NEC

User's Manual

*start*WARE-VR4181A

Starter Kit VR4181A

applies to: *start*WARE-GHS-VR4181A
*start*WARE-WinCE-VR4181A
*start*WARE-Linux-VR4181A

Document No. U16646EE2V0UM00
Date Published September 2003

Printed in Germany

- The information in this document is current as of 02.09, 2003. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC Electronics data sheets or data books, etc., for the most up-to-date specifications of NEC Electronics products. Not all products and/or types are available in every country. Please check with an NEC sales representative for availability and additional information.

- No part of this document may be copied or reproduced in any form or by any means without prior written consent of NEC Electronics. NEC Electronics assumes no responsibility for any errors that may appear in this document.

- NEC Electronics does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from the use of NEC Electronics products listed in this document or any other liability arising from the use of such NEC Electronics products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Electronics or others.

- Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of customer's equipment shall be done under the full responsibility of customer. NEC Electronics no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.

- While NEC Electronics endeavors to enhance the quality, reliability and safety of NEC Electronics products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in NEC Electronics products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment and anti-failure features.

- NEC Electronics products are classified into the following three quality grades: “Standard”, “Special” and “Specific”.

The "Specific" quality grade applies only to NEC Electronics products developed based on a customer-designated “quality assurance program” for a specific application. The recommended applications of NEC Electronics product depend on its quality grade, as indicated below. Customers must check the quality grade of each NEC Electronics product before using it in a particular application.

"Standard": Computers, office equipment, communications equipment, test and measurement equipment, audio and visual equipment, home electronic appliances, machine tools, personal electronic equipment and industrial robots.

"Special": Transportation equipment (automobiles, trains, ships, etc.), traffic control systems, anti-disaster systems, anti-crime systems, safety equipment and medical equipment (not specifically designed for life support).

"Specific": Aircraft, aerospace equipment, submersible repeaters, nuclear reactor control systems, life support systems and medical equipment for life support, etc.

The quality grade of NEC Electronics products is “Standard” unless otherwise expressly specified in NEC Electronics data sheets or data books, etc. If customers wish to use NEC Electronics products in applications not intended by NEC Electronics, they must contact NEC Electronics sales representative in advance to determine NEC Electronics 's willingness to support a given application.

Notes: **1.** " NEC Electronics" as used in this statement means NEC Electronics Corporation and also includes its majority-owned subsidiaries.

2. " NEC Electronics products" means any product developed or manufactured by or for NEC Electronics (as defined above).

M8E 02.10

Regional Information

Some information contained in this document may vary from country to country. Before using any NEC product in your application, please contact the NEC office in your country to obtain a list of authorized representatives and distributors. They will verify:

- Device availability
- Ordering information
- Product release schedule
- Availability of related technical literature
- Development environment specifications (for example, specifications for third-party tools and components, host computers, power plugs, AC supply voltages, and so forth)
- Network requirements

In addition, trademarks, registered trademarks, export restrictions, and other legal issues may also vary from country to country.

NEC Electronics Inc. (U.S.)
Santa Clara, California
Tel: 408-588-6000
800-366-9782
Fax: 408-588-6130
800-729-9288

NEC Electronics (Europe) GmbH
Duesseldorf, Germany
Tel: 0211-65 03 01
Fax: 0211-65 03 327

Sucursal en España
Madrid, Spain
Tel: 091- 504 27 87
Fax: 091- 504 28 60

Succursale Française
Vélizy-Villacoublay, France
Tel: 01-30-67 58 00
Fax: 01-30-67 58 99

Filiale Italiana
Milano, Italy
Tel: 02-66 75 41
Fax: 02-66 75 42 99

Branch The Netherlands
Eindhoven, The Netherlands
Tel: 040-244 58 45
Fax: 040-244 45 80

Branch Sweden
Taeby, Sweden
Tel: 08-63 80 820
Fax: 08-63 80 388

United Kingdom Branch
Milton Keynes, UK
Tel: 01908-691-133
Fax: 01908-670-290

NEC Electronics Hong Kong Ltd.
Hong Kong
Tel: 2886-9318
Fax: 2886-9022/9044

NEC Electronics Hong Kong Ltd.
Seoul Branch
Seoul, Korea
Tel: 02-528-0303
Fax: 02-528-4411

NEC Electronics Singapore Pte. Ltd.
Singapore
Tel: 65-253-8311
Fax: 65-250-3583

NEC Electronics Taiwan Ltd.
Taipei, Taiwan
Tel: 02-2719-2377
Fax: 02-2719-5951

NEC do Brasil S.A.
Electron Devices Division
Guarulhos, Brasil
Tel: 55-11-6465-6810
Fax: 55-11-6465-6829

Preface

Readers

This manual is intended for users who want to understand the functions of the *start*WARE-VR4181A Starter Kit.

Purpose

This manual presents the hardware manual of *start*WARE-VR4181A Starter Kit.

Organization

This system specification describes the following sections:

- Board Features
- Detailed Functional Description
- Board Operation

Legend

Symbols and notation are used as follows:

Weight in data notation	:	Left is high-order column, right is low order column
Active low notation	:	$\overline{xxx}$ (pin or signal name is over-scored) or /xxx (slash before signal name) or xxx# (hash after signal name)
Memory map address:	:	High order at high stage and low order at low stage
Note	:	Explanation of (Note) in the text
Caution	:	Item deserving extra attention
Remark	:	Supplementary explanation to the text
Numeric notation	:	Binary . . . xxxx or xxxB Decimal . . . xxxx Hexadecimal . . . xxxxH or 0x xxxx

Prefixes representing powers of 2 (address space, memory capacity)

K (kilo): $2^{10} = 1024$

M (mega): $2^{20} = 1024^2 = 1,048,576$

G (giga): $2^{30} = 1024^3 = 1,073,741,824$

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 System Requirements

Host PC	For Green Hills MULTI® 2000 Embedded Development Kit, a PC supporting Windows 9x, Windows 2000 or Windows NT is required. Pentium 133 MHz (at least), 32 MB of RAM, 256-color display (1024 × 768), mouse, CD-ROM drive and 60 Mbytes of free hard disk space are required to install the GHS compiler and debugger package. An x86 based PC with a mainstream Linux distribution (like SuSE, RedHat or Debian) is required to install and run the Elinos tool chain.
Host I/F	Serial (RS232C) interface capable to handle communication at 115200 baud and lower speeds.

1.2 Described Products

This document describes three products: *start*WARE-GHS-VR4181A, *start*WARE-WinCE-VR4181A and *start*WARE-Linux-VR4181A. All products are based on the same hardware platform. As far as possible they will be commonly referenced under the name *start*WARE-VR4181A. We will use the original product names only when required.

1.3 Package Contents

Please verify that you have received all parts listed in the package contents list attached to the *start*WARE-VR4181A package. If any part is missing or seems to be damaged, please contact the dealer from whom you purchased your *start*WARE-VR4181A.

Note: Updates to this User Manual, additional documentation and/or utilities for *start*WARE-VR4181A, if available, may be downloaded from the NEC WEB page(s) *http://www.nec.de/support*.

1.4 Related Documents

VR4181A User's Manual Hardware, NEC Doc. Number U16049EJ1V0UM00
VR4181A Data Sheet, NEC Doc. Number U16277EJ1V0DS00
VR4100 Series User's Manual Architecture, NEC Doc. Number U15509EJ2V0UM00
Green Hills Documentation is provided with the installation of the MULTI® software.
ELinOS documentation is included in the ELinOS package, that is a part of *start*WARE-Linux-VR4181A.

1.5 Used Abbreviations

There are some abbreviations used in this document, which may require additional information to be understood correctly.

- RFU - reserved for future use
- NC - not connected
- GND - ground
- GHS - Green Hills
- GNU - S/W supplied under General Public License (GLP)
- MIPS - Microprocessor without interlocked pipeline stages

Chapter 2 Board Features

*Start*WARE-VR4181A is a low-cost evaluation board for NEC's VR4181A 64-bit high-performance microprocessor. It allows evaluation of the processor's performance as well as potential system performance, because the VR4181A can also be operated together with its typical system environment.

The *start*WARE-VR4181A board has a VR4181A-131 processor soldered; processor speed is configurable via DIP-switch. The VR4181A is directly connected to 32 MByte of SDRAM, built up with 2 Samsung K4S281632 devices (total bus width 32-bit). The SDRAM supports a max. 65.5 MHz SDRAM clock. 32 MByte of (Intel Strata) flash are connected to the VR4181A memory bus directly. Mictor connectors are attached to the memory bus before the isolation buffer. A short summary of the board features is given below:

- VR4181A-131 CPU
- 32 MByte Flash
- 32 MByte SDRAM
- N-wire debug interface (via KEL connector as used with the Midas ICE)
- Serial interface on conventional SUB-D connector, buffered via RS232 line driver
- Ethernet Interface (SMC LAN91C111 controller) with RJ45 connector
- USB host and function interfaces available
- One Compact Flash slot available
- Analog inputs and output directly accessible on multipoint connector
- System bus and most other interfaces available on two extension connectors
- Mictor Connectors for measurement purpose
- 4 LEDs for lowest-level debugging purposes
- 10 DIP switches for selection of SDRAM clock, bus clock and other settings
- Single Voltage Power Supply

*start*WARE-WinCE-VR4181A and *start*WARE-Linux-VR4181A are additionally equipped with

- A 5.5" QVGA TFT display (NEC NL3224BC35-20) with back light power supply
- Touch screen panel
- Plastic housing

The common board software on both versions consists of three monitor programs:

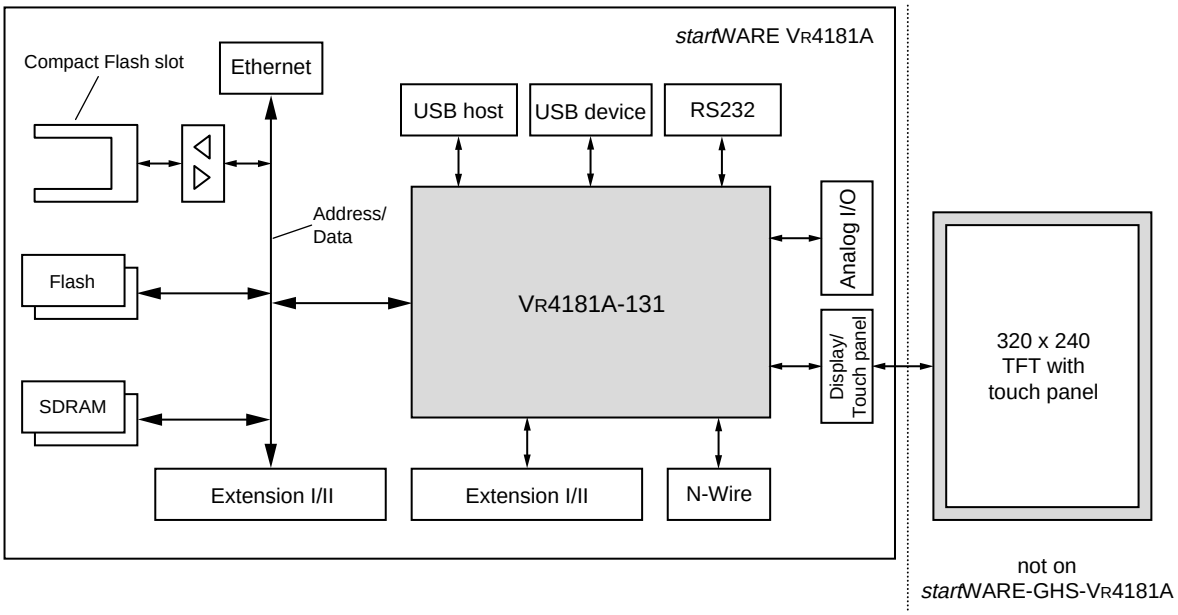
- A simple download monitor program named "S-Boot". The S-Boot monitor downloads files to the processor's SDRAM and/or Flash memory and optionally starts the program. A standard terminal emulation program can be used on the host PC side for communication.
- The "E-Boot" download monitor for high-speed downloads via Ethernet. E-Boot is identical to the WinCE Ethernet Bootloader.
- The Green Hills target monitor, that can communicate with the Green Hills Multi Compiler/ Debugger on the host PC.

Depending on the *start*WARE-VR4181A version you will also find the following software added:

- For *start*WARE-GHS-VR4181A an evaluation license of Multi 2000 is included in the package (on CD).
- On *start*WARE-WinCE-VR4181A a complete WinCE 4.1 operating system is pre-installed.
- On *start*WARE-Linux-VR4181A the ELinOS (Embedded Linux Operating System) V2.2 is pre-installed. The ELinOS tool chain is included in the package.

Note that the so-called “Platform Builder”, the integrated debug environment for WinCE, is **not** included in the *start*WARE-WinCE-VR4181A package.

Figure 2-1: Simplified Block Diagram



Chapter 3 Functional Description

The *start*WARE-VR4181A evaluation board is designed to evaluate the VR4181A MIPS RISC processor and uses the SMC LAN91C111 chip for ethernet downloading of software. It can also be used as development platform for operating systems like WinCE, VxWorks, Linux or Integrity. *Start*WARE-VR4181A can be used in a stand-alone mode using a 12 V AC adapter, but it can alternatively be extended with customer specific hardware using its extension connectors. It provides 32 MByte of SDRAM memory and 32 MByte of Flash memory with two 128 Mbit chips each.

The I/O capabilities of the VR4181A evaluation board include an on-chip UART, USB host and function interfaces, an Ethernet interface realized with the LAN9C111 controller, analog I/O, a colour LCD and touch panel interface as well as other VR4181A interfaces which can be accessed via the extension connectors. These boards are shipped with the NEC S-Boot and E-Boot monitors and an additional Green Hills monitor pre-installed in Flash memory. Furthermore WinCE 4.1 is installed on *start*WARE-WinCE-VR4181A and ELinOS V2.2 on *start*WARE-Linux-VR4181A.

3.1 Flash Memory

The VR4181A evaluation board has 32 MByte on-board Flash memory where the user can store data or transfer code to. The Flash memory is implemented using two 128 Mbit Intel Strata Flash 28F128J3A chips (IC7 and IC8). IC8 is connected to the lower 16 bits of the VR4181A data bus; IC7 is connected to the upper 16 bits. The system normally boots from the on-board Flash memory. Due to the VR4181A's flexible address assignment, almost 4 MByte of the complete Flash memory are used as boot memory; the remaining 28 MByte can be filled with application programs like WinCE or ELinOS. The upper 4 MByte cover an address range from 0xBFC0 0000 to 0xBFFF FFFF. After reprogramming the ROMCS register in the VR4181A the full 32 MByte range can be used as an address range from 0xBE00 0000 to 0xBFFF FFFF. Flash memory is accessed via the VR4181A chip selects signal ROMCS#. If the VR4181A is configured to 16-bit wide boot memory, only one half of the physically provided Flash memory can be used.

Write operations to the Flash memory by VR4181A program execution are possible; note that electrically the Flash memory is permanently write-enabled; therefore watch your step when writing to the ROM area.

3.2 Main Memory

The VR4181A Evaluation board has 32 MByte of main memory implemented in one physical bank and realized with two SAMSUNG K4S281632D 128 Mbit SDRAM chips connected to the SDCS0# chip select. They can be accessed with a maximum SDRAM clock frequency of 65.5 MHz. The physical address map for this memory is 0x0000 0000 to 0x01FF FFFF. The CPU address is 0xA000 0000 to 0xA1FF FFFF for uncached or 0x8000 0000 to 0x81FF FFFF for cached accesses.

3.3 Memory Mapping

The VR4181A uses a flexible memory mapping scheme that allows to move certain address blocks within the device's total addressing space. The 32-bit physical address space encompasses a total of 4 GByte; however these 4 GByte consist of 8 mirror images of a 512 MByte space subdivided in SDRAM, ROM and I/O spaces. The most significant address bits control, if the 512 MByte space is accessed cached or uncached respectively mapped or unmapped.

Figure 3-1 shows an example for the memory map of the *start*WARE-VR4181A board for uncached and unmapped addresses (kseg1) ranging from 0xA000 0000 to 0xBFFF FFFF. This configuration is used by the WinCE operating system, that is pre-installed on the VR4181A evaluation board. Note that the same structure is seen for other address space segments like kseg0 (0x8000 0000 to 0x9FFF FFFF). Consult the processor documentation for details.

3.4 VR4181A Communication Interfaces

*start*WARE-VR4181A has the most important communication interfaces of the VR4181A directly accessible on standard connectors: a UART, USB host and function, Compact Flash, display and touch panel. Additionally the N-Wire debug interface is wired out, however it is not mandatory to have an N-Wire based debugging tool available in order to operate the board.

VR4181A has several UARTs on chip that can be configured to work with different sets of control signals. From these UARTs UART0 in a 4-line configuration (Rx, Tx, CTS# and RTS#) is available on standard Sub-D connector X4. A Maxim MAX3232CUE level converter (IC11) ensures correct RS 232 signal levels at the connector. Note that jumper J1 must be inserted to operate UART0, as it is at the time of shipment.

The VR4181A processor provides USB 1.1 host and function interfaces. The USB host interface is accessible via a standard "Type A" USB connector. Its signal lines are directly connected to the VR4181A; USB power switching and over-current control is implemented with a Texas Instruments TPS2014 device (IC13). The USB function interface is wired to a standard "Type B" USB connector.

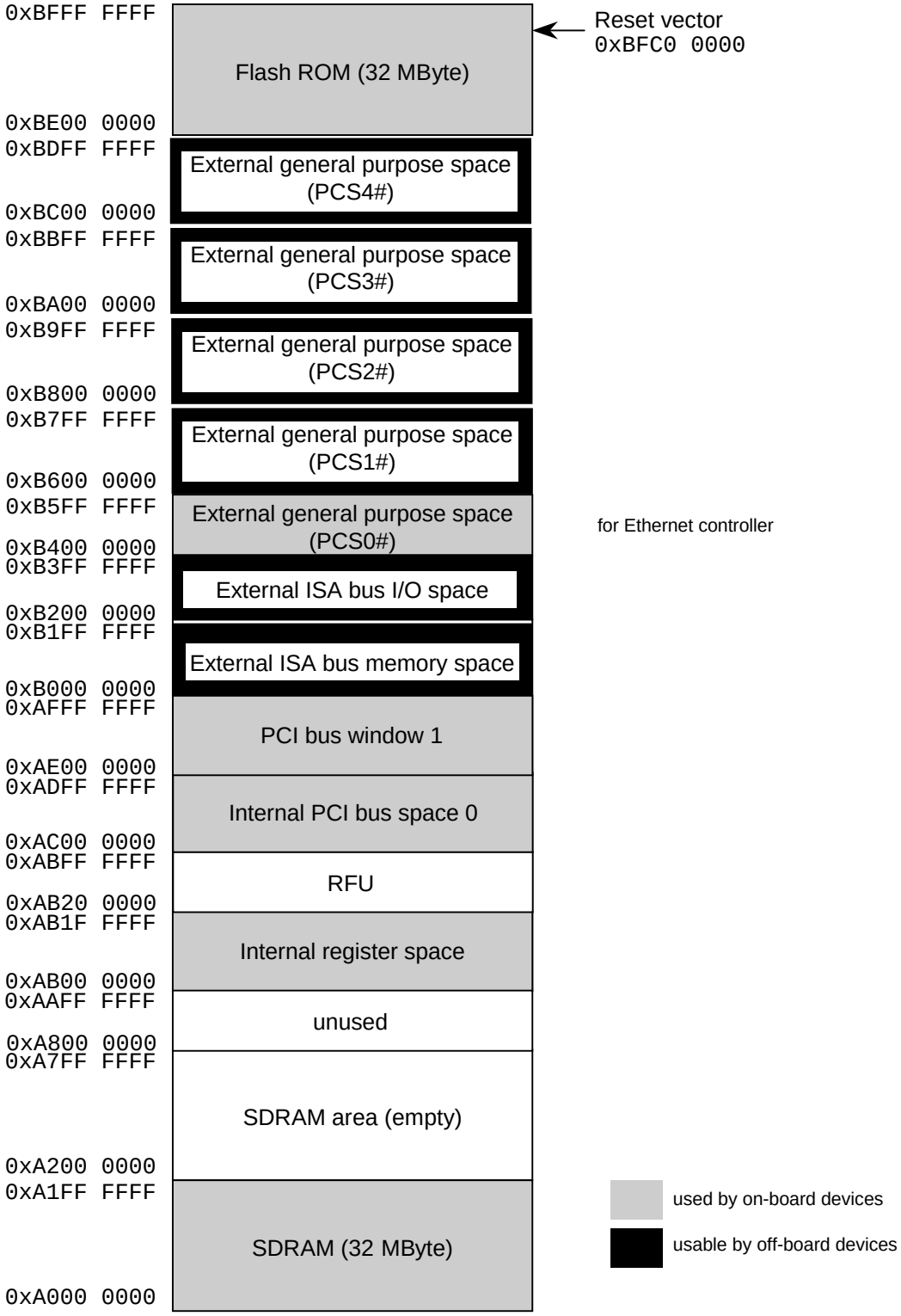
VR4181A supports two Compact Flash interfaces; one of these interfaces can be accessed using a standard CF card. Of course this interface can not only be used for flash memory extensions, but also for disk drives, modems, WLAN cards etc. under the assumption that the required driver software is available and installed. Electrically the CF slot is connected to the VR4181A address/data bus via bi-directional buffers (IC16, IC17) and to the CF control signals of the VR181A directly.

*start*WARE-VR4181A allows to operate an LCD display with a connector for the actual display signals and an additional connector for the back light inverter. Display connector X8 is prepared for a 64k-colour TFT display; however it is also possible to configure VR4181A to operate with cheaper displays. This may require an individual adapter between X8 and the actually used display. The same holds basically true for the back light connector X7. Most signals on X7 and X8 are directly connected to the VR4181A. Only the HSYNC line has a buffer which is required to operate the N-Wire interface **and** a display simultaneously.

The touch panel connector X9 has a very simple RC-network (R21, C58-61) for noise reduction. Apart from this it is directly connected to the VR4181A.

