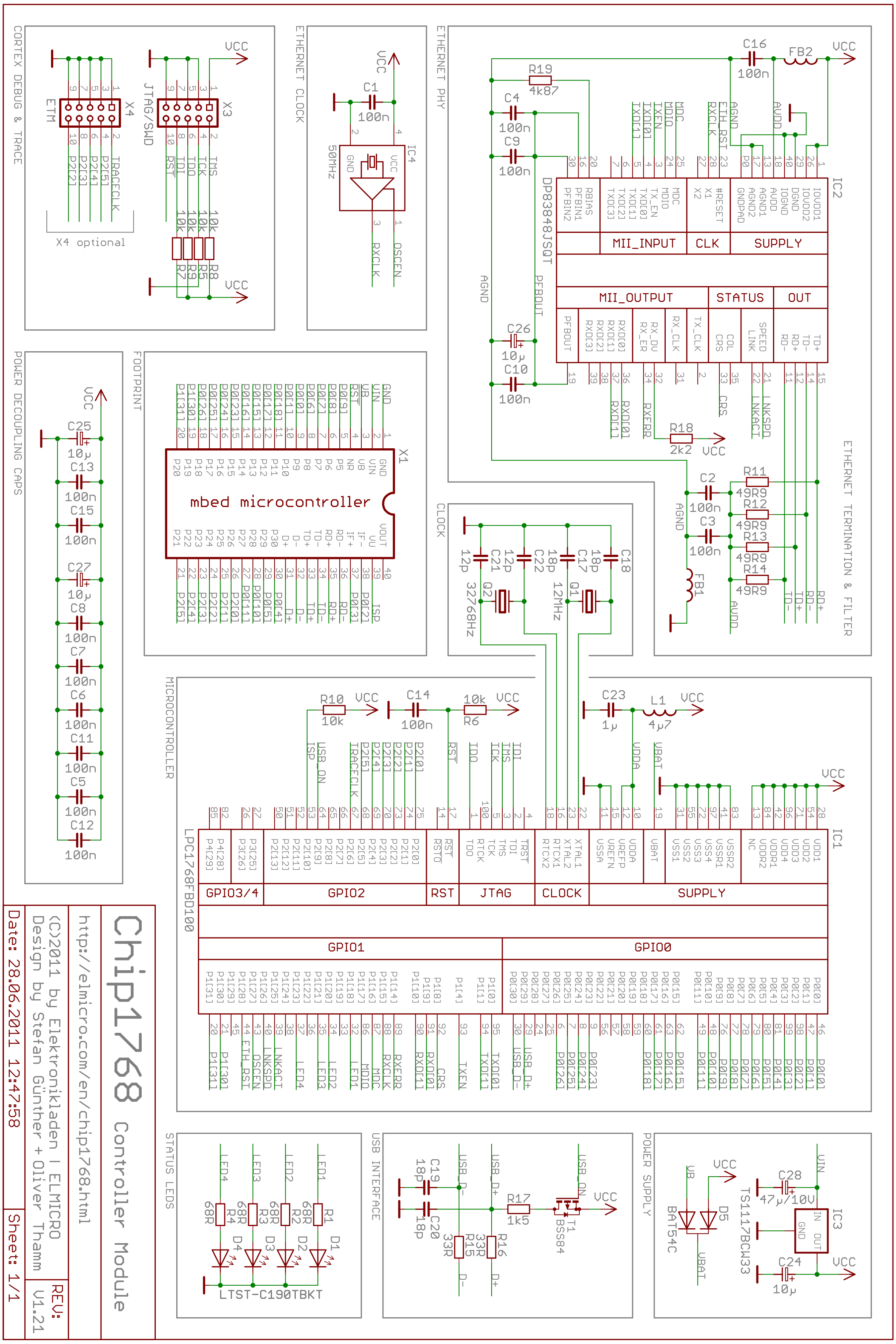
Another GPIO function of `IF+` und `IF-` is the usage as data lines for the UART0-peripheral of the LPC1768. UART0 is fully useable by Chip1768 whereas mbed1768 lacks support of this serial port (mbed uses UART0 for some of its "‘magic"’).

NMI

The LPC1768's Non-Maskable Interrupt is connected to Pin `ISP` of Chip1768.