*start*WARE-VR4181A is equipped with an Ethernet interface that is implemented with a LAN91C111 controller from SMSC. This device is directly connected to the VR4181A address/data bus and it uses PCS0# as programmable chip select. From software point of view the Ethernet controller can be accessed in polling mode or interrupt mode. For interrupt mode the interrupt output of the LAN91C111 is connected to GPIO38 of the VR4181A. GPIO38 must then be configured as interrupt input. The connection can be de-activated by removing jumper J2 for applications that require to use GPIO38 in another function.

Figure 3-1: startWARE-VR4181A address map example for kseg1 (unmapped, uncached)



[MEMO]

Chapter 4 Detailed Functional Description

4.1 Usage of VR4181A GPIO Pins

Many of VR4181A GPIO pins are shared with other pin functions. This chapter explains, which of the GPIO pins have been used on the VR4181A evaluation board and which ones are still usable. Generally all unused GPIO pins have been wired to extension connector X12. Their exact position at connector X12 is explained in the respective chapter. With the VR4181A internal PINMODEn registers these pins can be configured to be either GPIO pins or to provide other interface functions. Please consult the VR4181A user manual for details.

The following table lists the usage of all GPIO pins; the freely usable pins (22 pins in total) are shown as shaded:

Table 4-1: VR4181A GPIO pin usage

GPIO0 - GPIO12	unused, wired to extension connector X12
GPIO13 – GPIO16	used for LEDs LED1 – LED4, additionally wired to extension connector X12
GPIO17	unused, wired to extension connector X12
GPIO18 – GPIO19	used for RS-232 serial interface
GPIO20 – GPIO23	unused, wired to extension connector X12
GPIO24 – GPIO36	used for CompactFlash, slot 0
GPIO37	unused, wired to extension connector X12
GPIO38	used, wired to INTR0 output of ethernet controller, if J2 is inserted
GPIO39	unused, wired to extension connector X12
GPIO40 – GPIO51	used for LCD display interface signals FPD4 - FPD15
GPIO52 – GPIO53	unused, wired to extension connector X12
GPIO54 – GPIO61	used as A15 to A22 for ROM address
GPIO62 – GPIO63	used as VPBIAS and VPLCD signals for LCD display control

4.2 Configuration Settings

The VR4181A evaluation board has two jumpers and ten DIP switches for configuration purposes. Their position should be checked carefully before powering up the board. Function and default position of jumpers and switches are explained in this chapter.

4.2.1 Jumper settings

Before power-on, the board should be configured through the jumpers J1 and J2. Jumper J1 connects an output of the serial driver IC11 to the CTS0# input of the VR4181A. This is required for proper operation of the RS 232 interface on the VR4181A evaluation board. The CTS0# input of the VR4181A can also be configured as GPIO19, a potential output. To avoid driver conflicts you have to remove J1, if you want to use the CTS0#/GPIO19 pin as GPIO.

Jumper J2 controls if the interrupt output of the LAN91C111 ethernet controller is directly connected to GPIO38 of the VR4181A. Applications that use the ethernet controller in an interrupt driven way, require this jumper set. Other applications, that require GPIO38 as an output, require this jumper removed.
An overview about the placement of the jumpers on the evaluation board is shown in figure 4-2 at the end of chapter 4.2. At time of shipment both jumpers are inserted by default.

4.2.2 Switch Settings

The VR4181A evaluation board has a 10-position DIP-switch SW1 which is used for various initial settings of the VR4181A processor. This chapter explains the functions and the default settings of SW1. An illustration of the switch and its default positions is given in Figure 4-1:

Figure 4-1: DIP switch setting for SW1 (default position)

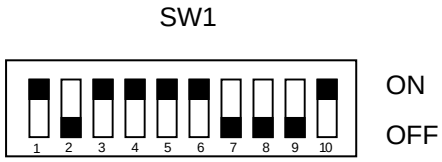


Table 4-2: DIP switch function assignment

SW1 switch position	Function
SW1-1	ROM bus width setting
SW1-2…4	pipeline clock speed setting
SW1-5	enable N-Wire usage
SW1-6	must always be switched on
SW1-7	must always be switched off
SW1-8	MIPS16 instruction set usage

(1) Boot ROM Bus Width Setting

The VR4181A processor uses the DBUS32 pin to control the width of the boot ROM. On the VR4181A evaluation board a 32-bit wide ROM configuration is provided; therefore this switch is positioned for 32-bit ROM width. Note, that if the switch is positioned for 16-bit ROM width all pre-programmed software will not work any more.

Table 4-3: Boot ROM bus width setting for VR4181A

SW1-1 setting	Function
ON	select 32-bit ROM bus width (default)
OFF	select 16-bit ROM bus width

(2) Pipeline Clock Setting

The VR4181A uses three pins (CLKSEL(2:0)), to configure the pipeline clock (AClock); these pins are sampled when the RTCRST# input is de-asserted. SW1-2…4 control the level of these pins and select the pipeline clock as described in the table below:

Table 4-4: Pipeline clock setting for VR4181A

SW1-2 (CLKSEL0)	SW1-3 (CLKSEL1)	SW1-4 (CLKSEL2)	AClock
OFF	OFF	OFF	73.7 MHz
ON	OFF	OFF	78.5 MHz
OFF	ON	OFF	84.1 MHz
ON	ON	OFF	90.7 MHz
OFF	OFF	ON	98.3 MHz
ON	OFF	ON	118 MHz
OFF	ON	ON	131.1 MHz
ON	ON	ON	RFU

(3) NWIREEN setting

The NWIREEN pin controls, if the on-chip debug logic on the VR4181A, which is accessed with the so-called “N-Wire” interface, can be used or not. Like with the CLKSEL pins the logical level of this pin is sampled during reset; it is selected with the SW1-5 switch.

Using the N-Wire interface requires connecting an N-Wire ICE to the connector X2; the default position for this switch is „ON“. There is normally no need to change this switch position, even if the N-Wire interface is not used.

Table 4-5: NWIREEN pin setting for VR4181A

SW1-5 setting	Function
ON	enable usage of N-Wire interface (default)
OFF	disable usage of N-Wire interface

(4) MIPS16EN setting

The VR4181A processor supports the MIPS16 instruction set extension. To allow the processor to switch to MIPS16 mode, the MIPS16EN must be pulled high during reset. This is done with DIP switch SW1-8 set to ON. By default MIPS16 is disabled.

Table 4-6: MIPS16EN pin setting for VR4181A

SW1-8 setting	Function
ON	enable usage of MIPS16 instruction set
OFF	disable usage of MIPS16 instruction set (default)

(5) DIVMODE Setting

With the CLKSEL(2:0) pins the VR4181A can be configured to various pipeline clock speeds. Additionally the external bus clock for SDRAM accesses and for the internal bus (named TClock) can be set to 1/2 or 1/3 of the pipeline clock speed. The DIVMODE(1:0) pins are used for this purpose; their setting is controlled with DIP switches SW1-9 and SW1-10.

Table 4-7: Bus clock division mode setting for VR4181A

SW1-9 (DIVMODE0)	SW1-10 (DIVMODE1)	bus clock division mode
OFF	OFF	RFU
ON	OFF	Div3 mode (TClock = 1/3 AClock)
OFF	ON	Div2 mode (TClock = 1/2 AClock, default)
ON	ON	RFU

4.3 Push Button Switches

The VR4181A evaluation board has three push button switches, which are connected to VR4181A inputs. S3 generates a low level pulse on the RTCRST# input and causes a cold reset on the VR4181A. This resets the processor completely.

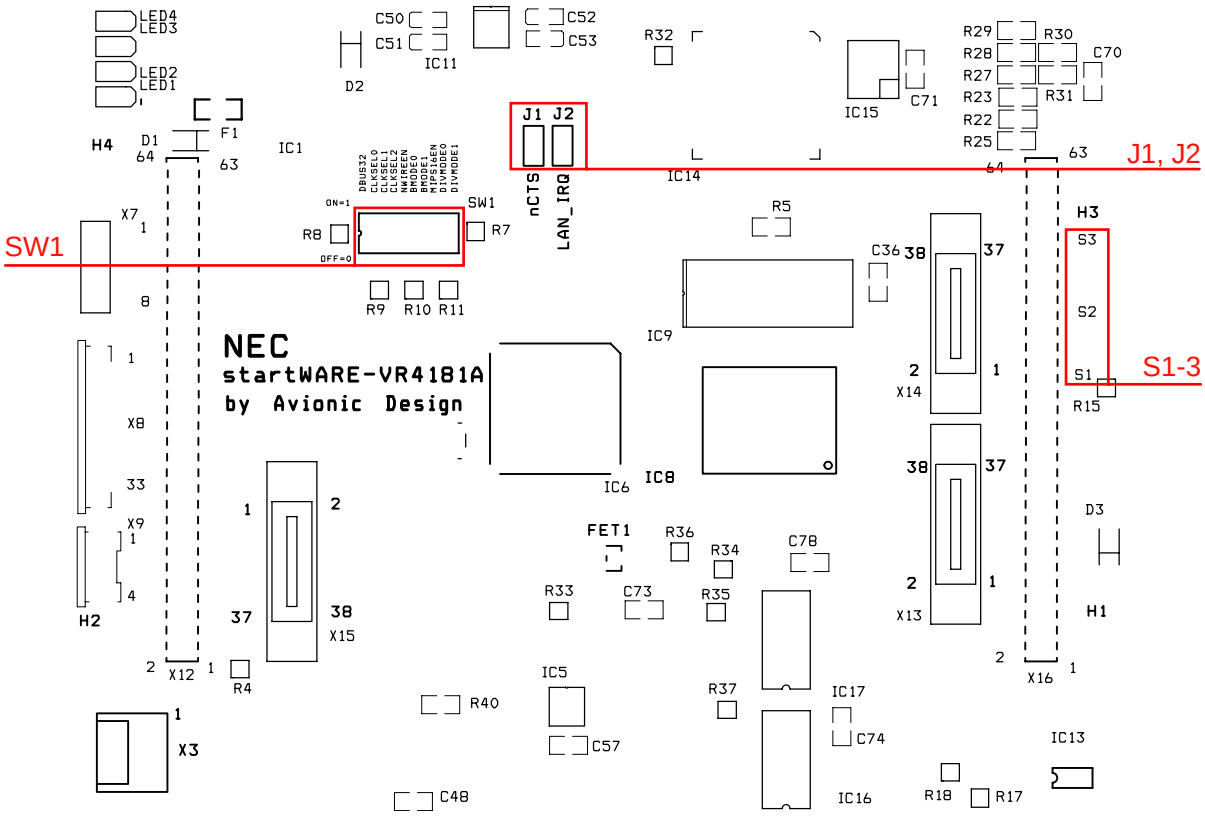
S2 generates a low level pulse on the RSTSW# input, which issues a reset as well however leaves a couple of internal VR4181A settings (parts of registers in the RTC, GIU, PWM and PMU units) unchanged. (Please consult the VR4181A user manual for details.)

Push button switch S1 is connected to the POWER input of the VR4181A. If the VR4181A has been driven into the hibernate mode by software, S1 will restart the processor.

Table 4-8: Push button switch function assignment

Number	Function
S1	POWER (activation from hibernate mode)
S2	Warm Reset (connected to the RSTSW# input)
S3	Cold Reset (connected to the RTCRST# input)

Figure 4-2: Jumper and Switch Positions



4.4 Connectors

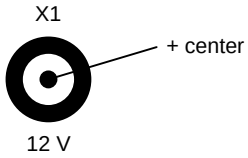
The VR4181A evaluation board provides several connectors for testing purpose and for system extensions. This chapter will describe usage of the connectors one by one.

4.4.1 Power Supply Connector (X1)

The power supply connection is done with a conventional “coaxial” type connector with the positive line at the centre. The power consumption of the board is approximately 800 mA @ 12 V with the LCD display activated and no external hardware connected. The power supply that is delivered together with the board is able to provide1.25 A @ 12 V.

If you would like to use other power supplies, be sure to supply 12 V, if the board is operated with display. If the board is operated without display, lower voltages can be applied as well.

Figure 4-3: Power supply connector (X1)

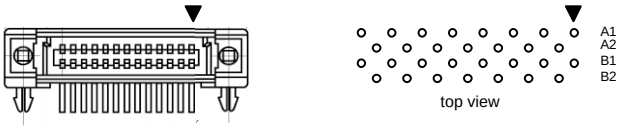


Caution: We urgently recommend to check polarity of the power supply that comes together with the board before connecting it to X1.

4.4.2 Usage of the N-Wire ICE Connector (X2)

For using the N-Wire ICE (RTE-1000-TP) connect the ICE to the N-Wire connector X2.

Figure 4-4: N-Wire Connector X2



X2 is directly connected to the VR4181A N-Wire-Interface. Note that the DIP switch SW1-5 must be set to “ON”, if an N-Wire ICE is used. The pin assignment for the connector is as follows:

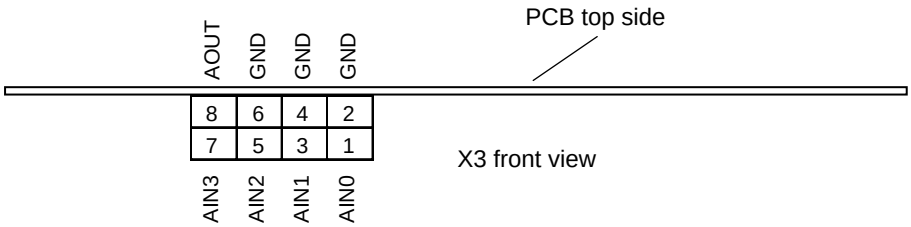
Figure 4-5: N-Wire connector X2 pin assignment

GND	A1	B1	GND
GND	A2	B2	GND
GND	A3	B3	GND
GND	A4	B4	GND
GND	A5	B5	GND
GND	A6	B6	GND
JTDI	A7	B7	GND
JTCK	A8	B8	GND
JTMS	A9	B9	GND
JTDO	A10	B10	GND
nJTRST	A11	B11	NC
nBKTGIO	A12	B12	NC
NC	A13	B13	VDD3.3

4.4.3 Analog I/O connector X3

The VR4181A processor has four A/D input channels and a single D/A output channel for simple measurement and signalling functions. These analog I/Os are made available on a multipoint connector, which is directly wired to the respective VR4181A pins. So there is no amplification nor filtering provided on the board. The connection scheme for the analog I/O is shown below:

Figure 4-6: Analog I/O connector X3 pin assignment



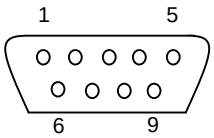
4.4.4 Using the Serial Interface (X4)

The VR4181A processor has three on-chip serial interfaces, called SIU0, SIU1 and SIU2 (**s**erial **i**nterface **u**nit). The SIUn interfaces conform to the RS232-C communication standard and support up to 1.15 Mbps. The units are functionally compatible with the NS16550. For details refer to the VR4181A Users Manual. The different SIUs provide different numbers of control and handshake signals. SIU0 can be operated as a full 7-wire interface; however on the VR4181 evaluation board it is configured as a 4-wire interface.

The processor's serial interface SIU0 is connected to a RS232-C electrical interface via a level converter. It uses a standard male 9-pin SUB/D connector X4 on the CPU PCB. The pin definitions are described in the following figure.

Figure 4-7: Serial interface connector X4

Pin	Function
1	NC
2	RX0
3	TX0
4	NC
5	GND
6	NC
7	nRTS0
8	nCTS0
9	NC

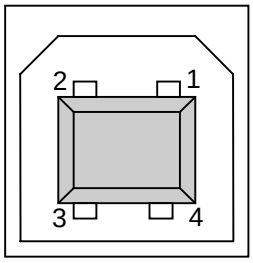


front view

4.4.5 USB Device Connector X5

The VR4181A evaluation board has a standard “Series B” USB device receptacle, which is directly connected to the USB device interface on the VR4181A. The pinning of this receptacle X5 is shown below:

Figure 4-8: USB device connector X5 pin assignment

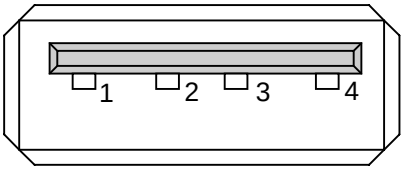


Pin	Function
1	NC
2	D-
3	D+
4	GND

4.4.6 USB Host Connector X6

The VR4181A evaluation board has a standard “Series A” USB host receptacle, which is connected to the USB device interface on the VR4181A. The USB power supply is provided through a TPS2014 USB power controller, The pinning of this receptacle X6 is shown below:

Figure 4-9: USB Host connector X6 pin assignment



Pin	Function
1	Vbus
2	D-
3	D+
4	GND

4.4.7 LCD Display Connectors X7, X8 and X9

A total of three connectors is provided for the LCD display on the bottom side of the PCB. X7 is an 8-pin, single-in-line connector for the high-voltage converter for the LCD back light. X8 is a 33-pin ZIF connector for the display itself and last but not least. X9 is a 4-pin ZIF connector for the touch panel. The Figure 4-10 with the pin assignments uses the signal names as they are used in the circuit diagram.

Figure 4-10: LCD display connectors X7, X8, X9 pin assignment

High-voltage converter connector X7

Pin	Function
1	V12
2	V12
3	GND
4	GND
5	VPLCD
6	VPBIAS
7	GND
8	NC

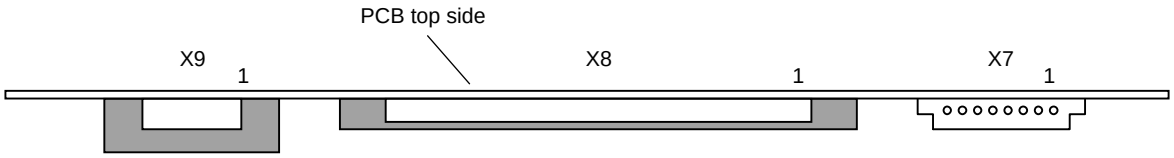
LCD display connector X8

Pin	Function
1	GND
2	DCLK
3	HSYNC
4	VSYNC
5	GND
6	GND
7	R1
8	R2
9	R3
10	R4
11	R5
12	GND
13	G0
14	G1
15	G2
16	G3
17	G4
18	G5
19	GND
20	GND
21	B1
22	B2
23	B3
24	B4
25	B5
26	GND
27	ENAB
28	VDIS
29	VDIS
30	GND
31	GND
32	GND
33	GND

Touch panel connector X9

Pin	Function
1	TPX1
2	TPY1
3	TPX0
4	TPY0

Figure 4-11: LCD display connector (X7, X8, X9) front view

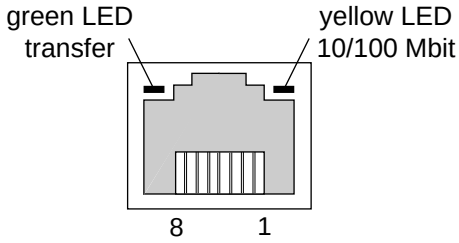


4.4.8 Ethernet Connector X10

The VR4181A evaluation board provides a standard RJ45 ethernet connector with integrated status LEDs. The green LED indicates an ongoing transfer; the yellow LED indicates the speed of the currently established Ethernet connection (LED on: 10 Mbps; LED off: 100 Mbps).

Figure 4-12: Ethernet connector X10 pin assignment

Pin	Function
1	Tx+
2	Tx-
3	Rx+
4	NC
5	NC
6	Rx-
7	NC
8	NC



4.4.9 CompactFlash Connector X11

The VR4181A supports a maximum of two CompactFlash slots; one of them (CF0) is made available on CompactFlash connector X11. If required, a second CompactFlash slot can be realized with external hardware. Connector X11 provides a 3.3 V slot.

Figure 4-13: CompactFlash connector X11 pin assignment (as originally specified)

nCD1	26	1	GND1
D11	27	2	D03
D12	28	3	D04
D13	29	4	D05
D14	30	5	D06
D15	31	6	D07
nCE2	32	7	nCE1
nVS1	33	8	A10
nIORD	34	9	nOE
nIOWR	35	10	A09
nWE	36	11	A08
RDY/IREG	37	12	A07
VCC2	38	13	VCC1
nCSEL	39	14	A06
nVS2	40	15	A05
RESET	41	16	A04
nWAIT	42	17	A03
nINPACK	43	18	A02
nREG	44	19	A01
BVD2	45	20	A00
nSTSCHG	46	21	D00
D08	47	22	D01
D09	48	23	D02
D10	49	24	WP/nIOIS16
GND2	50	25	nCD2

Caution: The board prototypes (with serial numbers CA20Y0001D – CA20Y0004D) have been layouted with a wrong footprint for connector X11 and the pin assignment is not as shown in Figure 4-13. As a consequence CompactFlash cards must be inserted upside down on these boards. The mechanical protection, that normally prevents this, has been removed on the prototypes.

4.4.10 Extension Connectors X12, X16

The VR4181A Evaluation Board has two extension connectors X12 and X16 which can be used for customer specific add-on hardware. The processors' address and data bus together with control and chip select signals are available on these connectors. They also allow to use the VR4181A interfaces that are not directly wired to the other special function connectors.

Note that the VR4181A has to be configured accordingly for using these additional interfaces.

Figure 4-14: Extension connector X12, X16 pin assignment (1/2)

(a) Extension Connector X12 (seen from the bottom of the PCB)

TxD2/IRDOUT/MIPS16EN	64	63	RxD2/IRDIN
nDTR2/SDATAOUT/SDO/DIVMODE0	62	61	nDCD2/SDATAIN/SDI
nDSR2/nSRESET	60	59	nRTS2/SYNC/WS/DIVMODE1
nCTS2/BITCLK/SCLK	58	57	nDTR0/nRTS1/GPIO17/CLKSEL0
nDCD0/GPIO16	56	55	nDSR0/nCTS1/GPIO15
RxD1/SCL1/GPIO14	54	53	TxD1/SDA1/GPIO13
nCTS0/GPIO18	52	51	nRTS0/GPIO19/CLKSEL1
SCK/KSCAN11/GPIO23	50	49	SI/KSCAN10/GPIO22
SO/KSCAN9/GPIO21	48	47	FRM/KSCAN8/GPIO20
PWM0/KSCAN7/GPIO8	46	45	PWM1/KSCAN6/GPIO9
PWM2/KSCAN5/GPIO10	44	43	nCF1_VCCEN/KSCAN4/GPIO37
KSCAN3/GPIO3	42	41	KSCAN2/GPIO2
KSCAN1/GPIO1	40	39	KSCAN0/GPIO0
SCL0/KPORT7/GPIO12	38	37	SDA0/KPORT6/GPIO11
nCF1_EN/KPORT5/GPIO38	36	35	CF1_DIR/KPORT4/GPIO39
KPORT3/GPIO7	34	33	KPORT2/GPIO6
KPORT1/GPIO5	32	31	KPORT0/GPIO4
GND	30	29	GND
GND	28	27	GND
GPIO53/nTC1	26	25	GPIO52/nTC0
nDAK1	24	23	nDAK0
nDRQ1	22	21	nDRQ0
nNMI	20	19	nSYSEN
SYSDIR	18	17	nROMCS
nPCS4	16	15	nPCS3
nPCS2	14	13	nPCS1
nPCS0	12	11	DQM3/nLBE3
DQM2/nLBE2	10	9	DQM1/nLBE1
DQM0/nLBE0	8	7	nUBE
nIOCS16	6	5	IORDY
nIOWR	4	3	nIORD
nWR	2	1	nRD

Figure 4-14: Extension connector X12, X16 pin assignment (2/2)

(b) Extension Connector X16 (seen from the bottom of the PCB)

V5.0	64	63	V5.0
TPI-	62	61	TPI+
TPO-	60	59	TPO+
MPWR	58	57	A24/CKE1
A23/nRP	56	55	A22/GPIO61
A21/GPIO60	54	53	A20/GPIO59
A19/GPIO58	52	51	A18/GPIO57
A17/GPIO56	50	49	A16/GPIO55
A15/GPIO54	48	47	A14
A13	46	45	A12
A11	44	43	A10
A09	42	41	A08
A07	40	39	A06
A05	38	37	A04
A03	36	35	A02
A01	34	33	A00
D31	32	31	D30
D29	30	29	D28
D27	28	27	D26
D25	26	25	D24
D23	24	23	D22
D21	22	21	D20
D19	20	19	D18
D17	18	17	D16
D15	16	15	D14
D13	14	13	D12
D11	12	11	D10
D09	10	9	D08
D07	8	7	D06
D05	6	5	D04
D03	4	3	D02
D01	2	1	D00

4.4.11 Logic Analyzer Connectors X13, X14 and X15

For debug purposes the VR4181A evaluation board is equipped with three Mictor connectors for easy hook up of HP logic analysers. The pinning of these connectors is illustrated below.

Figure 4-15: Logic analyzer connectors X13, X14, X15 pin assignment

Mictor connector X13 (seen from the bottom of the PCB)

NC	38	19	NC
SDCLK	37	18	nWE
D31	36	17	D30
D29	35	16	D28
D27	34	15	D26
D25	33	14	D24
D23	32	13	D22
D21	31	12	D20
D19	30	11	D18
D17	29	10	D16
D15	28	9	D14
D13	27	8	D12
D11	26	7	D10
D09	25	6	D08
D07	24	5	D06
D05	23	4	D04
D03	22	3	D02
D01	21	2	D00
NC	20	1	NC

Mictor connector X14 (seen from the bottom of the PCB)

NC	38	19	NC
IORDY	37	18	CKE0
nIORD	36	17	nIOWR
nRAS	35	16	nCAS
nRD	34	15	nWR
nSDCS0	33	14	A24
A23	32	13	A22
A21	31	12	A20
A19	30	11	A18
A17	29	10	A16
A15	28	9	A14
A13	27	8	A12
A11	26	7	A10
A09	25	6	A08
A07	24	5	A06
A05	23	4	A04
A03	22	3	A02
A01	21	2	A00
NC	20	1	NC

Mictor connector X15 (seen from the bottom of the PCB)

NC	1	20	NC
nIOCS16	2	21	SA10
SYSDIR	3	22	nSYSEN
DQM0	4	23	DQM1
DQM2	5	24	DQM3
nDRQ0	6	25	nDRQ1
nDAK0	7	26	nDAK1
nTC0	8	27	nTC1
nSDCS1	9	28	nSDCS2
nSDCS3	10	29	nUBE
CLK48	11	30	UPON
UHDP	12	31	UHDN
UDP	13	32	UDN
TPO+	14	33	TPO-
TPI+	15	34	TPI-
nROMCS	16	35	nPCS0
nPCS1	17	36	nPCS2
nPCS3	18	37	nPCS4
NC	19	38	NC

[MEMO]

Chapter 5 Board Operation

This chapter explains practical usage of the *start*WARE-VR4181A evaluation board. As board operation is slightly different for the WindowsCE and the Green Hills version of the VR4181A evaluation board, both versions will be treated separately in this chapter.

5.1 Getting Started

In this chapter the very first steps for a “vitality” test of *start*WARE VR4181A will be described.

5.1.1 *start*WARE-WinCE-VR4181A

This chapter gives a “step-by-step” description, how to start up the board in the *start*WARE-WinCE-VR4181A version, that has WinCE pre-installed.

- Check if all jumpers are in their default position
- Check if all DIP switches are in their default position
- Check power supply plug polarity
- Connect the power supply to the board
- Push the cold reset button (S3)

WindowsCE will then pop-up after a while and is ready to use.

Note: Like on the *start*WARE-GHS-VR4181A board, the first program, that is started, is the S-Boot monitor. S-Boot waits 5 seconds for an input from the terminal; if no input is given, WinCE will be started. Therefore the start-up time for WinCE is relatively long; however this is not caused by the VR4181A or the operating system itself.

5.1.2 *start*WARE-Linux-VR4181A

This chapter gives a “step-by-step” description, how to start up the board in the *start*WARE-Linux-VR4181A version, that has Linux pre-installed.

- Check if all jumpers are in their default position
- Check if all DIP switches are in their default position
- Check power supply plug polarity
- Connect the power supply to the board
- Push the cold reset button (S3)

Linux will then pop-up after a while and is ready to use.

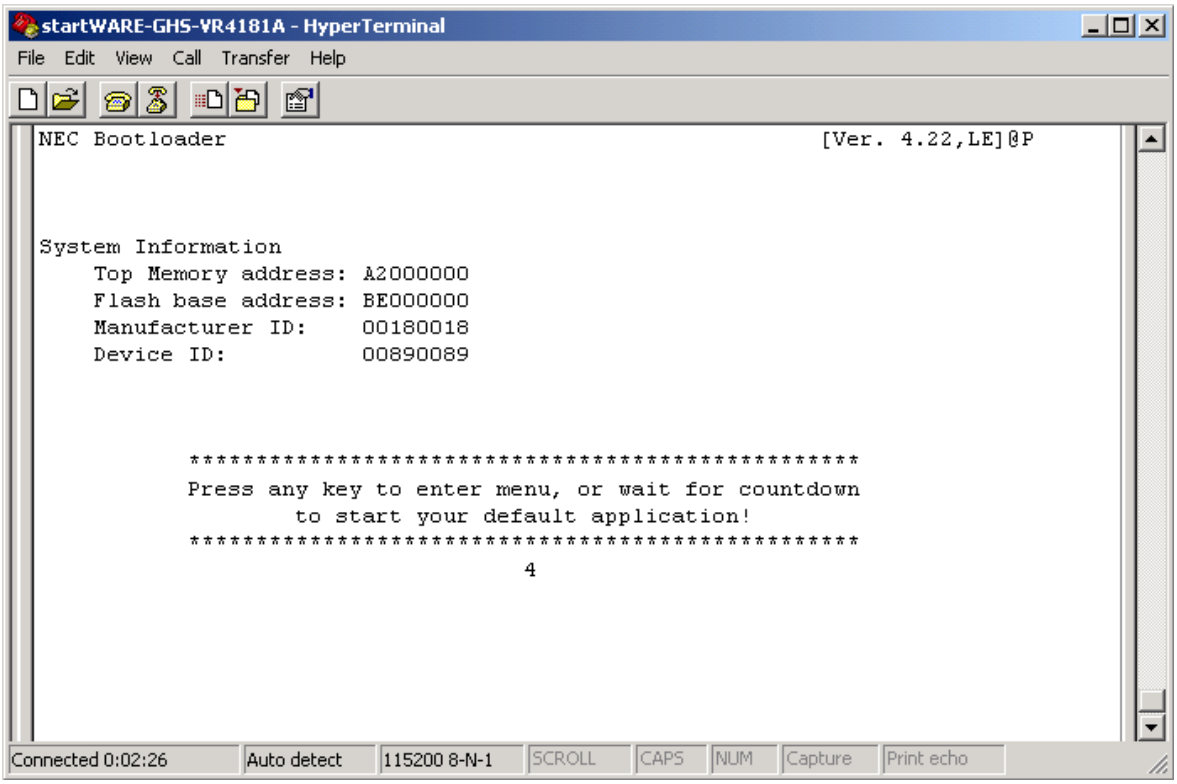
Note: Like on the *start*WARE-GHS-VR4181A board, the first program, that is started, is the S-Boot monitor. S-Boot waits 5 seconds for an input from the terminal; if no input is given, Linux will be started. Therefore the start-up time for Linux is relatively long; however this is not caused by the VR4181A or the operating system itself.

5.1.3 *start*WARE-GHS-VR4181A

This chapter gives a “step-by-step” description, how to start up the board in the *start*WARE-GHS-VR4181A version, which has no operating system pre-installed.

- Check if all jumpers are in their default position
- Check if all DIP switches are in their default position
- Connect the *start*WARE-GHS-VR4181A evaluation board with a serial cable to a host PC with a terminal emulation program (115200 baud, 8 bit, no parity, 1 stop bit, no handshake)
- Check the power supply plug polarity
- Connect the power supply to the board
- Start the terminal emulation program
- Push the cold reset button (S3)
- LED1 will be activated and the following welcome message from the NEC bootloader program “S-Boot” will be displayed in the terminal emulation window

Figure 5-1: Startup Screen



When the countdown has expired, S-Boot will automatically invoke the Green Hills debug monitor to which you can connect after starting Multi.

5.2 S-Boot Operation

5.2.1 Features

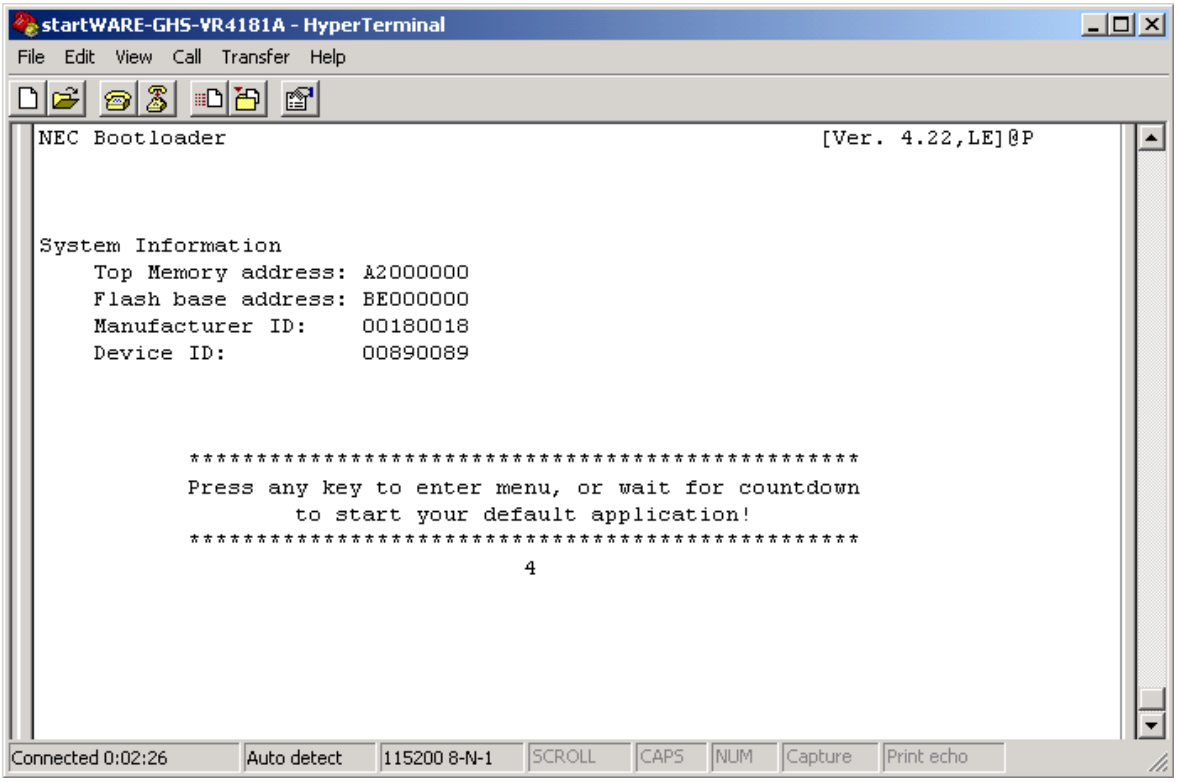
A simple boot monitor named S-BOOT is implemented on both versions of the VR4181A evaluation board. This monitor offers the following features:

- A fast serial connection using the CPUs SIU0 at 115200 Baud,
- Download capabilities to SDRAM,
- Download to FLASH,
- Automatic, programmable start of applications available in RAM or Flash,
- A small Flash File System with simple directory structure.

5.2.2 S-Boot Startup Message

Once the VR4181A evaluation board is powered up and turned on, you will find this screen on the host PC:

Figure 5-2: Startup Screen



The screen gives information about the current system configuration in the upper right hand corner, please see Figure 5-2.

Remarks: **1.** Version information (Example - 4.22)

2. Endianess:
LE - Little Endian,
BE - Big Endian

3. Run mode:
@ - start from Flash
* - start from RAM

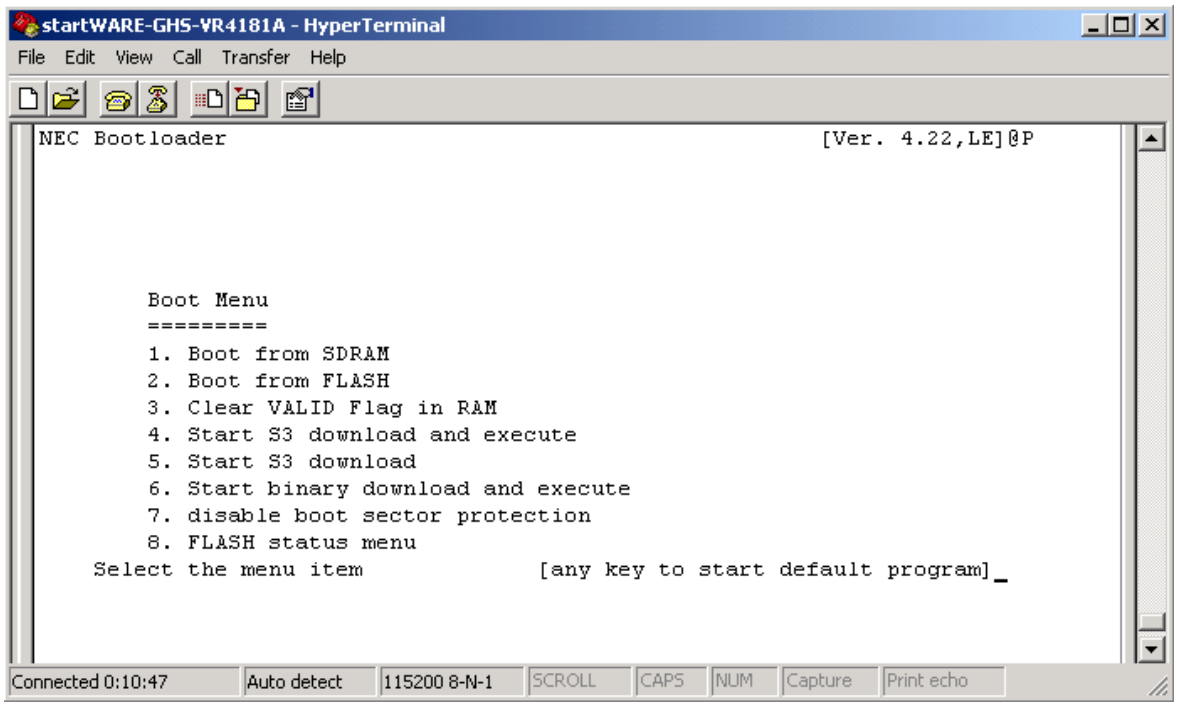
4. Protection settings:
P - Protected,
N - Not protected

The running countdown can be stopped by any input. Once the countdown reaches 0, the system will try to boot. In the first instance, it will try booting from SDRAM, if any information is available that there is something loaded. If nothing is found, it continues to check for information in the Flash memory. If there is an application available in any of the Flash locations, it starts this application. Please see Figure 5-6, “startWARE-GHS-VR4181A Boot Flow Chart,” on page 44 for a more detailed description of the procedure.

5.2.3 S-Boot Boot Menu

When the countdown is interrupted, the S-BOOT stops the boot procedure and the boot menu is shown:

Figure 5-3: S-Boot Monitor Boot Menu



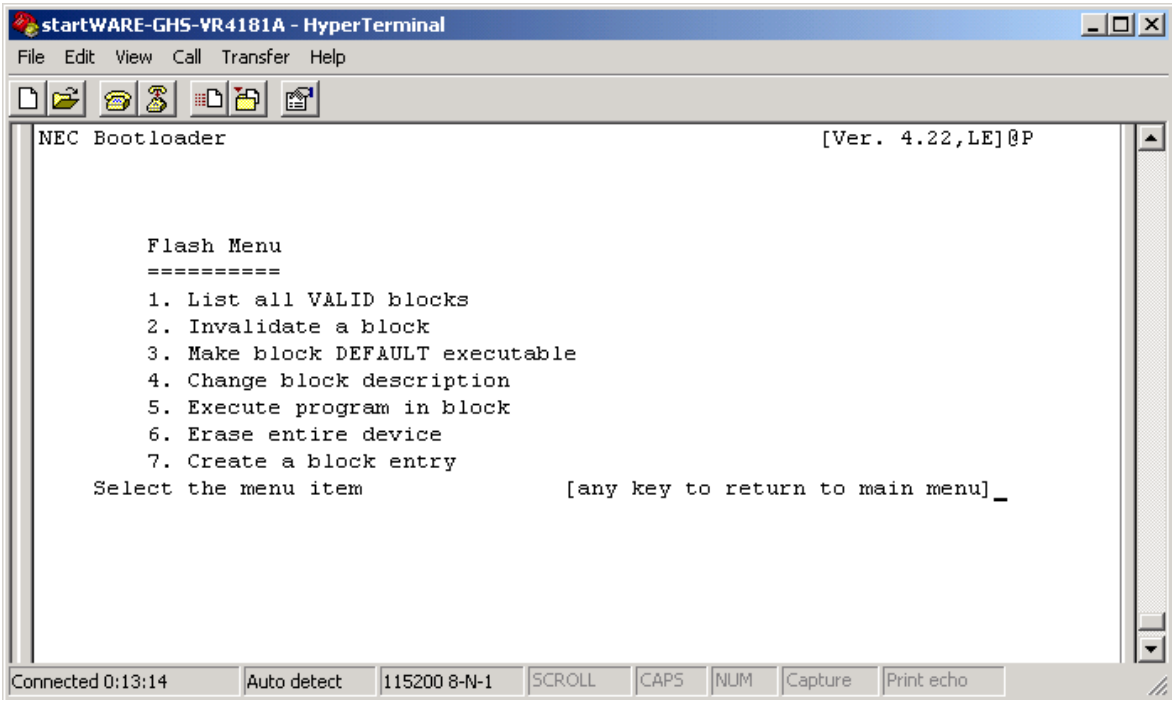
This is the main menu of the system and provides access to various other options of the S-BOOT. The user may manually boot from RAM or FLASH.

1. Boot from SDRAM
   SBOOT tries to boot from memory location most recently loaded with an application.
2. Boot from Flash
   SBOOT tries to boot from a valid Flash memory location available in the directory list and marked as ‘default’ start-up.
3. Clear VALID Flag in RAM
   Makes an application located in main memory unavailable for next boot process.
4. Start S3 download and execute
   Implies the option to download an S-record type file and start it directly after download.
5. Start S3 download
   Behaves like item 4, but does not automatically start the application.
6. Start binary download and execute
   Using this option starts a download of a raw binary file, recommended for larger images. Once this option is invoked, the user is asked for
   a) a flash memory address, where the application is downloaded to,
   b) a start address of the application.
   This is necessary for images providing a different entry point than the first target address is.
7. Disable boot sector protection
   This option allows you to overwrite the current S-Boot image in flash and is only used for S-Boot updates. Normally the boot sector should always be protected. The option is disabled on *start*WARE-VR4181A.
8. Flash status menu
   Opens a submenu handling the various Flash options.