3 Functional Description

3.1 Circuit Schematic



3.2 Power Supply

Power supply has to be connected with Chip1768's pin $\mathtt{V}_{IN}$. A voltage regulator converts the input voltage down to 3,3V the module needs for operation. Whereas the LDO (TS1117B) accepts input voltages of up to 12V, the used buffer tantal capacitor C28 is rated at 10V - therefore never attempt to feed a voltage higher than 10V to Chip1768!

It has to be considered that, based on the linear regulator principle, the voltage regulator creates the more heat the higher its input voltage is. Keeping the input voltage at about 5V would therefore be the best choice. The power supply must be able to keep up the voltage at currents of max. 200mA.

As explained, the input voltage is buffered by a tantal capacitor of 47μF and feeds the linar regulator IC3. This LDO outputs 3,3V (named VCC in schematic). At important power supply inputs of Chip1768's ICs the power supply is bufferd with 47μF capacitors and decoupled using 100nF. Voltage drops due to short peaks in power consumption are attenuated and ripple on the VCC lane is reduced.

Attention:
Power supply input neither is protected against wrong polarity nor fused against short circuit! Voltage must never exceed 10V! Recommended input voltage is 5V.

3.3 Debug Interface

Chip1768 is programmed ("‘flashed"’) and debugged using either the "‘classical"’ JTAG Interface or the Serial Wire Debug (SWD) technique. The implementation of the JTAG support is, however, mainly done to accomplish a backward compatibility with older debug tools. None of ARMs in Cortex-M3 controllers introduced "‘CoreSight"’ debugging technologies are useable through JTAG.

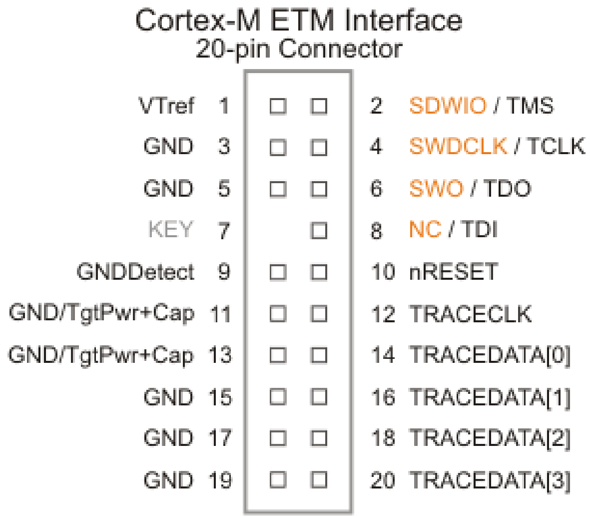
Using the Serial Wire interface enables the developer to take advantage of such features like "‘Data Trace"’ or "‘Instruction Trace"’ making the SWD interface the first choice to debug Chip1768.

An even more sophisticated debug technique is also introduced in Cortex-M3 controllers and is implemented in Chip1768: With "‘Embedded Trace Macrocell"’ (ETM), each and every step the controller does is communicated out of the controller via the Cortex Debug + ETM interface. If ETM is used, the Cortex Debug interface is enlarged by 2x5 pins making the Debug Port a total of 2x10 pins. Using ETM, the developer gains very detailled information about what's going on in the controller while the program runs at real-time.

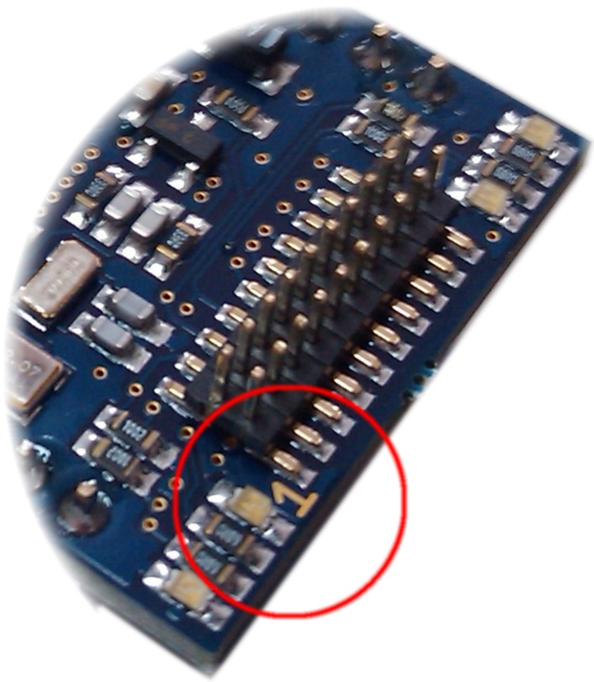
To use this Real-Time-Trace, the toolchain has to be carefully selected. Specialized hardware and software tools supporting ETM are required. As an example, chapter 4.5 deals with the development system from ARM/Keil which comfortably benefits from Chip1768's ETM functionality.