5.2.4 Flash Option menu

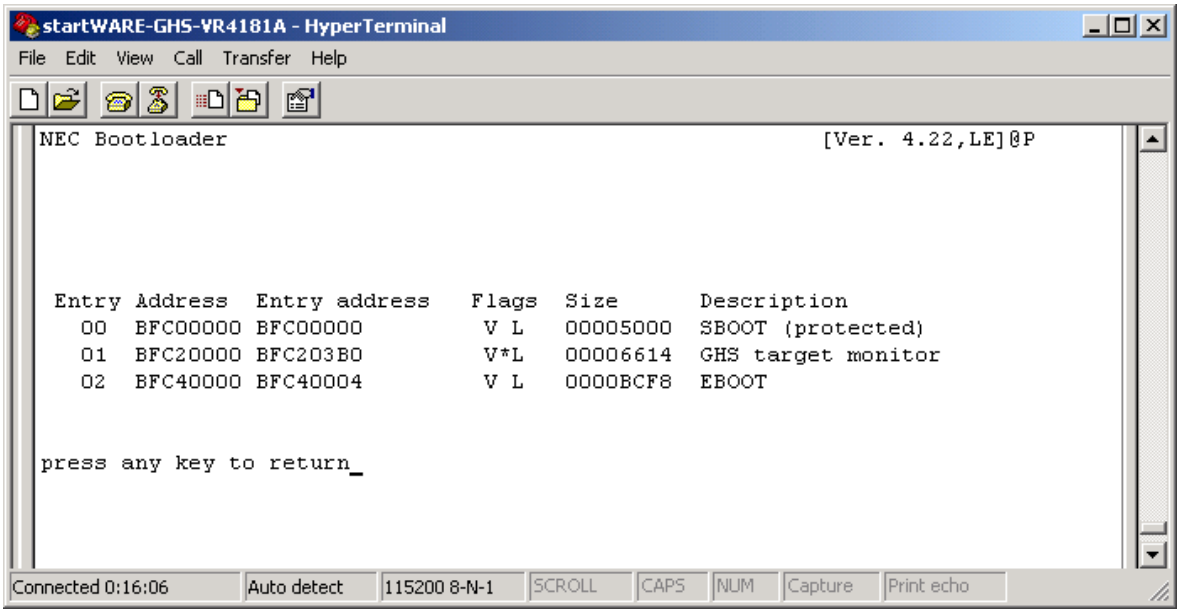
There are several options available making the handling of the Flash memory easier. Several target applications may reside in Flash memory and can be managed by S-BOOT.

Figure 5-4: S-Boot Flash Menu



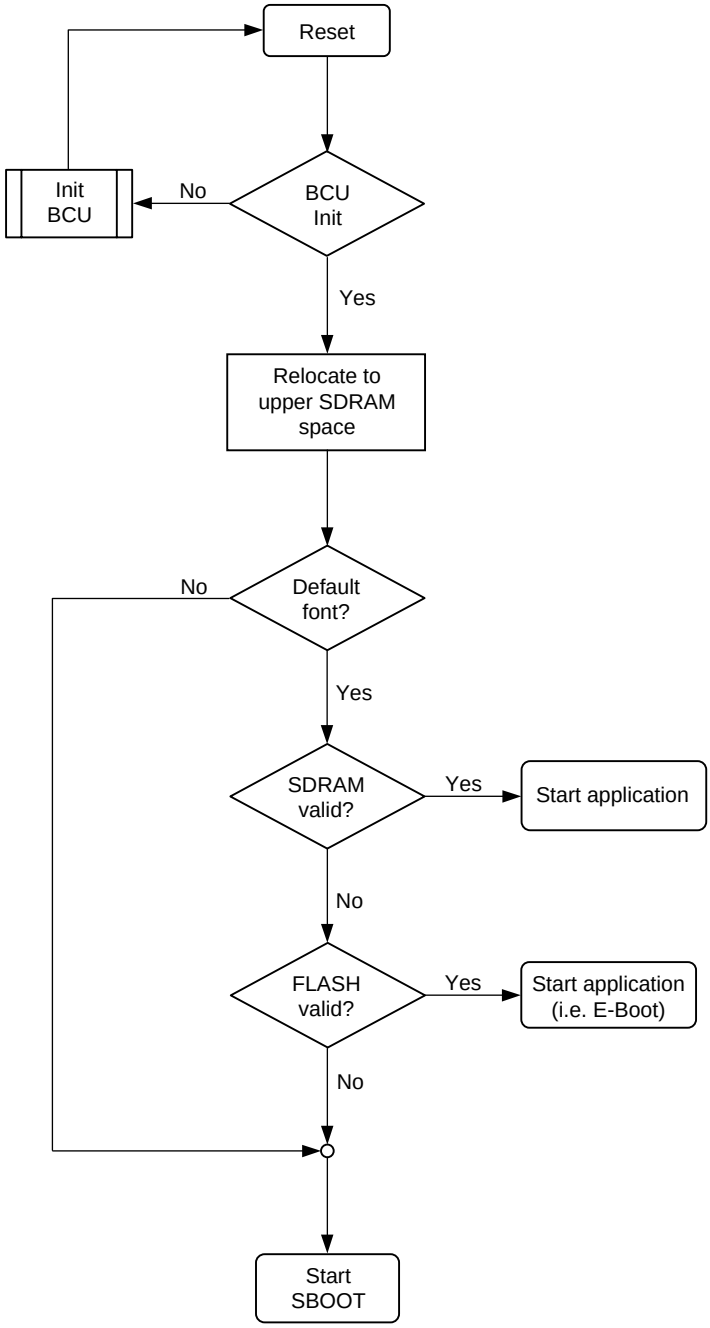
1. List all VALID blocks
   Displays a list of applications available in the Flash memory. The default screen on *start*WARE-GHS-VR4181A looks as shown in Figure 5-5, “S-Boot Directory List for startWARE-GHS-VR4181A,” on page 43.
2. Invalidate a block
   Disables the directory entry and makes the application unavailable for the next boot process.
3. Make block DEFAULT executable
   Gives the option to boot an application from a block specified by the user (after the S-Boot count-down has expired). The application must have a valid entry in the directory block structure, before it can be started by default boot process. Here, the GHS target monitor is the default application to start with. The least recently downloaded application to Flash will automatically get the DEFAULT boot flag. This menu entry gives the user the option to change it manually.
4. Change block description
   Allows to change the block description of a downloaded application, if the automatically provided name is not sufficient. It is a simple text entry to name that directory entry into a more specific description. A maximum number of 32 characters is allowed here.
5. Execute program in block
   Offers the option to start any of the available applications manually.
6. Erase entire device
   The entire FLASH device is erased, except for the protected boot block. All applications are lost and the directory entry list is emptied as well.
7. Create a block entry
   Allows manual entry of start and entry address of a program block. This way become necessary after download with a software other than S-boot.

Figure 5-5: S-Boot Directory List for startWARE-GHS-VR4181A



5.2.5 Reset Process Flow

Figure 5-6: startWARE-GHS-VR4181A Boot Flow Chart



5.2.6 S-BOOT Specification

(1) Interrupts

S-BOOT does not use any interrupt. The serial units are used in polling mode only. All interrupts are routed to addresses 0x8000 0000, 0x8000 0080, 0x8000 0100 and 0x8000 0180. It is recommended that all user applications are making usage of the BEV bit, so that own interrupt handling routines can be entered at the desired locations. The non-maskable interrupt (NMI) is treated as ordinary reset – no special handling implemented in here.

(2) Initialization

The caches are initialized prior to any download. A timer is not implemented. Only the peripheral units that are necessary for the operation of S-BOOT are supplied with a clock and are initialized.

(3) Memory Usage

The memory area used by S-BOOT is from ranging from 0xA3FF B000 to 0xA3FF FFFF.
The SDRAM boot flags are stored in ‘Top memory location’ –0x8 and 0x4. Once an application is loaded into RAM, one address is containing the text ‘EXEC’; the other one contains the execution address of the last recently loaded application.

S-BOOT allocates the reset vector location of the VR4181A, i.e. the Flash block located at address 0XBFC0 0000 cannot be used by any target applications. There is another monitor delivered with the Starter Kit: the Green Hills target monitor located at 0xBFC2 0000. It is running from Flash memory area, which prohibits a programming of the Flash, unless the Flash burning algorithm is located in SDRAM. In addition, S-BOOT is protected by H/W bit, so that no accidental erase may happen.

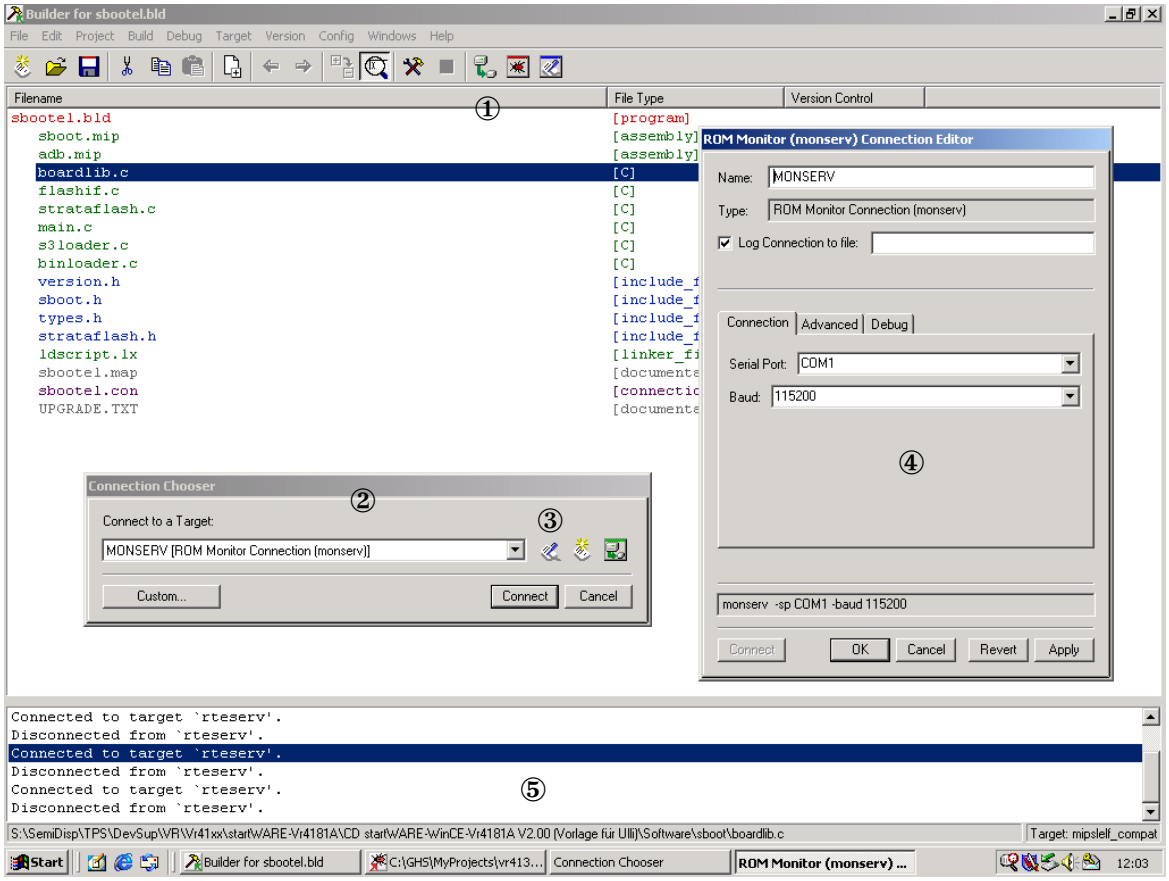
5.3 Green Hills Monitor Operation

The Green Hills monitor is used exclusively with the Green Hills Multi 2000 debug environment. Prior to using the Green Hills Monitor you must install Multi 2000 by following the instructions on the CD. Once a project (for example the sbootel project from the *start*WARE CD) has been opened with Multi 2000, you can establish a debug connection to your target system (i.e. *start*WARE-GHS-VR4181A hardware). This requires a monitor program running on the target hardware (i. e. the Green Hills Monitor) and a monitor server program running on the host PC side.

To start up the Green Hills Monitor on *start*WARE-GHS-VR4181A, just power up the board and wait, until the S-Boot countdown has expired. Then open the “Connection Choser” dialog in Multi 2000 by hitting the “Connect” icon (1) and select “monserv” as connection method (2). You may have to modify parameters for the connection with the “Edit” icon (3). Make sure that the connection parameters are set to 115 kbps in the connection editor (4). The screen shot in Figure 5-7 illustrates this process and the Green Hills manuals on the CD will give you all the details. Please refer especially to the “Target Connection Users Guide for MIPS” manual.

When the connection is successfully established, you will be informed in the lower portion of the Multi 2000 builder screen (5).

Figure 5-7: Establish target connection with Multi 2000



5.4 E-Boot Operation

The E-Boot monitor is installed only for operation with the Microsoft WincowsCE Platform Builder. Therefore the user is referred to Platform Builder documentation for details of E-Boot operation.

5.5 Using the WinCE Board Support Package

*start*WARE-WinCE-VR4181A comes with a board support package (BSP) for Microsoft's WindowsCE 4.1 operating system. This BSP has been prepared with and for Microsoft's WindowsCE Platform Builder 4.1, the integrated development environment for WindowsCE. Note that the Platform Builder itself is **not** included in the *start*WARE-WinCE-VR4181A deliverables. A trial version of Platform Builder is obtainable from Microsoft at the time when this manual is prepared.

5.5.1 Preparations for Using the WinCE BSP

Downloading and Debugging with WinCE requires an Ethernet connection between the host PC and *start*WARE-WinCE-VR4181A. This Ethernet connection can be provided either with an extra network adapter in the host PC (case A) or by re-using an already available, Ethernet-based network infrastructure (case B).

In case A the extra network adapter must be configured as follows:
no DHCP

- IP-address 192.168.xxx.yyy (where xxx/yyy are numbers between 1 and 255, preferably 1)
- SubnetMask 255.255.0.0
- no IP-forwarding

Note: The IP addresses used here are only examples.

The extra network adapter must be connected to the *start*WARE-WinCE-VR4181A board with an Ethernet cross-cable or via an Ethernet hub.

In case B *start*WARE-WinCE-VR4181A is connected to a free port of an Ethernet hub with a conventional Ethernet cable, so that host PC and *start*WARE can communicate over the network. In such an environment the IP-address for the *start*WARE board is normally automatically assigned via DHCP.

Microsoft Platform Builder must be installed on your host PC with the "MIPSII" processor selection included. Additionally a free serial port of your host PC must be connected to *start*WARE-VR4181A using the serial cable. Then *start*WARe-VR4181A can be powered up. Messages on the serial interface show, if the Ethernet connection has been successfully established.

Note: Make sure that E-Boot has been configured as default executable monitor program.

5.5.2 Remarks on Platform Builder

Building a binary WinCE image with Platform Builder is basically done in four process steps:

- SYSGEN
- BUILD
- BUILDREL
- MAKEIMG

In the SYSGEN phase the user selected components for a specific WinCE project are copied into a directory under the WinCE project directory (typical name:..`\wince410\pdx \m4181a_finish\wince410\eva4181a\cesysgen` with …`\m4181A_finish\`… being the project directory). Next the build process for the basic functions of the BSP is started. Details can be found in the Platform Builder on-line help under “platform headers”, “build process” or “sysgen phase”.

The SYSGEN phase is widely under user control by selecting components from the BSP function catalog and by setting build and environment options. These options will be described later.

In the BUILD process the platform specific components are built; as a rule of thumb these are the components found below the specific platform directory (…`\winCE410\platform\eva4181a`).

BUILDREL copies the relevant files generated in the SYSGEN and BUILD processes into the so-called “Release Directory”. Finally MAKEIMG glues these files together to a binary image file.

Note: We recommend to de-activate all on-line virus scanners while a binary image is created.

The four build process steps are controlled using the “Build” menu of the Platform Builder. In the following lines the related functions are described briefly; for details the user is referred to the Platform Builder on-line help function.

Table 5-1: Build menu and related build process steps

“Build” menu function	Comment
Generate Platform Headers	starts the SYSGEN process checks if parts of SYSGEN have already been executed and do not need to be repeated
Regenerate Platform Headers	starts the SYSGEN process all SYSGEN steps are unconditionally executed
Build Platform	starts BUILD, BUILDREL and MAKEIMG processes checks, if an additional SYSGEN is required and executes it eventually
Rebuild Platform	starts BUILD, BUILDREL and MAKEIMG without optimizations checks, if an additional SYSGEN is required and executes it eventually without optimizations
Makeimg	starts MAKEIMG without any check, if any of the proceeding steps is required as well very fast, but for experienced users only
Clean	deletes all files that have been processed with SYSGEN and all files in the release directory should normally only be executed if SYSGEN fails in a very early phase without an obvious reason

5.5.3 Integrating the BSP into Platform Builder

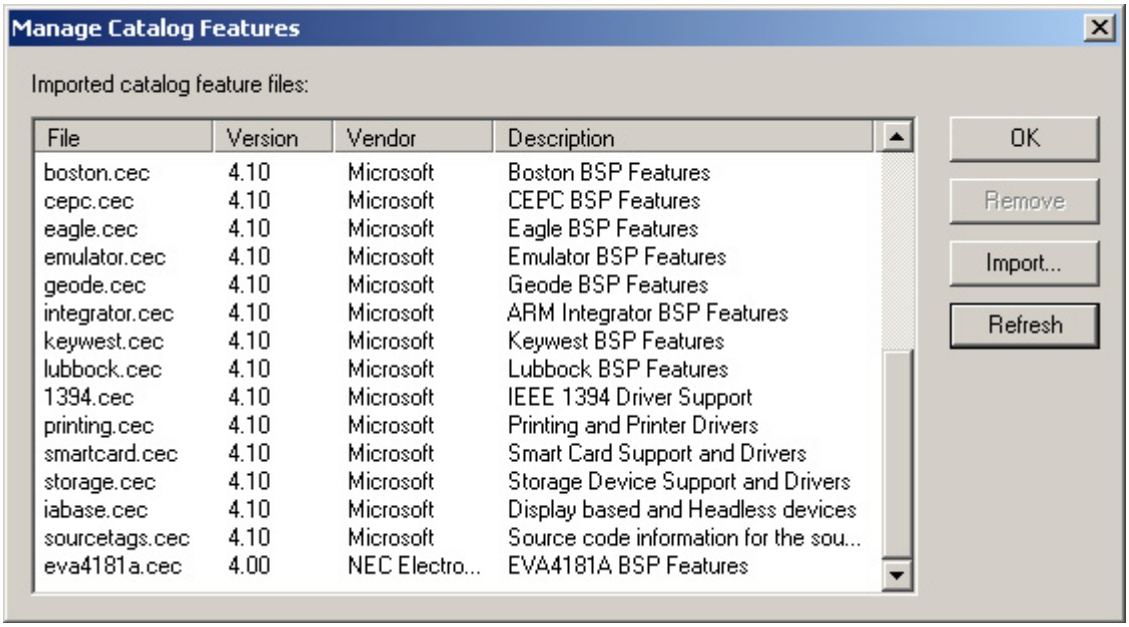
The BSP for *start*WARe-WinCE-VR4181A is delivered as a ZIP file on the enclosed CD (filename: `EVA4181A-CE41-V2.2.zip` or similar). This ZIP file must be unpacked with the “Recurse Folders” option checked. The WinCE410 directory (typically `C:\wince410`) must be selected as target directory.

Platform Builder requires, that any platform is “introduced” using a CEC file, which has information about the drivers that are implemented as part of the BSP. Importing the CEC file goes like this:

- Select “Manage Catalog Features” in the Platform Builder “File” menu
- Select “Import”
- Specify the path to the file eva4181a.cec (typically `c:\wince410\platform\eva4181a\cec-files`) and select this file
- Select “OK”.

Figure 5-8 illustrates these steps.

Figure 5-8: Importing the CEC file



5.5.4 Generating a Demo Image

After the BSP has been unpacked into the Platform Builder directory tree and after the CEC file has been imported, you can open the so-called “Workspace” in Platform Builder with “Open Workspace” in the File menu. The workspace for *start*WARE-WinCE-VR4181A is called `M4181A_FINISH.pbw` and can be found in the `c:\wince410\pdx\M4181A_FINISH` directory. In the workspace a multitude of build options is selectable.

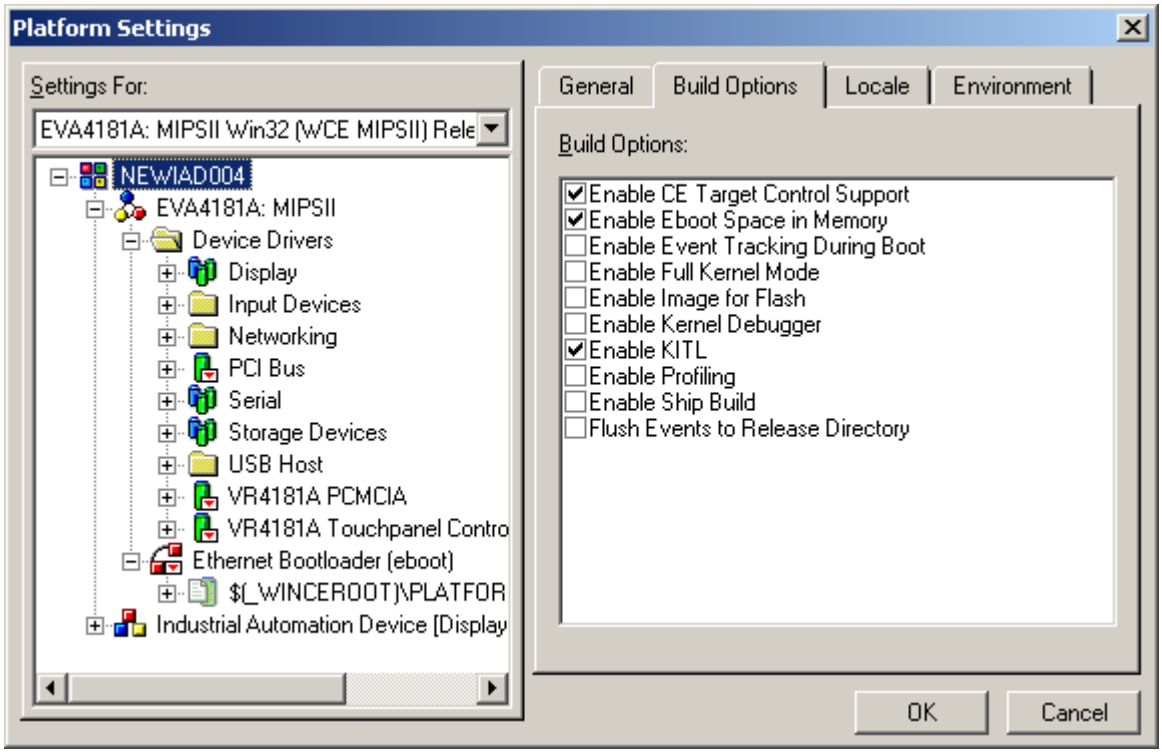
The primary decision to be made is between a “Debug” and a “Retail” image. This choice is made in the “Set Active Configuration” dialog of the “Build” menu. The following table summarizes the main differences between a Debug and a Retail image.

Table 5-2: Differences between debug and retail image

Debug image	Retail image
low compiler optimization, easy to debug, large image	high compiler optimization, tricky to debug, less large image
debug zones can be set	no debug zones possible
uses KITL	uses eventually KITL
supports CE-target control	supports eventually CE-target control
supports kernel debugging	supports eventually kernel debugging
no full kernel mode	mostly full kernel mode

Many of these configuration settings can be selected in the “Platform Settings” dialog shown in Figure 5-9, “Setting platform options,” on page 51. KITL, the **k**ernel **i**ndependent **t**ransport **l**ayer, is required for a debug connection to the respective target. If “Enable Full Kernel Mode” is checked, the image will be executed with reduced memory protection. Consequently exceptions due to memory protection violations will not be handled and the image will simply hang. On the other hand execution speed of the image will be higher.

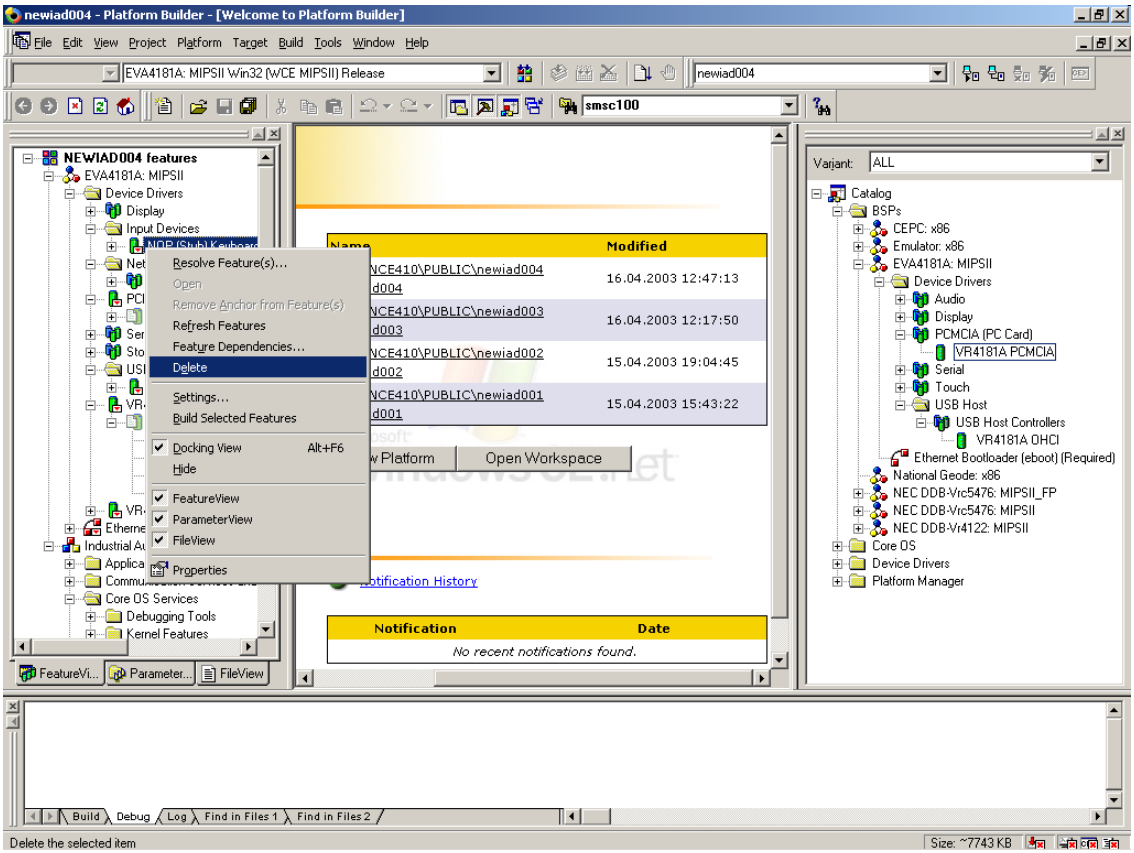
Figure 5-9: Setting platform options



The “Enable Image for Flash” option selects a different base address and prepares an image that can be stored in and executed from Flash memory. “Enable Profiling” supports program optimization using the Platform Builder’s profiling tool. Finally “Enable CE Target Control Support” links CEShell functions into the image.

Another important step in the creation of a WinCE image is the selection of the desired BSP components for a specific image. This is simply done with “drag and drop” of BSP components from the “Catalog View” into the “Feature View” of the platform. Removal of a component from a platform is done by marking it in the “Feature View” and subsequently hitting the “Delete” key.
Figure 5-10, “Adding/Removing BSP Components,” on page 52 illustrates these views.

Figure 5-10: Adding/Removing BSP Components

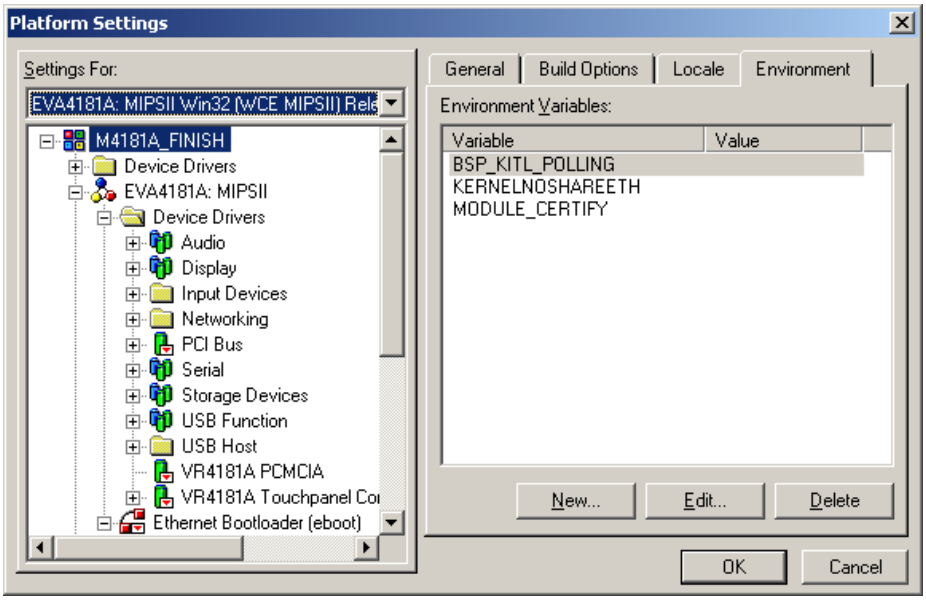


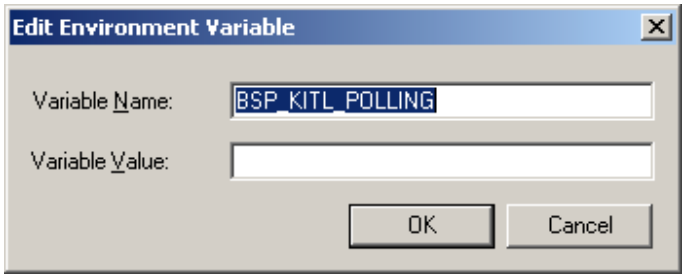
Finally more configuration possibilities are given by switch settings and defines. Common Platform settings are usually done under the Platform Builder. However, some switches necessary to operate specific modules or operational modes of Windows CE require a different handling from the common settings.

There are two ways to access those switches:

- The PB4 Platform settings->Environment tab menu is used (see Figure 5-11, “Editing Platform Switches,” on page 53), or
- A batch file located in the platform root directory is used.

Figure 5-11: Editing Platform Switches





The *start*WARE-WinCE-VR4181A configuration uses the following settings:

1. MODULE_CERTIFY (switch)
   Kernel checks certification of software components when loaded, if this switch is activated; default setting: off)
2. KITL_USE_INTERRUPT (switch)
   If this switch is set, network controller for kernel debugging is used in interrupt mode.
   If this switch is not set; network controller for kernel debugging is used in polling mode.
   Default setting: interrupt mode.

Note: It is recommended to have this switch set to interrupt mode.

Using the IDE to control the switch settings has the disadvantage that most of the switches are not visible in the environment variable tab. Even if the switches are defined in the CEC file, the basic description file for the platform BSP, the settings are NOT visible under PB40.

Shown below is the batch file of the EVA4181A (`c:\wince410\platform\eva4181a\eva4181a.bat`) to initialize the platform as currently implemented.

```
REM Disable the Cursor for touch panel access
set BSP_NOCURSOR=1

REM Use interrupts for KITL, if not set polling mode is used.
REM Recommended and default is to use interrupt mode.
set KITL_USE_INTERRUPT=1

REM Default is no audio support on EVA4181A
set BSP_NOAUDIO=1

REM Enable always the touch screen
set BSP_NOTOUCH=

REM No battery handling is implemented, skip this driver
set BSP_NOBATTERY=1

REM We don't use USB slave functionality, skip this driver
set BSP_NOUSBSER=1

REM Enable handling for VR4181A Rev1.2 BCU
set BCU_REV12=1

REM Describe handling for EVA4181A board using serial I/F DCD pin
REM Default is: no DCD connected
SET BSP_SERIAL_VR4181A_USE_DCD=
```

Any change in this batch file will have impact on the next build process.

There are 4 classes of switches available:

1. SYSGEN switches defining the modules used for system build process
2. BSP_xxx switches to explicitly include BSP features from build
3. BSP_NOxxx switches to explicitly exclude BSP features from build
4. IMG_xxx switches related to the image building process

For more details please refer to the Windows CE help files provided along with the platform builder.

Defines allow configuration of further options. Defines must be changed by editing the respective source file. A brief description of the most frequently used defines is given below:

1. SMC_DUMP_FRAMES (define in `c:\wince410\platform\eva4181a\kernel\hal\smc.c`)
   Frames sent or received by the kernel debugger are output via the serial debug interface; default setting: not defined.
2. FBBPP (define in `c:\wince410\platform\eva4181a\drivers\display\ddi_4181a\vr4181disp.h`)
   Sets the number of bits per pixel for the display driver; default setting: 16
3. SCREEN_REFRESH_30HZ / SCREEN_REFRESH_60HZ (define in `c:\wince410\platform\eva4181a\drivers\display\ddi_4181a\ddi_4181a.h`)
   Sets the refresh rate for the display; default setting: 60
4. DDI_4181A_Width (define in `c:\wince410\platform\eva4181a\drivers\display\ddi_4181a\ddi_4181a.h`)
   Sets the horizontal resolution for the display driver; default setting: 320
5. DDI_4181A_Height (define in `c:\wince410\platform\eva4181a\drivers\display\ddi_4181a\ddi_4181a.h`)
   Sets the vertical resolution for the display driver; default setting: 240

5.5.5 Starting WinCE on *start*WARE-WinCE-VR4181A

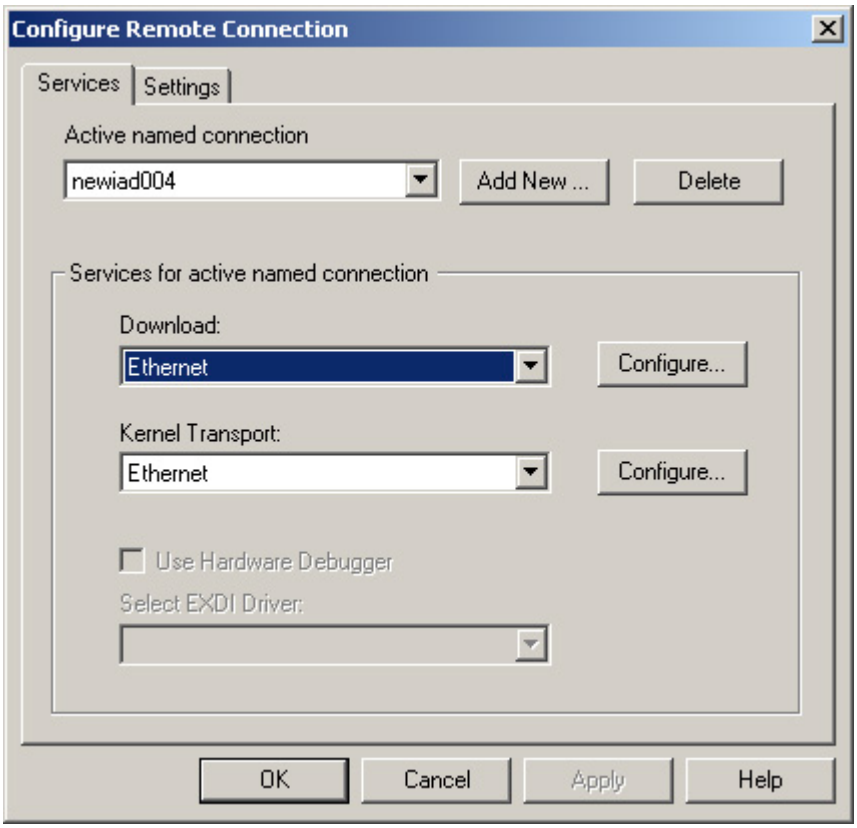
This chapter describes the required steps to download a WinCE image to the target system and to start it.

Before download of the WinCE image file `nk.bin`, the remote connection between host PC and target must be configured. Platform Builder provides the “Target Configure Remote Connection” dialog (see Figure 5-12, “Configuring a Remote Connection,” on page 56) for this purpose. If no active named connection has been prepared so far, this needs to be done now with the “Add New” button. Ethernet must be selected for both, download and kernel transport, in the “Services for active named connection” section of the dialog.

The “Configure” button allows you to test, if Platform Builder is able to find a target system. With “Available Devices” you should see an entry like “EVA4181A11” (see Figure 5-12, “Configuring a Remote Connection,” on page 56). In case that your Ethernet connection does not provide a DHCP service, this will require additional settings on the target system, that have to be done via a terminal emulation program. In case that EVA4181A11 is listed as available device, the “Configure Remote Connection” dialog can be closed and the download can be started with “Target Download/Initialize”.

Before the actual download takes place, additional settings must be done on the target side. Start a terminal emulation on the host PC (115200 bps, 8 bits, no parity, 1 stop bit, no flow control) and reset *start*WARE-WinCE-VR4181A. This will start execution of the S-Boot monitor program and your terminal window will show you a message as shown in Figure 5-2, “Startup Screen,” on page 40. S-Boot must be configured in such a way that the E-Boot monitor is the default executable file; details are given in Chapter 5.2.4 ”Flash Option menu” on page 42. Next, the target must be reset again; S-Boot and subsequently E-Boot monitors will be executed. E-Boot will generate a message as shown if Figure 5-13, “E-Boot start-up message,” on page 57 and prompt you for a static IP address.

Figure 5-12: Configuring a Remote Connection



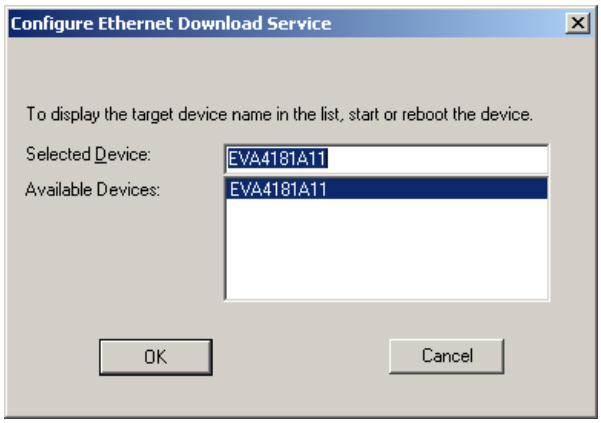
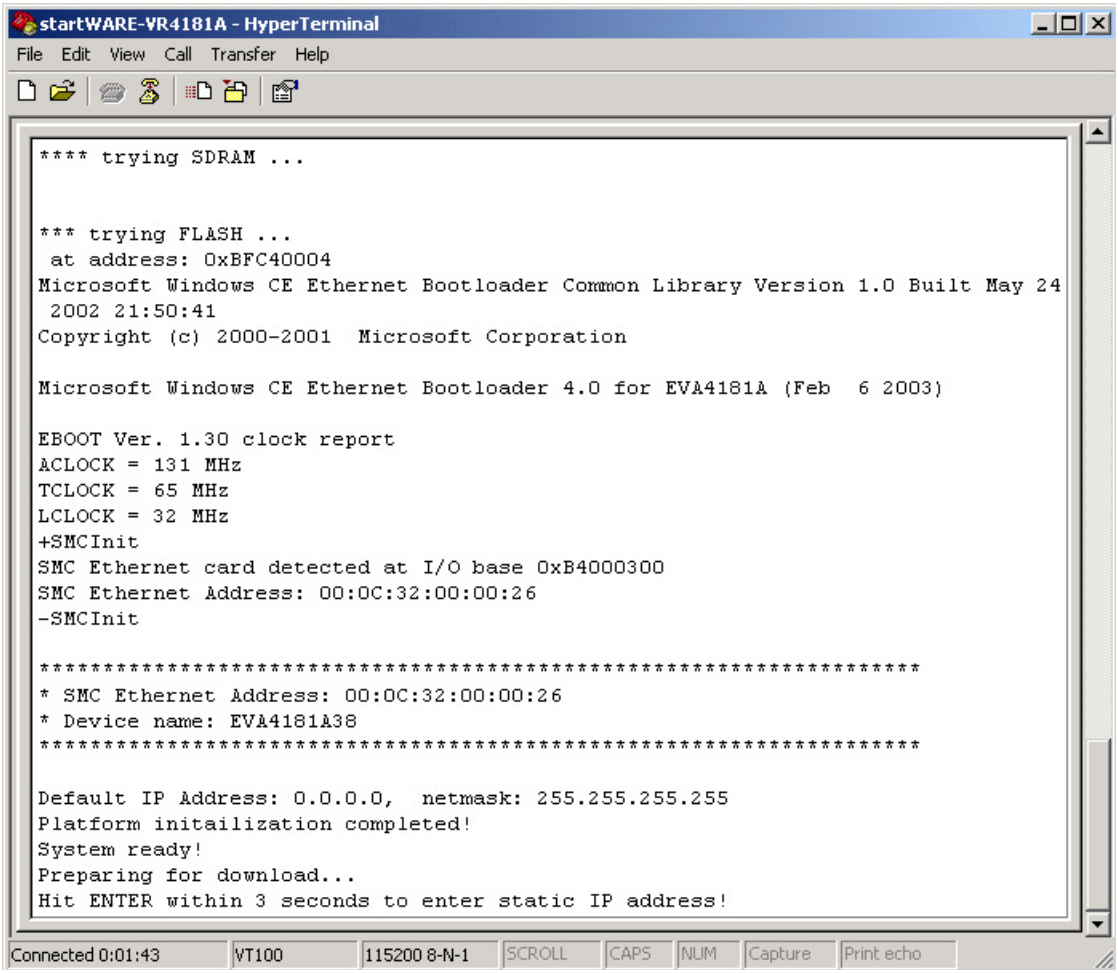


Figure 5-13: E-Boot start-up message



If your environment provides a DHCP service, no further action is required: E-Boot will wait for three seconds and then contact the DHCP server via broadcast. An IP address will be automatically assigned to the target and it should then occur in the “Configure Remote Connection” dialog of Platform Builder.

If no DHCP service is available, the IP address for the target must be entered manually. Hit “Enter” within three seconds after E-Boot begins to prompt for the IP address and then specify an IP address which is in the same subnet than the host PC and which is not used by any network adapter in the host. Normally the first six digits are identical to the IP address of the host, the rest is different. Next, E-Boot will prompt for a subnet mask; normally 255.255.0.0 can be used for this purpose.

When IP address and subnet mask have been correctly specified, the target will broadcast “BOOTME” to the network. The target must then be listed in the “Configure Remote Connection” dialog of the Platform Builder; this however is reflected by a message like

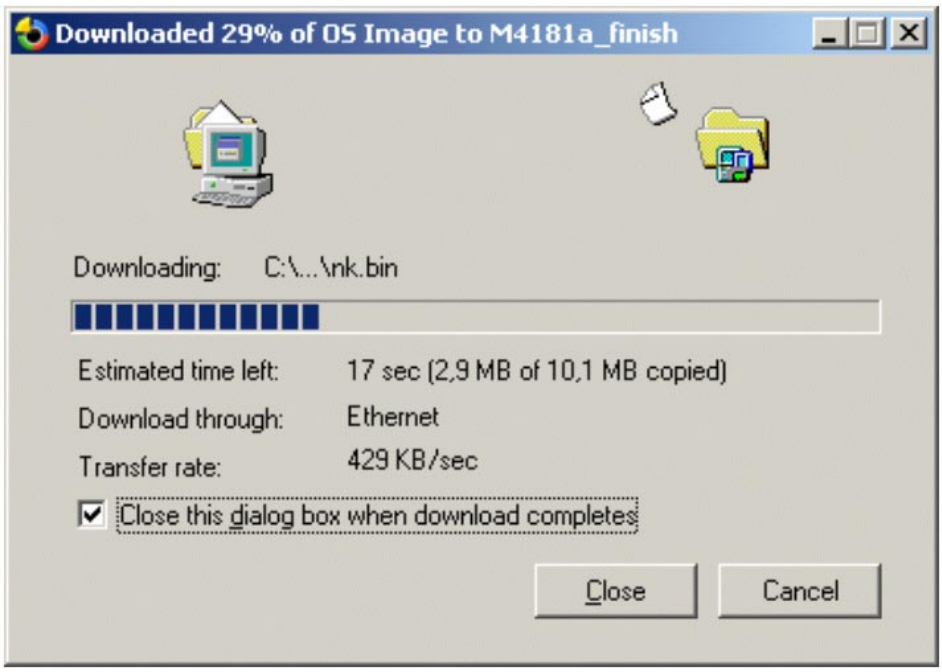
```
Locked Down Link 1
Src IP 192.168.1.2 Port 0400   Dest IP 192.168.2.202 Port 08B9
EthDown::TFTPD_OPEN::boot.bin
-EbootSendBootmeAndWaitForTftp
```

in the terminal window.

Note: The IP addresses used here are only examples.