Chip1768 comes with the fully populated Cortex Debug + ETM interface with connectorX3/X4.

The following diagram shows the signal naming according to ARM recommendations:



When connecting the debug adapter with Chip1768, correct polarity has to be ensured - a "‘1"’ is printed near pin 1 of the debug connector:



It is further to be considered, that there is no "‘keying"’ on pin nr. 7 and therefore the 2x5 resp. 2x10 connector must not have a closed connector hole.

3.4 Clock distribution

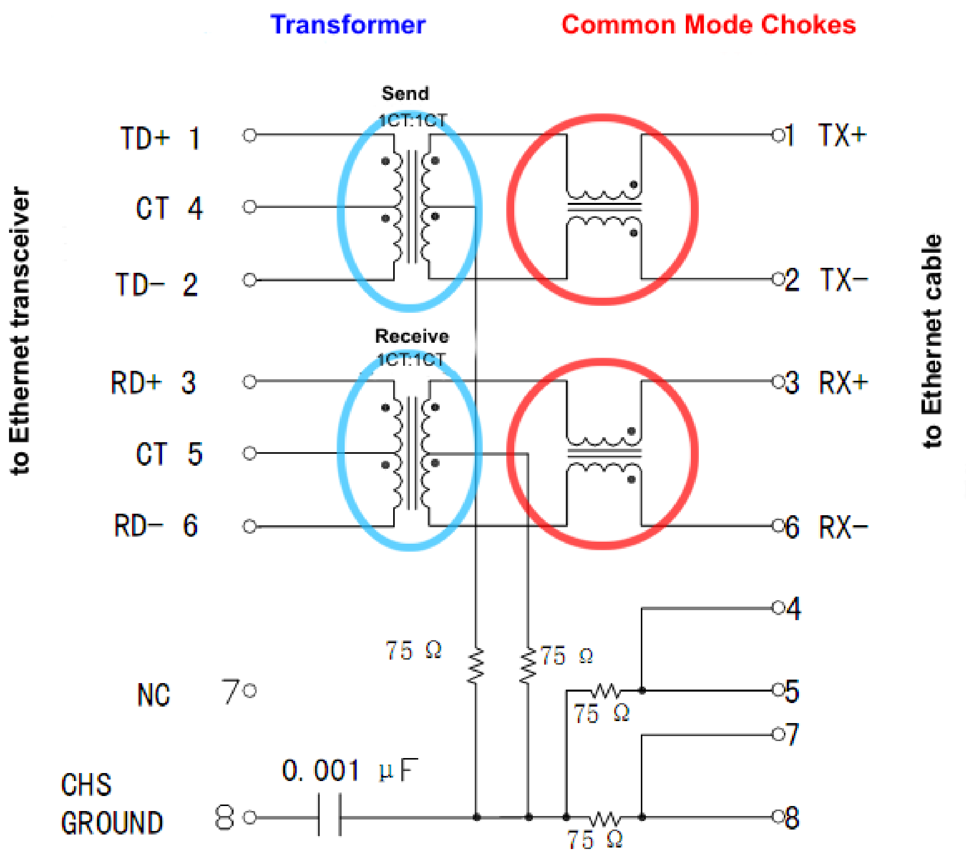
Besides the internal RC oscillator with its 4MHz frequency, Chip1768 is equipped with more clock sources:

As main clock for the LPC1768 microcontroller there is a crystal with 12MHz frequency. Its frequency stability is specified with +-30ppm at 25°C.