During the actual download process, Platform Builder informs you about the download progress in the indicator box shown in Figure 5-14:

Figure 5-14: Download progress indicator



Simultaneously the download progress is indicated in the terminal emulation window. After download to SDRAM, the monitor will automatically jump to the entry address of the WinCE image. An example for a download is shown below. In case that a Flash image is downloaded, the S-Boot monitor will take over after the download and copy the downloaded image into Flash. Note that this copying takes a substantial amount of time.

```
Image will be transferred to Memory
 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74
75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98
99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116
117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134
135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152
153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170
171 172 173 174Found pTOC signature.
ROMHDR at Address A0090044h
RomHdr.ulRAMStart=81800000h RomHdr.physfirst=80090000h.
Download successful! Jumping to image at A0090004h...
Got EDBG_CMD_JUMPIMG
Got EDBG_CMD_CONFIG, flags:0x00000000
pCfgData == 0x8001020E
Flags are: 00000000Network Transfer Complete
pDriverGlobals == 0xA0002100
pDriverGlobals->eth.EbootMagicNum == 0x45424F54
pDriverGlobals->eth.etherEnabled == 0x0
pDriverGlobals->eth.etherFlags == 0x1
pDriverGlobals->eth.OdoAddr.dwIP == 0x201A8C0
pDriverGlobals->eth.OdoAddr.wMAC[0] == 0xC00
pDriverGlobals->eth.OdoAddr.wMAC[1] == 0x32
pDriverGlobals->eth.OdoAddr.wMAC[2] == 0xB00
pDriverGlobals->eth.SubnetMask == 0xFFFF
pDriverGlobals->eth.DownloadHostAddr.dwIP == 0xCA02A8C0
pDriverGlobals->eth.DbgHostAddr.dwIP == 0x0
pDriverGlobals->eth.KdbgHostAddr.dwIP == 0x0
pDriverGlobals->eth.PpshHostAddr.dwIP == 0x0

Jumping image at A0090004h...
.....................................................................
```

During start up of the WinCE image, target and host PC exchange information via Ethernet and via the serial line. The “Debug Window” within Platform Builder will show a string of messages like this (example):

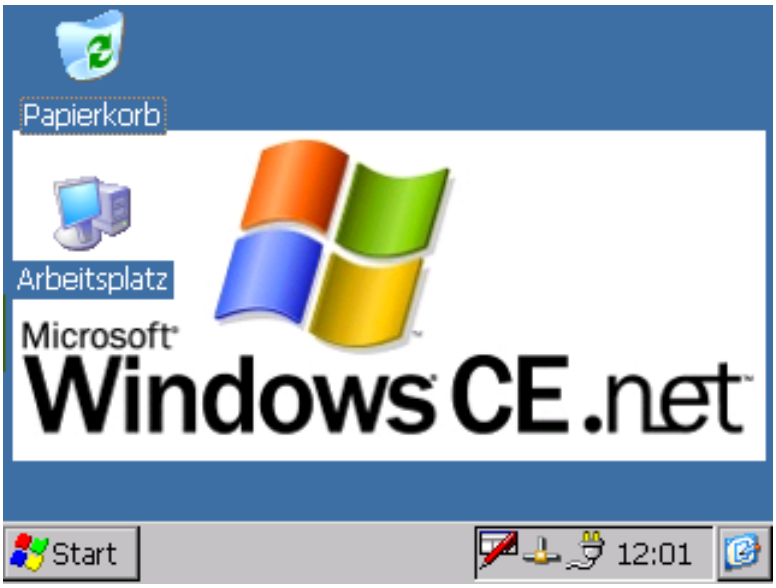
```
Kernel debugger is waiting to connect with target.
      0 PID:0 TID:0 Windows CE Firmware Init
      0 PID:0 TID:0
Start memclear
      0 PID:0 TID:0 Memory range from 81819000 to 82000000.
      0 PID:0 TID:0 End memclear
      0 PID:0 TID:0
ACLOCK = 131 MHz
      0 PID:0 TID:0 TCLOCK = 65 MHz
      0 PID:0 TID:0 LCLOCK = 21 MHz
      0 PID:0 TID:0 Firmware Init Done.
Kernel debugger connected (KDBG stream opened).
Kernel Version 1169 Checked loaded at 0x80090000, data relocated at 0x81800000
Debugger connection established (Target CPU is MIPS).
Loaded symbols for 'C:\WINCE410\PDX\M4181A_FINISH\RELDIR\EVA4181A_MIPSIIRELEASE\NK.EXE'
Loaded symbols for 'C:\WINCE410\PDX\M4181A_FINISH\RELDIR\EVA4181A_MIPSIIRELEASE\KD.DLL'
Finished re-loading kernel modules.
Loaded symbols for 'C:\WINCE410\PDX\M4181A_FINISH\RELDIR\EVA4181A_MIPSIIRELEASE\COREDLL.DLL'
Loaded symbols for 'C:\WINCE410\PDX\M4181A_FINISH\RELDIR\EVA4181A_MIPSIIRELEASE\FILESYS.EXE'
Loaded symbols for 'C:\WINCE410\PDX\M4181A_FINISH\RELDIR\EVA4181A_MIPSIIRELEASE\FSDMGR.DLL'
Loaded symbols for 'C:\WINCE410\PDX\M4181A_FINISH\RELDIR\EVA4181A_MIPSIIRELEASE\RELFSD.DLL'
Loaded symbols for
'C:\WINCE410\PDX\M4181A_FINISH\RELDIR\EVA4181A_MIPSIIRELEASE\TOOLHELP.DLL'
Loaded symbols for 'C:\WINCE410\PDX\M4181A_FINISH\RELDIR\EVA4181A_MIPSIIRELEASE\SHELL.EXE'


Welcome to the Windows CE Shell. Type? for help.

Loaded symbols for 'C:\WINCE410\PDX\M4181A_FINISH\RELDIR\EVA4181A_MIPSIIRELEASE\PM.DLL'
Loaded symbols for 'C:\WINCE410\PDX\M4181A_FINISH\RELDIR\EVA4181A_MIPSIIRELEASE\CESHELL.DLL'
Loaded symbols for
'C:\WINCE410\PDX\M4181A_FINISH\RELDIR\EVA4181A_MIPSIIRELEASE\EXPLORER.EXE'
Loaded symbols for
'C:\WINCE410\PDX\M4181A_FINISH\RELDIR\EVA4181A_MIPSIIRELEASE\SERVICES.EXE'
Loaded symbols for 'C:\WINCE410\PDX\M4181A_FINISH\RELDIR\EVA4181A_MIPSIIRELEASE\MSIM.DLL'
Loaded symbols for
'C:\WINCE410\PDX\M4181A_FINISH\RELDIR\EVA4181A_MIPSIIRELEASE\HTTPLITE.DLL'
Loaded symbols for 'C:\WINCE410\PDX\M4181A_FINISH\RELDIR\EVA4181A_MIPSIIRELEASE\UPNPSVC.DLL'
Loaded symbols for 'C:\WINCE410\PDX\M4181A_FINISH\RELDIR\EVA4181A_MIPSIIRELEASE\HTTPD.DLL'
Loaded symbols for 'C:\WINCE410\PDX\M4181A_FINISH\RELDIR\EVA4181A_MIPSIIRELEASE\WSPM.DLL'
Loaded symbols for
'C:\WINCE410\PDX\M4181A_FINISH\RELDIR\EVA4181A_MIPSIIRELEASE\OBEXSRVR.DLL'
Loaded symbols for 'C:\WINCE410\PDX\M4181A_FINISH\RELDIR\EVA4181A_MIPSIIRELEASE\MSMQD.DLL'
Loaded symbols for 'C:\WINCE410\PDX\M4181A_FINISH\RELDIR\EVA4181A_MIPSIIRELEASE\NETUI.DLL'
Loaded symbols for
'C:\WINCE410\PDX\M4181A_FINISH\RELDIR\EVA4181A_MIPSIIRELEASE\SIPSELECT.EXE’
```

At the other end the messages in the terminal emulation window during WinCE startup look like this (example):

```
Initializing KITL with interrupts
+OEMKitlInit
pDriverGlobals == 0xA0002100
pDriverGlobals->eth.EbootMagicNum == 0x45424F54
pDriverGlobals->eth.etherEnabled == 0x0
pDriverGlobals->eth.etherFlags == 0x1

pDriverGlobals->eth.EdbgFlags == 0x0
pDriverGlobals->eth.OdoAddr.dwIP == 0x201A8C0
pDriverGlobals->eth.OdoAddr.wMAC[0] == 0xC00
pDriverGlobals->eth.OdoAddr.wMAC[1] == 0x32
pDriverGlobals->eth.OdoAddr.wMAC[2] == 0xB00
pDriverGlobals->eth.SubnetMask == 0xFFFF
pDriverGlobals->eth.DownloadHostAddr.dwIP == 0xCA02A8C0
pDriverGlobals->eth.DbgHostAddr.dwIP == 0x0
pDriverGlobals->eth.KdbgHostAddr.dwIP == 0x0
pDriverGlobals->eth.PpshHostAddr.dwIP == 0x0
+SMCInit

+SMCInit - MII Control Register = 0x3000

SMC Ethernet card detected at I/O base 0xB4000300

SMC Ethernet Address: 00:0C:32:00:00:0B

SMC Reset complete2

-SMCInit

Device EVA4181A11, IP 192.168.1.2, Port 981
Calling EdbgInitDHCP

-OEMKitlInit
Host connected

Host IP: 192.168.2.202, port 2161

SetHostCfg()

KITLGlobalState == 0x61
KITL Initialized
Windows CE Kernel for MIPS Built on May 31 2002 at 14:35:12
KITL: Leaving polling mode...
-SMCEnableInts
```

Finally, when WinCE has completely booted up, you should see a screen similar to Figure 5-15 on the display of *start*WARE-WinCE-VR4181A:

Figure 5-15: WinCE desktop example



For download three different variants need to be differentiated:

i	Ethernet debugging (via KITL) + Retail	The functionality from CEShell (starting programs, setting breakpoints, process visualisation) is available.
ii	Ethernet debugging (via KITL) + Debug	The complete functionality from (i) is available; additionally you can set debug zones, you can single-step and show all variables.
iii	no KITL	The image is downloaded and executed from its base address. Subsequently, no control of debugging of the target is possible. Debug messages can still be output, but only in the terminal emulation window and not in the Platform Builder debug window.

5.5.6 Role of the Releasedirectory

The releasedirectory within platform builder's fairly complex file structure is one of the most important directories within Platform Builder: It contains all files which are combined to the final image in the MAKEIMG process; however not all files in the releasedirectory are integrated into the image. In case of the EVA4181A platform the releasedirectory is found in

```
c:\wince410\pdx\M4181A_FINISH\RelDir\EVA4181A_MIPSII\Debug
```

for the debug version and in

```
c:\wince410\pdx\M4181A_FINISH\RelDir\EVA4181A_MIPSII\Release
```

for the release version. If CEShell (CE target control support + KITL) is present, it is possible to utilize components (executables, DLLs) from the releasedirectory directly. When accessing a file, WinCE tries first to find it in Flash, then in RAM and last but not least in the releasedirectory. Thus it is possible to leave certain components outside the image and to have them downloaded if required. This is very useful during development of new programs and/or DLLs.

In this case you must make sure that none of these files is registered in any.bib file (`common.bib`, `project.bib` and `platform.bib`). For verification it is most simple to look at the ce.bib file in the releasedirectory. `Ce.bib` holds the summarised content of all.bib files after evaluation of all command line switches; furthermore `ce.bib` is newly generated with every MAKEIMG. If it is guaranteed that the file under test is not part of the image, this file can be started, stopped, modified and re-started arbitrarily without rebuilding the complete image.

`Reginit.ini` is another file in the releasedirectory; it holds all entries in the start-up registry of the image. `Reginit.ini` is also generated with every MAKEIMG.

5.5.7 WinCE BSP Components and Configuration

This chapter summarises the characteristics of the main BSP components. It starts with some remarks about the kernel and describes the peripheral drivers.

BSP Kernel:	The BSP kernel supports profiling; check the “Enable Profiling” box in the “Platform Settings” dialog in order to use it. Details about profiling kernel can be found in the on-line help function of the Platform Builder.
LCD Display:	The BSP has a display driver ready for the 320 x 240 pixel colour TFT that is mounted in the *start*WARE-WinCE-VR4181A housing. Documentation for the display can be found on the *start*WARE CD. Adaptations to other display types are possible, but they require changes to the driver.
Touch Panel:	The display has a resistive touch panel film mounted. *start*WARE-WinCE-VR4181A uses the touch panel as standard input device for the WindowsCE GUI.
CompactFlash:	Extensions can be connected using the CompactFlash slot. The CompactFlash driver in the WindowsCE BSP supports Flash memory extension cards. Other cards for networking or disk drivers may require additional drivers which are not part of the *start*WARE-WinCE-VR4181A BSP.
Serial Interface:	The serial interface 0 (SIU0, please see processor documentation for details) of the VR4181A can be used as normal communication interface within WinCE using this driver. It is accessible on SUB-D connector X4. The maximum speed is 115 kbps.
USB Host:	An OHCI 1.0 compliant USB driver is included in the BSP as well. It has been verified with a USB keyboard and a mouse. Mouse operation does not support a mouse pointer.
Audio (AC’97):	Audio playback of.WAV files via the AC’97 interface of the VR4181A is supported with this driver. Note that this requires to connect an external, AC’97 compliant codec device at the extension connectors of the *start*WARE-VR4181A PCB.

These drivers can be used for building other WinCE implementations on the *start*WARE-VR4181A hardware platform. They can also be used as a reference code for driver modifications that are required for adaptations to the user’s proprietary hardware. Descriptions of the related steps and procedures are found in the Platform Builder documentation and they are not explained in his manual.

5.6 Using the ELinOS Board Support Package

*start*WARE-Linux-VR4181A comes with a board support package for the Linux based ELinOS V2.2 operating system. Additionally a complete ELinOS tool chain consisting of cross compiler, debugger and a graphic configuration tool is delivered, so that the customer has “all he needs”. Before using the ELinOS tool chain, its installation on the Host PC is required; please consult the ELinOS documentation for installation instructions.

In the following chapter we will show how a simple project – the legendary “Hello World” project - can be generated using a classic Linux shell or alternatively the graphic configuration tool ELK. Next, we will explain how to run this project on the startWARE-VR4181A board.

5.6.1 “Hello World” Project Using the Linux Shell

Before working with the ELinOS environment you should be logged into your system with your user name; in the following we will use “you” as a place holder for your user name. The first step in the Linux project generation is the initial creation of a new project; the easiest way of doing this is cloning an already existing reference project with the “clone-project” function. Cloning allows, but does not require, changing the project name and important configuration parameters. The ELinOS installation already comprises a project named “Hello” which can be cloned to the new project “Hello.Clone” in the Linux shell like this:

```
[you@yourcomp elinos]$/opt/elinos/bin/elinos-cloneproject /opt/elinos/
demos/Hello Hello.Clone

CLONING PROJECT `Hello.Clone' FROM `/opt/elinos/demos/Hello'
============================================================

Checking existing project /opt/elinos/demos/Hello... ok
Checking new project Hello.Clone... ok
Cloning project /opt/elinos/demos/Hello as Hello.Clone... ok

CONFIGURING PROJECT
===================

Current Settings:

   ELINOS_BOARD      = custom
   ELINOS_CPU        = mips
   ELINOS_ARCH       = r3kle
   ELINOS_LIBC       = libc6
   ELINOS_DOSNAME    = hello
   ELINOS_BOOT_STRAT = floppy

Please select your board type.
(1)     startWARE-VR4181A
(2)     UMC-VR4181A
(3)     custom
Board Type [custom]: 1
```

Now enter “1” to select the *start*WARE-VR4181A board. Next you are prompted for the name of the new project.

```
Project Name (8 characters at most, no blanks) [hello]
```

We take over the project name as it is by hitting the return key.

```
You must now select the boot strategy for your new project:
(1)     zImage -- simple zImage
(2)     zImage_initrd -- zImage with embedded ram-disk
(3)     eboot -- zImage converted to Windows CE binary file
(4)     eboot_initrd -- zImage and embedded ram-disk converted to Windows
CE binary file
Boot Strategy: [eboot_initrd]: 4
```

Enter “4” (or hit the return key) in order to select eboot_initrd. As ELinOS allows the simultaneous installation of several Linux kernels, it is now required so select one of the appropriate kernels for the actual hardware platform. Immediately after installation of the ELinOS package, there is only one selection for the kernel and the choice is simple.

```
Please enter your kernel source tree. The suggested order is:
(1)     linux-mips-2.4.18
Kernel Source Tree [1]: 1
```

After entering “1” or hitting the return key, all configuration parameters are defined and the cloning process continues, while your host system outputs the following information.

```
Checking Kernel-Source /opt/elinos/linux/linux-mips-2.4.18... ok
Wiping old kernel... ok
Setting up new kernel... ok

Writing file./project.config... ok
Writing file ELINOS.sh... ok


Configuring Features in /home/wagenerw/src/elinos/Hello.Clone
============================================================

*** initializing features
*** feature input complete
*** initializing features
*** feature input complete
*** running feature configuration scripts...
(prepare) done.
(commit) done.
(mkefs) done.
(kernel) done.
(unkernel) done.
*** merging kernel configuration with feature-config...

Feature build complete.

-----------------------------------------------
Your new project has been set up successfully.
To work on it, type:

  sh# cd "/home/wagenerw/src/elinos/Hello.Clone"
  sh#. ELINOS.sh
-----------------------------------------------

CLONING DONE.
```

After successful cloning, the new project has been generated under “you@yourcomp/Hello.Clone”. The next step is now to source the newly generated script ELINOS.sh. When this script is asking for input after sh#, please enter “make boot” as shown below. Make boot” starts a lengthy process and the output messages are not printed below.

```
sh# cd Hello.Clone
sh#. ELINOS.sh

STARTING ELINOS SESSION
=======================

Setting up CDK mips_r3kle for libc6

$ELINOS_BOARD      = startWARE-Vr4181A
$ELINOS_BIN_PREFIX = mips_r3kle
$ELINOS_PROJECT    = /home/wagenerw/src/elinos/Hello.Clone
$ELINOS_DOSNAME    = hello
$CC                = mips_r3kle-gcc
$CXX               = mips_r3kle-g++
$AS                = mips_r3kle-as
$GDB               = mips_r3kle-gdb

#sh make boot
...
```

The project is now ready to download; download itself is described in Chapter 5.6.3 ”Downloading an ELinOS Project to startWARE-Linux-VR4181A” on page 70.

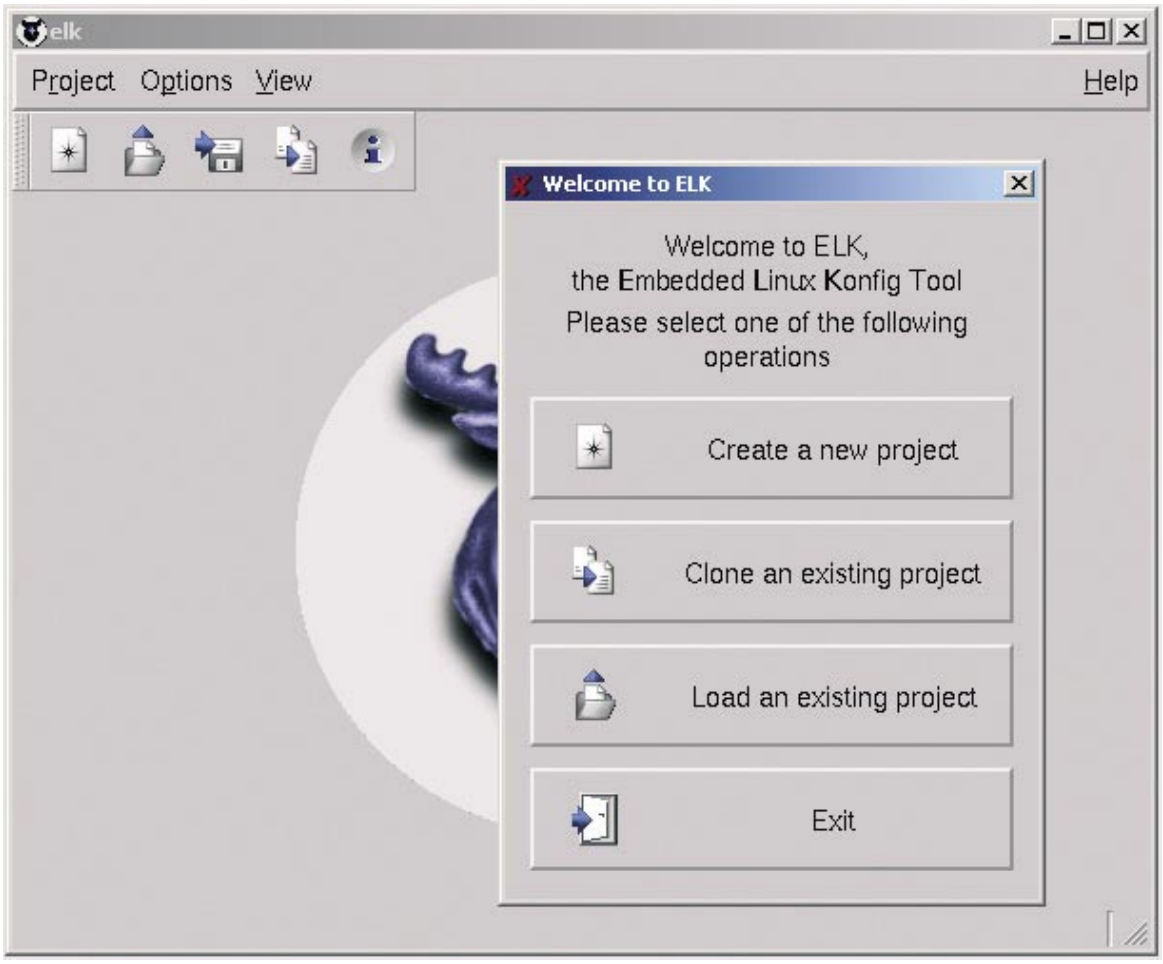
5.6.2 “Hello World” Project Using ELK

The ELinOS development environment contains a graphic configuration tool named ELK (**E**mbedded **L**inux **K**onfigurator). This chapter will basically describe how to repeat the steps from the previous chapter in a more graphic, windows-style fashion.
After ELK has been started with

```
sh# /opt/elinos/bin/elk
```

ELK will try to open an existing project in the current directory. If this is not possible, ELK will ask for initial project actions to be taken, as shown in Figure 5-16. We select “Clone an existing project” and choose “Hello” from the list of projects. Next, we are asked by the ELK configuration wizard for the various configuration options that have also been set in the Linux shell in Chapter 5.6.1 ””Hello World” Project Using the Linux Shell” on page 65.

Figure 5-16: ELK start-up screen



After these basic configuration settings have been made, ELK will come up with the project editor screen that allows you to perform settings with respect to

- Features included in the project
- Kernel configuration
- User application inclusion
- File system and
- Bootfiles

Selecting one of these configuration item categories in the project navigation bar will generate different lists of options in the lower portion of the ELK window. Figure 5-17 shows the screen for the kernel configuration step. After going through these configuration categories one-by-one – the buttons are arranged in the normal sequence of usage – you generate the downloadable system with the equivalent to the “makeboot” command by pushing the “Bootfiles” button. This will make another set of buttons accessible and with the “Create Files” buttons the system generation ELK is started.

The many steps of the system generation are recorded and documented in the ELK window. At the successful end of system generation you should see something similar to Figure 5-18, “ELK “Bootfiles” screen after system generation,” on page 70.

Figure 5-17: ELK project editor screen

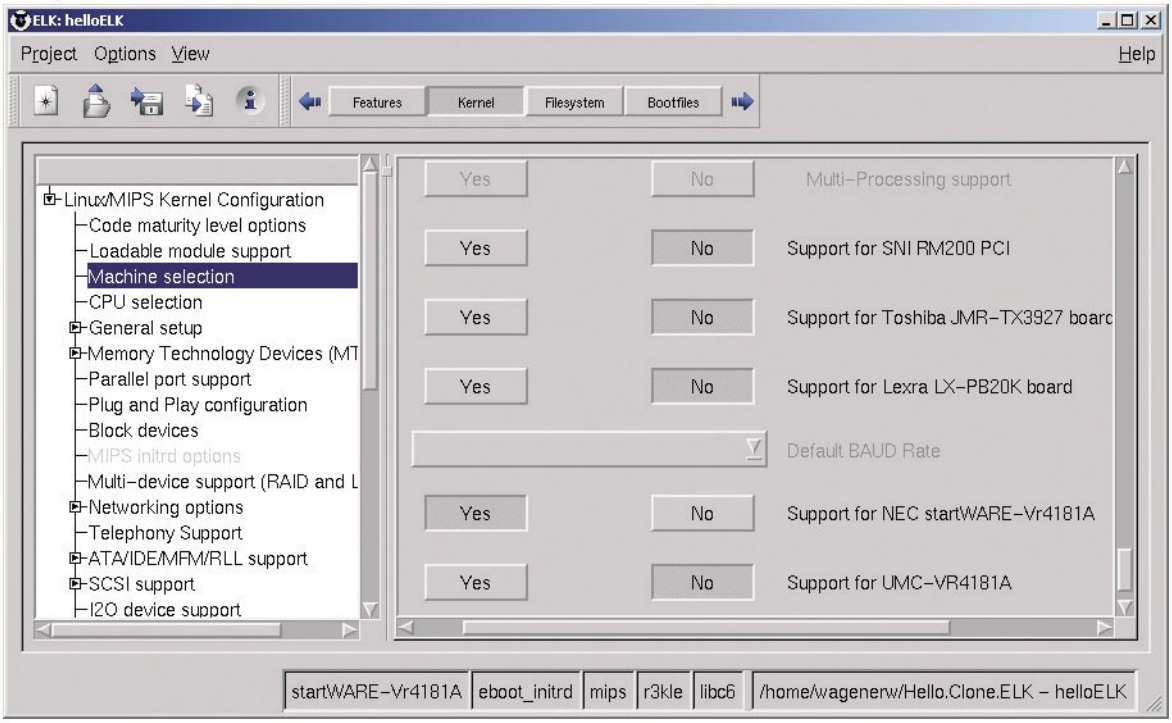
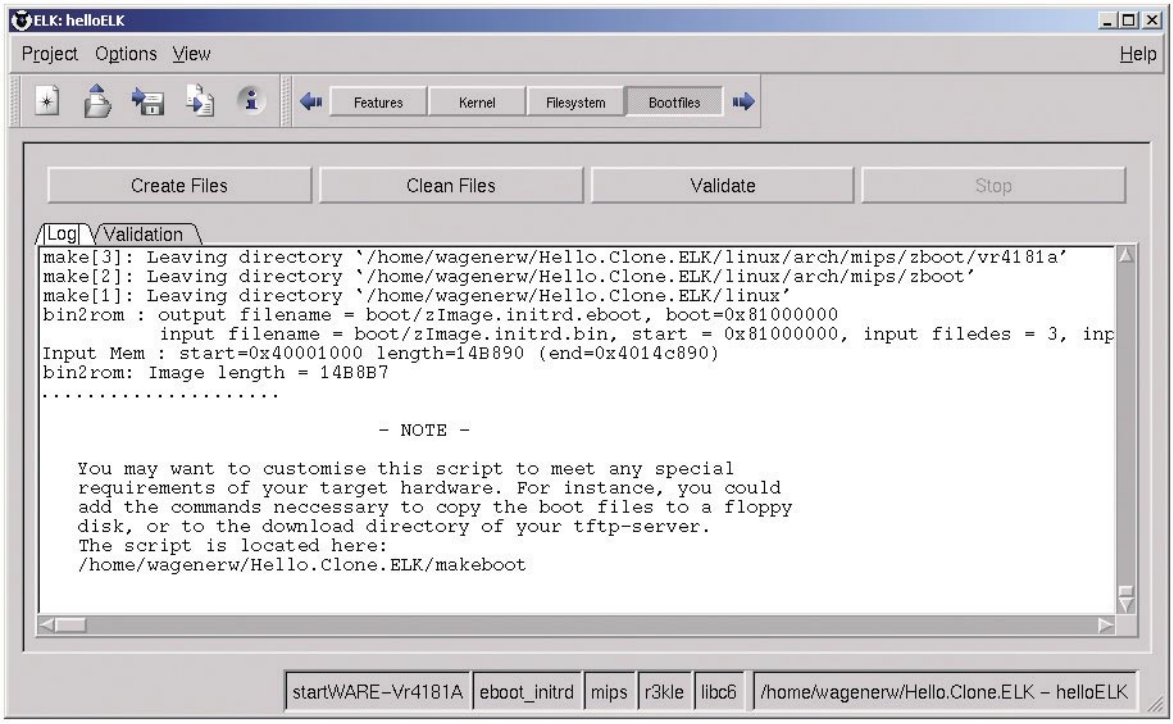


Figure 5-18: ELK "Bootfiles" screen after system generation



5.6.3 Downloading an ELinOS Project to *start*WARE-Linux-VR4181A

After generating the Linux operating system on command line or GUI level, the new binary file must be downloaded to the *start*WARE-Linux-VR4181A board. This is done using the E-boot monitor on the *start*WARE-VR4181A board and a utility called "ethload" that is part of the ELinOS distribution.

The startWARE-VR4181A board and your host computer must have a serial and (!) an Ethernet connection. Preferably your network environment should provide DHCP service and the Ethernet connection between target board and host PC should go via an Ethernet hub.

Before the actual download takes place, additional settings must be done on the target side. Start a terminal emulation on the host PC (115200 bps, 8 bits, no parity, 1 stop bit, no flow control) and reset *start*WARE-Linux-VR4181A. This will start execution of the S-Boot monitor program and your terminal window will show you a message as shown in Figure 5-2, "Startup Screen," on page 40. S-Boot must be configured in such a way that the E-Boot monitor is the default executable file; details are given in Chapter 5.2.4 "Flash Option menu" on page 42. Next, the target must be reset again; S-Boot and subsequently E-Boot monitors will be executed. E-Boot will generate a message as shown if Figure 5-13, "E-Boot start-up message," on page 57 and prompt you for a static IP address.

If your environment provides a DHCP service, no further action is required: E-Boot will wait for three seconds and then contact the DHCP server via broadcast. An IP address will be automatically assigned to the target and the *start*WARE-VR4181A board will begin to broadcast a "BOOTME" message into the network.

If no DHCP service is available, the IP address for the target must be entered manually. Hit "Enter" within three seconds after E-Boot begins to prompt for the IP address and then specify an IP address which is in the same subnet than the host PC and which is not used by any network adapter in the host. Normally the first six digits are identical to the IP address of the host, the rest is different. Next, E-Boot will prompt for a subnet mask; normally 255.255.0.0 can be used for this purpose.

When IP address and subnet mask have been correctly specified, the target will broadcast “BOOTME” to the network (as in the DHCP case). Now the ethload utility must be started in your ELinOS shell like this:

```
#sh# /opt/elinos/bin/ethload –f boot/zImage.initrd.eboot
```

Ethload will then try to find a boot device in the network; after finding a device the specified file will be downloaded. Ethload’s responses will look like this:

```
ethload v1.2 (c)1999, 2001 Ludovic LANGE
ethload comes with ABSOLUTELY NO WARRANTY; for details see COPYING.
 This is free software, and you are welcome to redistribute it
 under certain conditions; see COPYING for details.
Waiting for any device to boot:
 Got BOOTME from device IP: 172.29.29.135 type EVA4181A53 (EVA4): \
 eboot v4.0 arch:Mips R41xx/R5xxx
Sending file elinos/EVA4181A/umc3/boot/zImage.initrd.eboot
Sent 510739 bytes in 1.8 seconds
Sending jump packet
Acknowledge ok
```

After the download the Linux image is started on the *start*WARE board; in case of the “Hello World” project the following output or something similar will be visible in the terminal emulation window:

```
Sent BOOTME to 255.255.255.255
Locked Down Link 1
Src IP 172.29.29.135 Port 0800   Dest IP 172.29.29.180 Port 03D4
EthDown::TFTPD_OPEN::boot.bin
-EbootSendBootmeAndWaitForTftp

Image will be transferred to Memory
 0Found pTOC signature.
ROMHDR at Address 81000044h
RomHdr.ulRAMStart=80000000h RomHdr.physfirst=80800000h.
Download successful! Jumping to image at 81000000h...
Got EDBG_CMD_JUMPIMG
Got EDBG_CMD_CONFIG, flags:0x00000000
pCfgData == 0x8001020E
Flags are: 00000000Network Transfer Complete
pDriverGlobals == 0xA0002100
pDriverGlobals->eth.EbootMagicNum == 0x45424F54
pDriverGlobals->eth.etherEnabled == 0x0
pDriverGlobals->eth.etherFlags == 0x7
pDriverGlobals->eth.OdoAddr.dwIP == 0x871D1DAC
pDriverGlobals->eth.OdoAddr.wMAC[0] == 0xC00
pDriverGlobals->eth.OdoAddr.wMAC[1] == 0x32
pDriverGlobals->eth.OdoAddr.wMAC[2] == 0x3500
pDriverGlobals->eth.SubnetMask == 0xC0FFFFFF
pDriverGlobals->eth.DownloadHostAddr.dwIP == 0xB41D1DAC
pDriverGlobals->eth.DbgHostAddr.dwIP == 0x0
pDriverGlobals->eth.KdbgHostAddr.dwIP == 0x0
pDriverGlobals->eth.PpshHostAddr.dwIP == 0x0

Jumping image at 81000000h...
*****************************
loaded at:     81000000 81007000
relocated to:  80800000 80807000
zimage at:     81034538 8107CAEA
```

```
    relocated to:  8080B000 808535B2
    initrd at:     810079D8 81034538
    relocated to:  80854000 80880B60
    first four zImage words:
    08088B1F
    3F40FCCF
    6D760300
    756E696C
    Uncompressing Linux at load address 80100000
    Now booting the kernel
    Autodetected SDRAM: base 00000000, size 32MB
    CPU revision is: 00000c74
    Primary instruction cache 8kb, linesize 32 bytes.
    Primary data cache 8kb, linesize 32 bytes.
    ...
```

.

Note: The IP addresses used here are only examples.

Downloading an ELinOS image to the *start*WARE-VR4181A board with the E-Boot monitor does not generate any changes in the S-Boot flash directory. Therefore, if your downloaded image has been flashed, manual changes in the S-Boot flash directory are required: Restart your *start*WARE-VR4181A board with a serial terminal connection established, and press any key within five seconds after start-up, so that S-Boot is not left. Then enter “8” (to come to the flash status menu) and “7” (to create a new block entry). After entering the required parameters, S-Boot re-programs the flash memory accordingly and you can make this new entry the default executable entry by selecting function “3” in S-Boot’s flash status menu.

After the next reset, S-Boot will be started first, and after five seconds waiting time, the default executable block, i.e. Linux, will boot up.

5.6.4 Supported functions in ELinOS BSP

*start*WARE-Linux-VR4181A has a Linux operating system pre-installed that was generated based on the ELinOS board support package. This board support package can handle the following VR4181A peripherals:

- Serial interface
- 100 Mbps Ethernet interface (via off-chip Ethernet controller)
- USB host
- CompactFlash
- Real Time Clock
- 320 $\times$ 240 pixel TFT LCD display
- Touchscreen

The pre-installed image serves just as an example that illustrates the possibilities of ELinOS; other configurations and adaptations to user specific hardware and software requirements can be created using the ELinOS tool chain.

[MEMO]