For low-power applications using the RTC peripheral, another crystal with a 32,768kHz frequency of +-20ppm impreciseness.
For clocking the RMI interface for ethernet connectivity, a 50MHz oscillator IC is used. This clock feeds the DP83848J transceiver as well as the EMAC peripheral of the LPC1768. Using the controller signal `P1[27]` this oscillator can be switched on and off. When set to "‘high"’, `P1[27]` activates the oscillator which then causes a current flow of up to 15mA. Its precision is rated at +-50ppm.

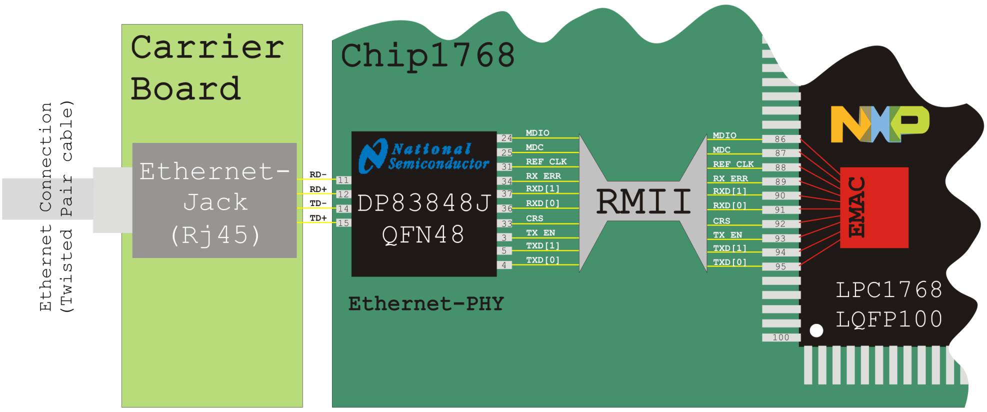
3.5 Ethernet

Ethernet functionality bases on the transceiver-IC ("‘PHY"’) DP83848J from National Semiconductors. In combination with LPC1768’s 10/100MBit ethernet peripheral ("‘EMAC"’ - Ethernet Media Access Controller) Chip1768 supports ethernet enabled applications. According to the "‘Rapid Prototyping"’ concept, all that is needed to implement an ethernet application is a RJ45-jack. Ideally, this jack involves a transformer and chokes ("‘magnetics"’) to ensure a high safety and data integrity standard. Most ethernet jacks are built with the following magnetics and will work well with Chip1768:



CHS GROUND shall be connected with signal ground

The interconnection between the two ICs is done according to the RMII standard (Reduced Media Independent Interface). Following a pictured explanation of the involved signals and their corresponding controller I/Os.

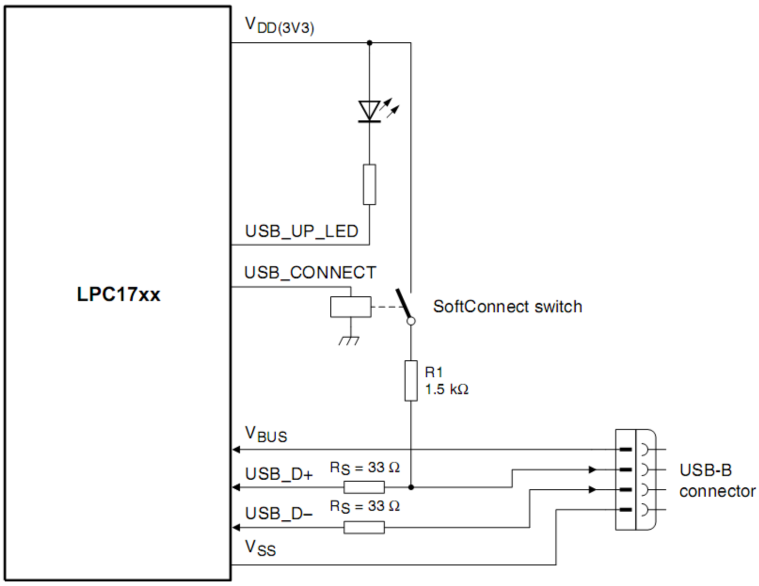


The signal `ETH_RST`, which is internally connected to LPC1768's GPIO P1[28], triggers a reset of the DP83848J when set to LOW. When LOW, signal `OSCEN` (LPC1768 GPIO P1[27]) switches off the 50MHz oszillator for the ethernet transceiver. This is useful for low-power applications which don't need ethernet connectivity all the time or no connection at all.

Both center taps of the transformer's coils can be pulled high and coupled against ground using small capacitors. Doing so could enhance signal integrity and noise immunity of the ethernet connection.

3.6 USB

Chip1768 can be used as "'USB-Device"'. The necessary circuitry for this purpose is implemented according to NXP's recommendation. The following diagram is taken from NXP's user manual of the LPC1768.



The software controllable MOSFET named as "'SoftConnect Switch"' offers the possibility to pull up the `USB_D+` signal with a 1,5kΩ resistor `R1`. Pulled high, `USB_D+` signals the USB host that a Full-Speed-Device is present. As the host pulls both data lines down using a 15kΩ resistor, the smaller value of `R1` overrides the host's pull-down.

The signal `USB_CONNECT` from the schematic above is named `USB_ON` on Chip1768's schematic. It is accessigle through GPIO `P2[9]` of LPC1768.

When Chip1768 shall be used as a bus-powered USB device, some additional facts have to be considered: Attention has to be payed to the maximum current a bus device can draw out of the USB host. Specification restricts current to 100mA (called low power) respectively 500mA (high power). A current higher than 100mA has to be requested by the device, though. This request is done in software by the USB stack implementation.

Additionally, a fuse should be implemented in the USB circuit limiting maximum current to 500mA. Ferrite elements should also be considered as there is a good chance of high-frequency noise overlaying the USB's +5V voltage supply. At least the supply voltage of USB is specified rather weak and voltages from 4,4..5,25V are allowed.

3.7 PWM

The NXP LPC1768 offers pulse-width modulation with up to six independent channels. Chip1768 offers this feature at P21..P26. The PWM outputs can be used for light dimming, motor controlling, sound generation and similar applications.

3.8 Analog

There are eight analog output channels usable with Chip1768, each of which offering an accuracy of 12bits. Voltages to be digitized must be in the range of 0..3,3V. The Analog-Digital-Converter (ADC) operates at up to 200kHz. Chip1768 has the both reference voltages, V_{REFP} and V_{REFN}, tied to 3,3V resp. 0V/GND.

Attention:
It has to be assured that ADS input signals don't exceed 3,3V! Otherwise, damages of the ADC peripheral and/or the whole Chip1768 module can occur!

Apart from the ADC, Chip1768 offers an analog output channel. Using P18, Chip1768 can be used to generate voltages from 0V up to 3,3V (reference voltages are fixed to these values). The accuracy of the Digital-Analog-Converter (DAC) is 10bits. It can operate at update frequencies of up to 1MHz.

3.9 Serial Interfaces

There are four Universal Asynchronous Receive/Transmit units (UARTs) accessible with Chip1768. UART1 offers full hardware handshake support (CTS, DCD, DSR, DTR).

CAN Bus is available at P9 and P10.

For implementing additional sound processing hardware, one I^2S interface can be activated.

For inter-IC-communication two I^2C-Bus interfaces are available.

The interconnection of SPI-based hardware can be carried out using one SPI or two SSP interfaces. Each of the SSP peripheral features a flexible DMA controller.

3.10 RTC

The Real Time Clock peripheral can be powered by a dedicated battery (or another voltage source). Powered this way, it will continously count the time even if the main power for Chip1768 is cut off. RTC power input should have a value of about 3V and must be fed into `VBAT`. A good choice for example would be a CR2302 coin battery.

4 Application Notes

4.1 The Bootloader

NXP delivers a bootloader stored in LPC1768's ROM for updating the program memory (flash) with new firmware. The bootloader is accessed with serial communication and therefore there's no need for using a dedicated debug/flash tool for pure flash programming the Chip1768. NXP supports FlashMagic, a software free for non-commercial use. FlashMagic offers a comfortable way of communicating with the bootloader and updating the firmware with compiled images.

To enter the bootloader, following procedure has to be performed:
Chip1768's pin `ISP` has to be in LOW state (connected to ground) and a reset has to be initiated. Five milliseconds after releasing reset state, the LPC1768 checks for the state of the `ISP` input. If it is LOW then the bootloader will start. If it is HIGH (a pull-up ensures this if `ISP` is left open) then the user program will start.

The communication with the bootloader then is possible with the signals `IF+` and `IF-` erfolgen.

Starting the bootloader could be made more comfortable with little external circuitry. By using the signals `DTR` or `CTR` (hardware handshake) of the serial interface, an reset and/or LOW state at `ISP` could be automatically issued[3]. FlashMagic supports this trick.

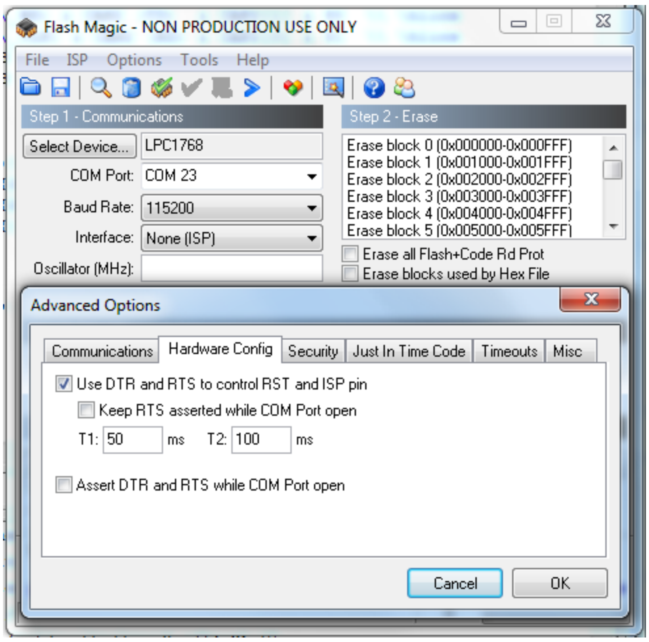


Figure 1: FlashMagic for communicating with the built-in bootloader

[3] Most of the USB-to-serial-converters (like from FTDI) only support one of these signals - so either reset or the ISP state can be generated automatically.

4.2 Editing mbed programs

4.2.1 Using Ethernet

Chip1768 doesn't provide a dedicated control processor like the "'Magic Box"' on mbed1768. Mbed stores a unique MAC address in this special IC. As Chip1768 lacks it, some workaround has to be made for telling the ethernet software library which MAC address to use. The standard library for ethernet applications is the EthernetNetIf-library. Implemented without changes causes Chip1768 to lock at startup as the LPC1768 tries to communicate with the absent mbed1768's magic IC.

To circumvent this behaviour, the function for obtaining the MAC address has to be "'overwritten"' by a new declaration. The simplest solution would be to hard-code a MAC address in the sourcecode like this:

```
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);
}
```

The LPC1768's unique ID could also be used for generating the MAC address. Storing it into an external ROM would also be an option.

4.3 Performance out-of-reset

- After approx. five seconds the controller checks for input level at pin `ISP` - and selects to start the bootloader if level is LOW, otherwise the user application is started.
- All GPIOs are set to digital input.
- Internal RC oscillator will start feeding the main core with 4MHz clock (10% accuracy).
- The user program will start and can access peripherals and memories according to the Memory Map on page 21.
- The Watchdog timer is disabled.

4.4 SRAM

NXP has equipped the LPC1768 with a total of 64kB SRAM. It is, however, to be noted that the SRAM is divided into two blocks of 32kB each. This has to be taken in consideration when handling with larger amounts of contigous data is necessary which then needs to be splittet apart.

4.5 Using KEIL MDK-ARM

With the "'MDK-ARM"' IDE (Integrated Development Environment) KEIL (an ARM company) offers a very comfortable software for developing program code with ARM microcontrollers.[4] The IDE unifies a project management, C-compiler, simulator and a debugger component in one Windows application. In particular, the debugger turns out to be a huge improvement for developing new firmware. In combination with the advanced debugging features of ARMs current Cortex microcontrollers, the time needed for firmware development can dramatically be shortened.

The MDK-ARM development software can be downloaded free of charge and used for evaluation and even small commercial projects. The free version is limited to 32kB codesize.

4.5.1 Requirements

Being a Windows software, a working installation of Microsoft's operating system of course is obligatory. A suitable debug adapter is also needed for connecting Chip1768 to the PC. This adapter allows flash programming as well as debugging.

Following a list of adapter hardware supported by the current version of MDK-ARM (μvision Version 4.20):

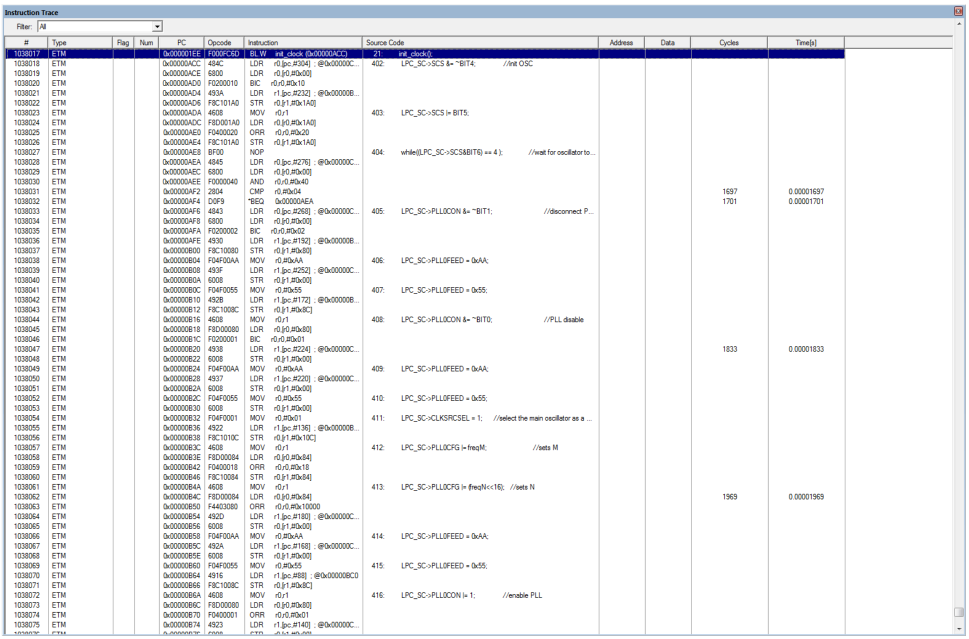
- Keil ULINK Pro, ULINK2, ULINK ME
- RDI Interface Driver
- Altera Blaster Cortex Debugger
- Stellaris ICDI
- Signum Systems JTAGjet
- Cortex-M/R J-LINK/J-Trace
- ST-Link Debugger
- NULink Debugger

[4] Supported ARM cores: ARM7, ARM9, Cortex-M, Cortex-R

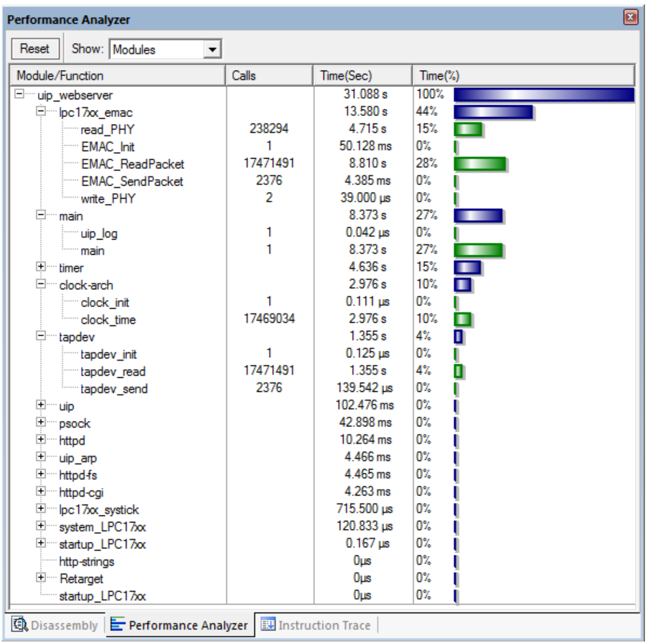
4.5.2 Real-Time Trace

The microcontroller from NXP includes ARMs "‘Cortex ETM Trace"’ peripheral. This enables Chip1768 to record every single step of the program code execution in Real-Time while the program is running. Advanced analysis of the controller application are made possible.

Among others, MDK-ARM is able to list, filter and sort all controller cycles:

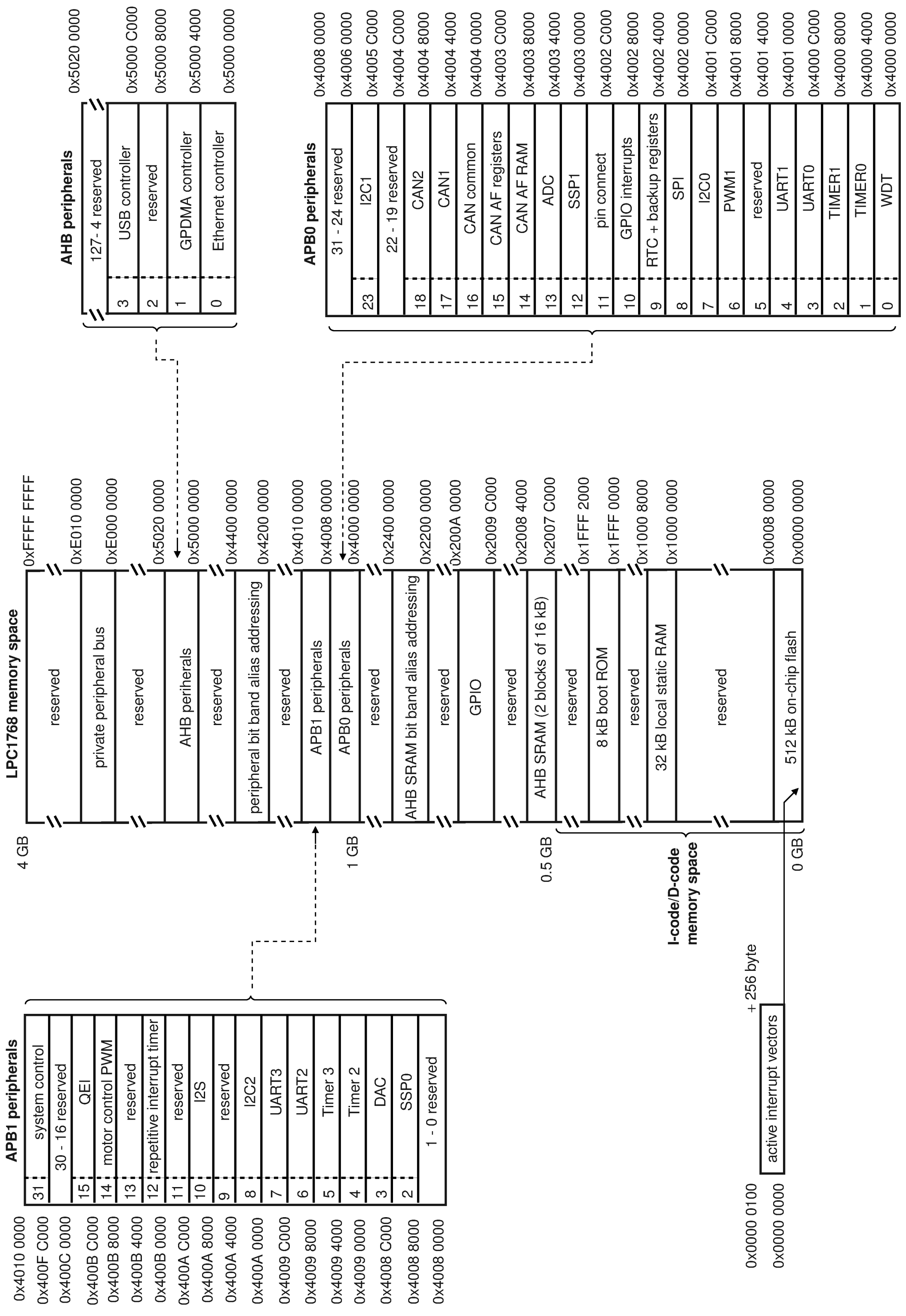


Another example of the powerful debugging features Real-Time Trace offers is the performanca analysis. MDK-ARM calculates which parts of the firmware application consume the most controller time allowing the developer to draw conclusions about software performance and controller load. Within seconds, problematic parts of the code can be identified:



5 Addendum

5.1 Memory Map



Quelle: Handbuch zum NXP LPC1768, Seite 13

5.2 Dimensional Drawing