Appendix A Circuit Diagrams

Figure A-1: VR4181A Power

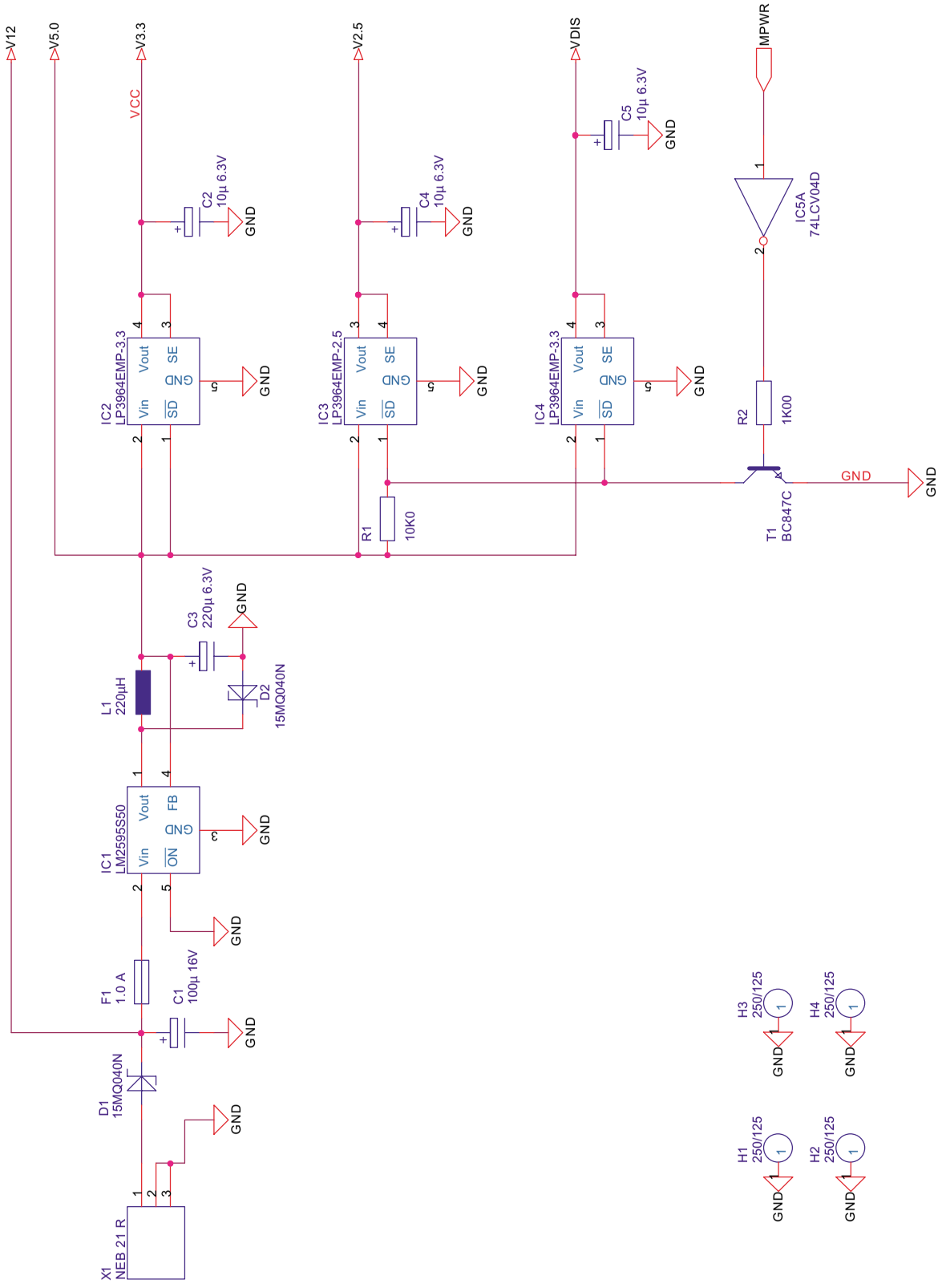


Figure A-2: VR4181A CPU

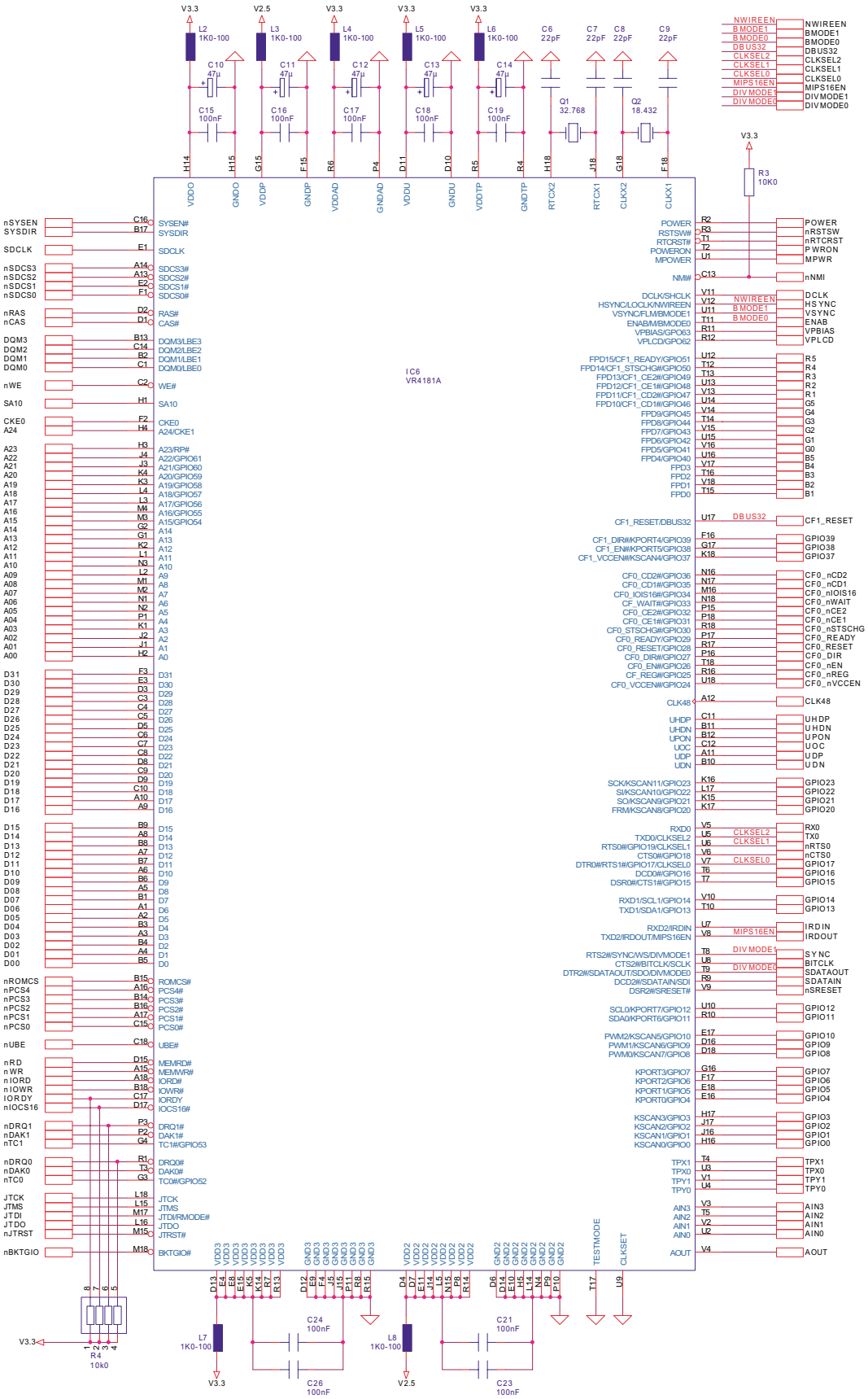


Figure A-3: VR4181A Flash

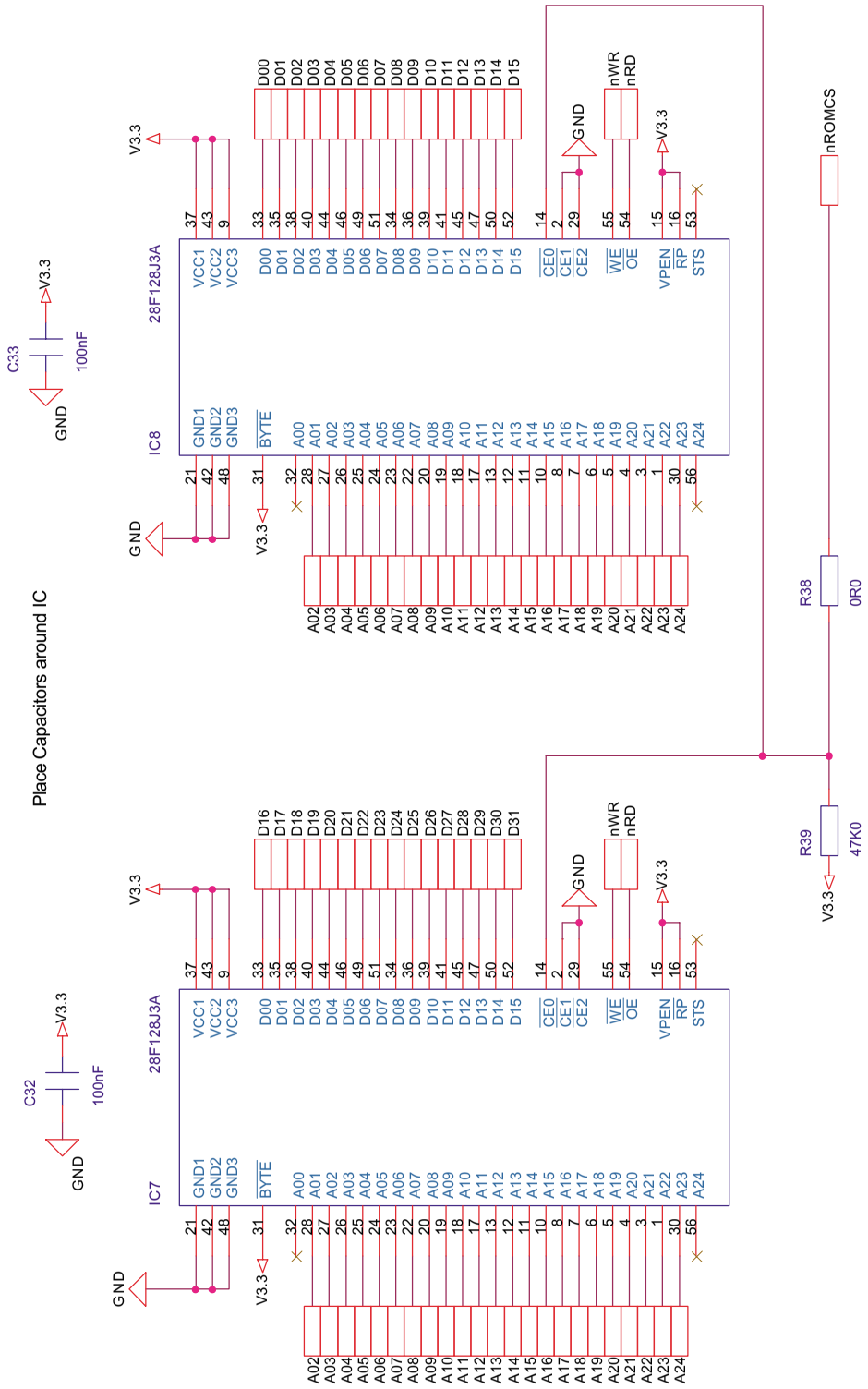


Figure A-4: VR4181A SDRAM

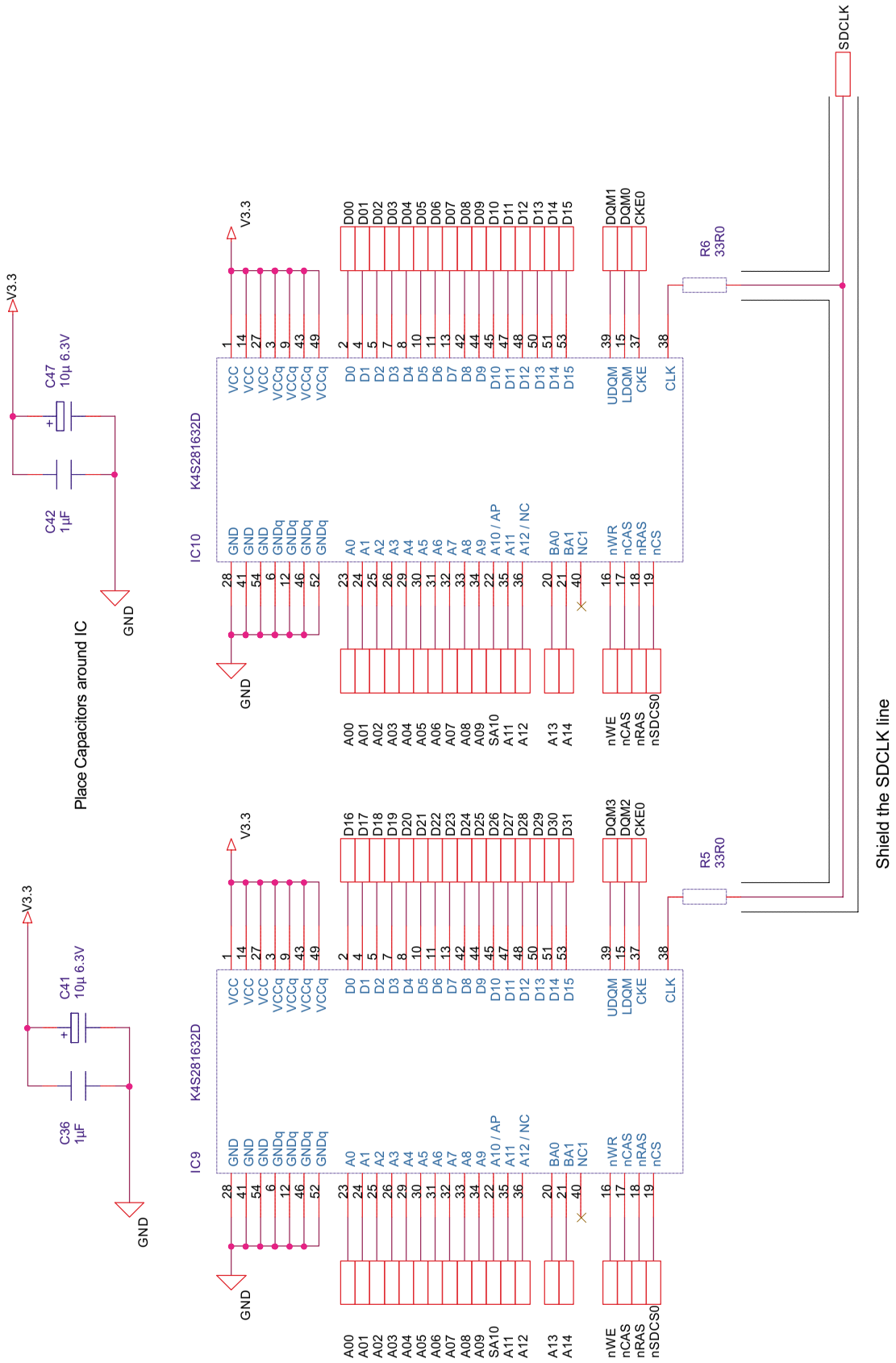


Figure A-5: N-Wire, Reset & Configuration

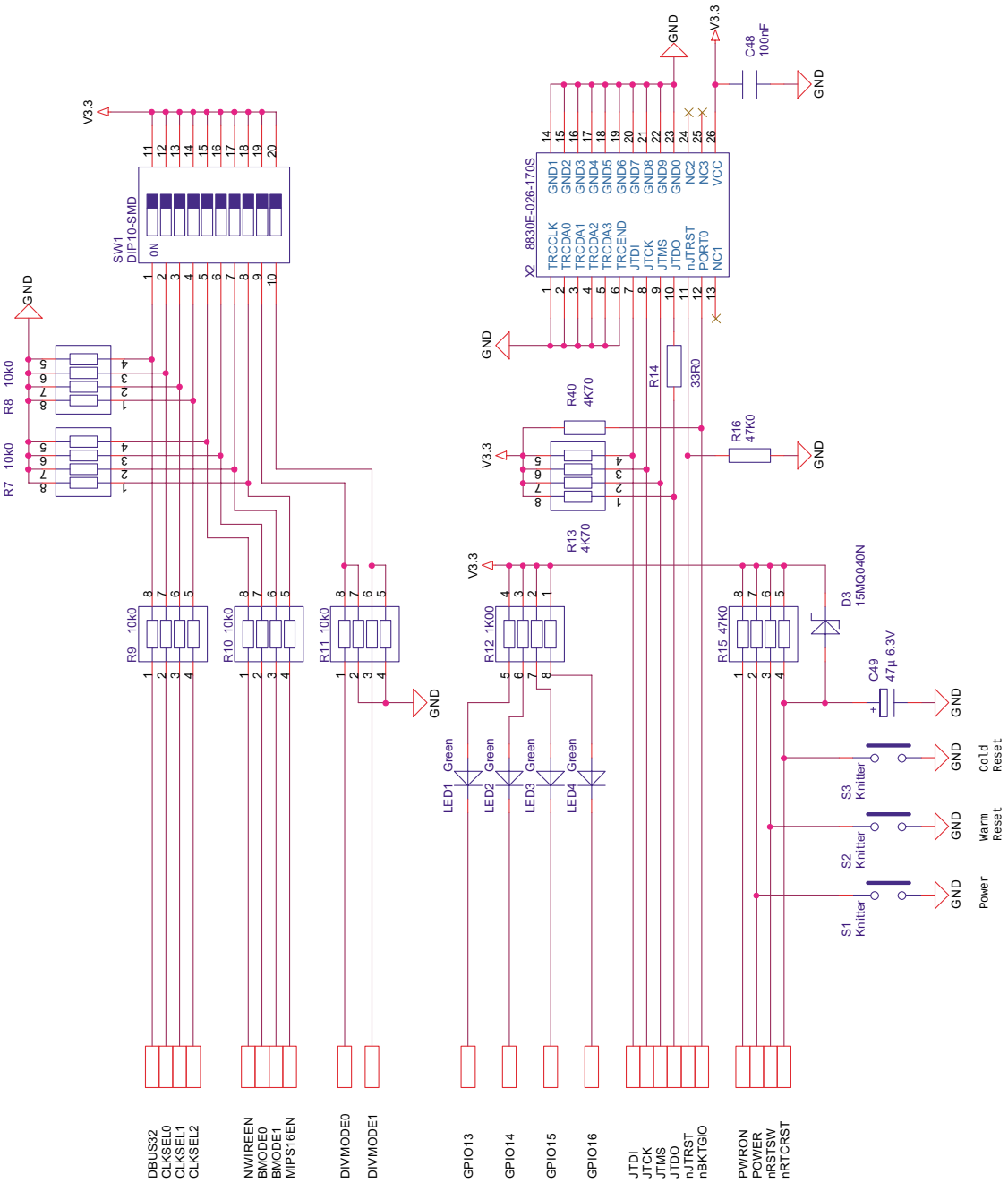


Figure A-6: VR4181A USB, RS232 & Analog

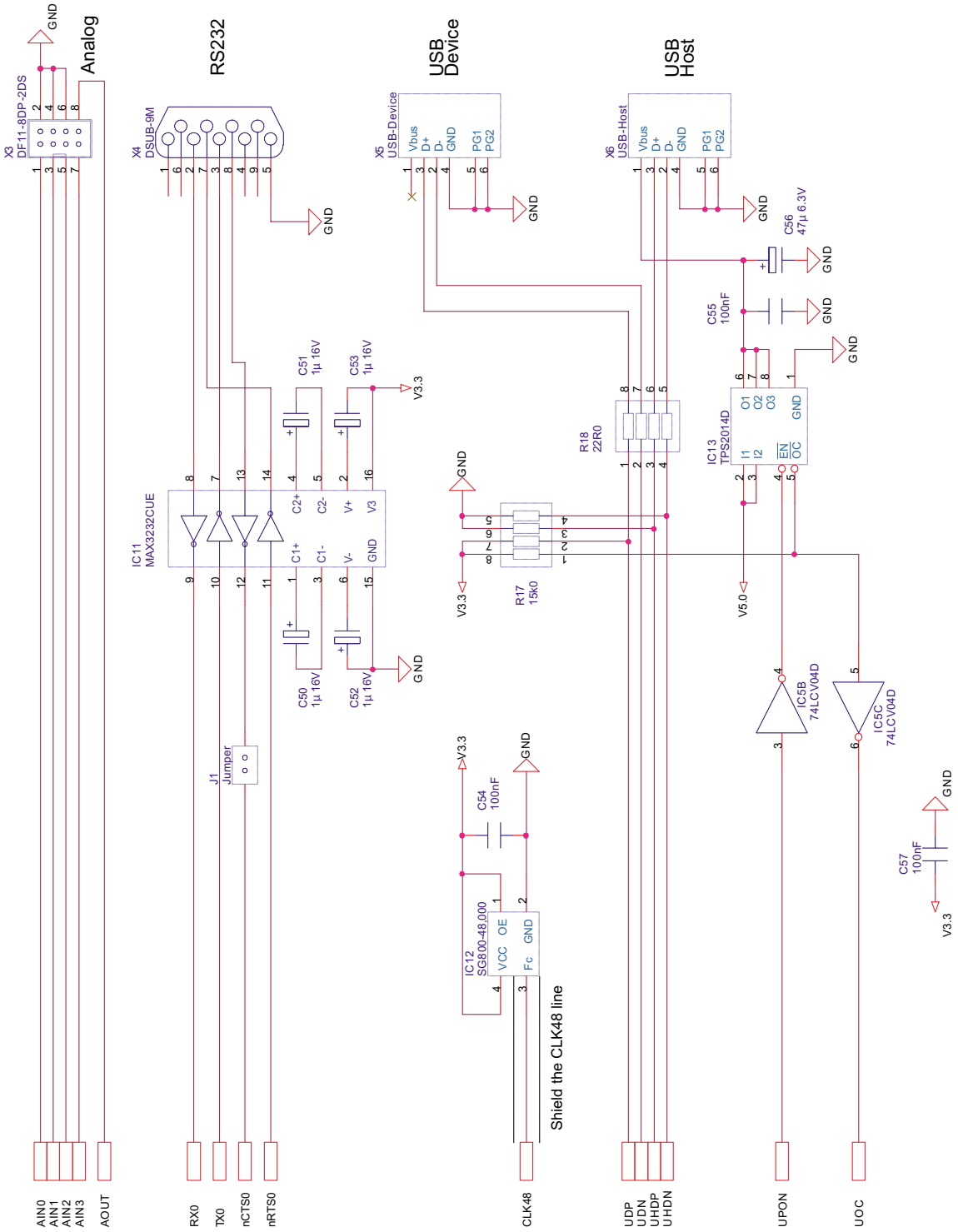


Figure A-7: VR4181A LCD & Touchscreen

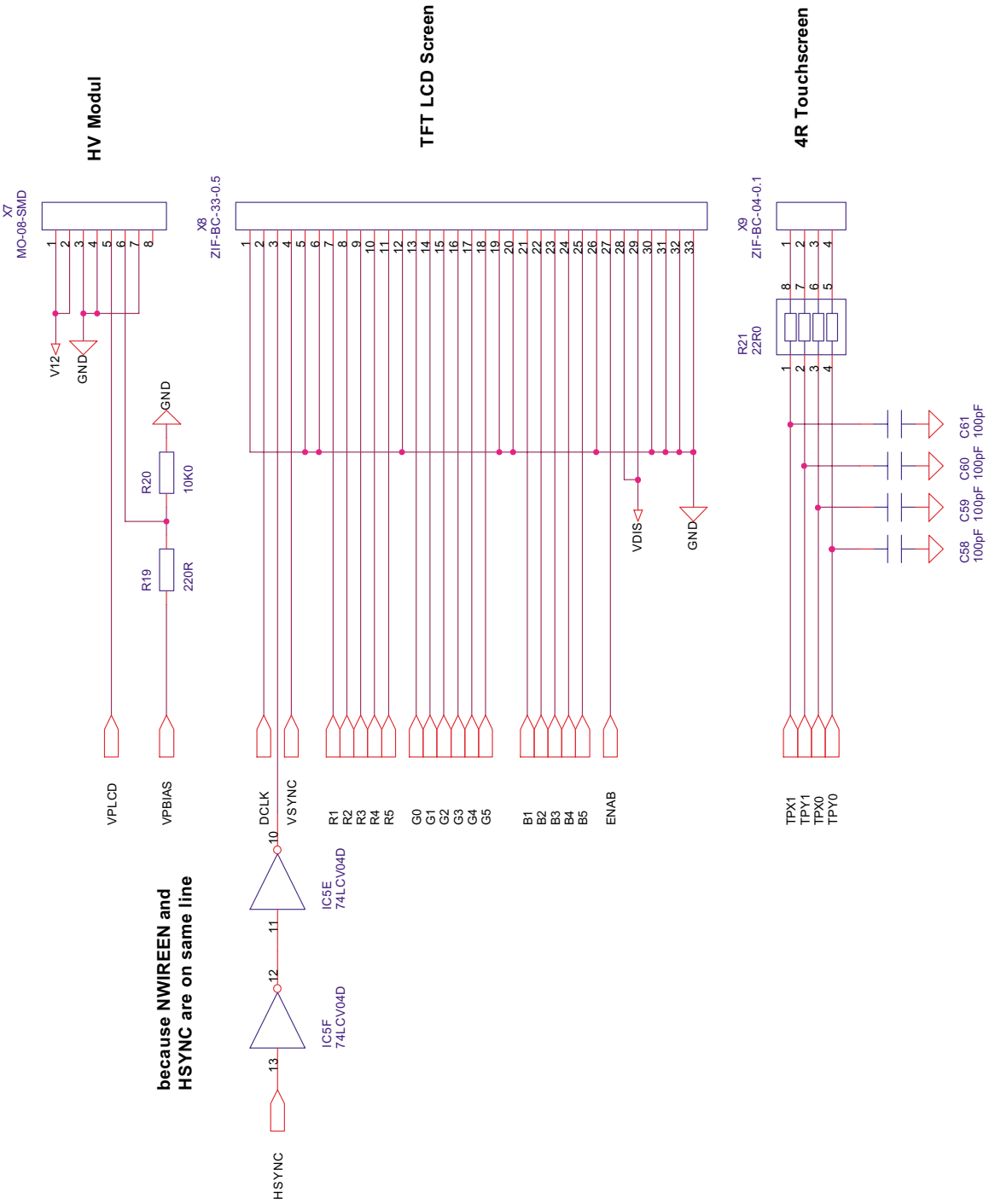


Figure A-8: Ethernet

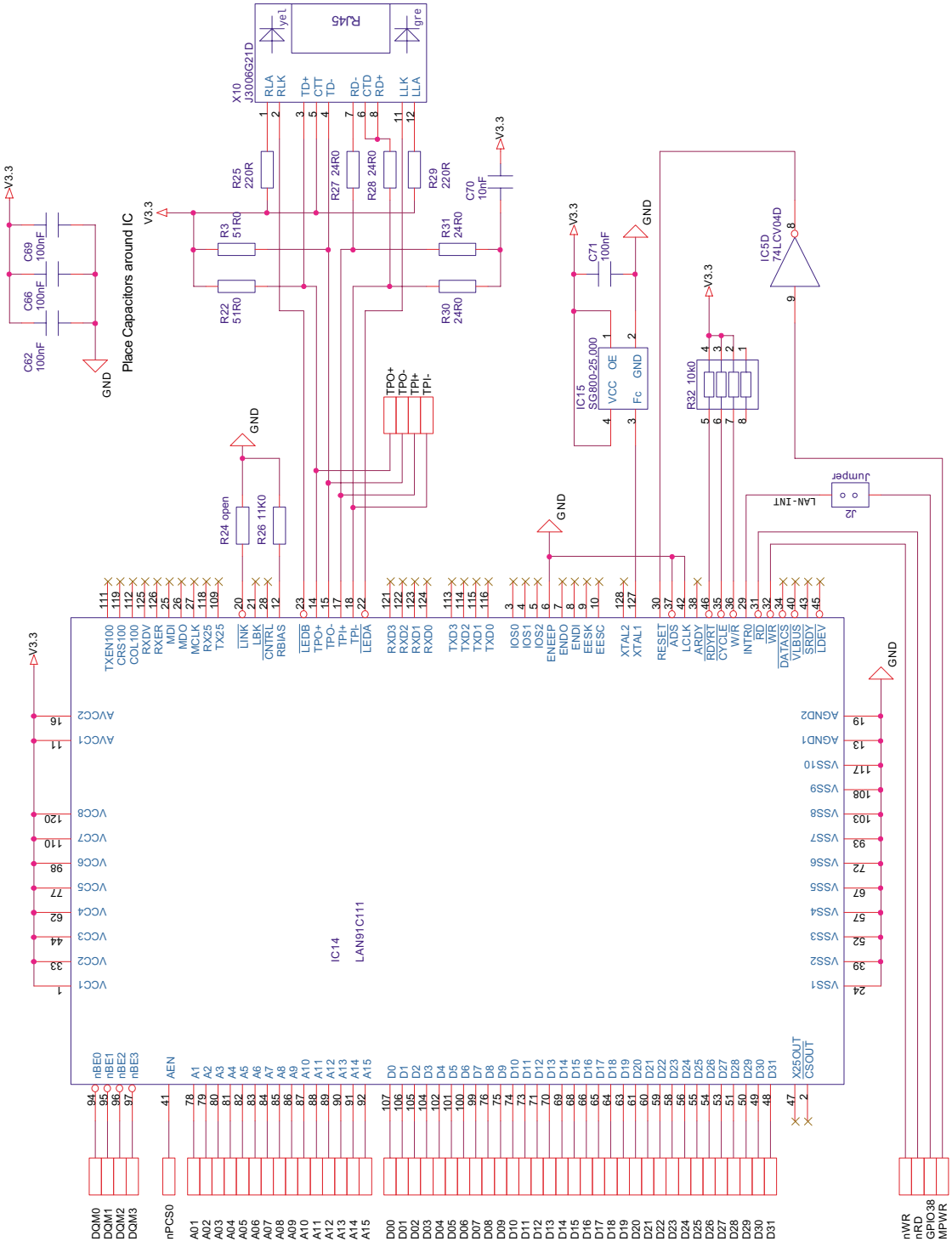


Figure A-9: Compact Flash

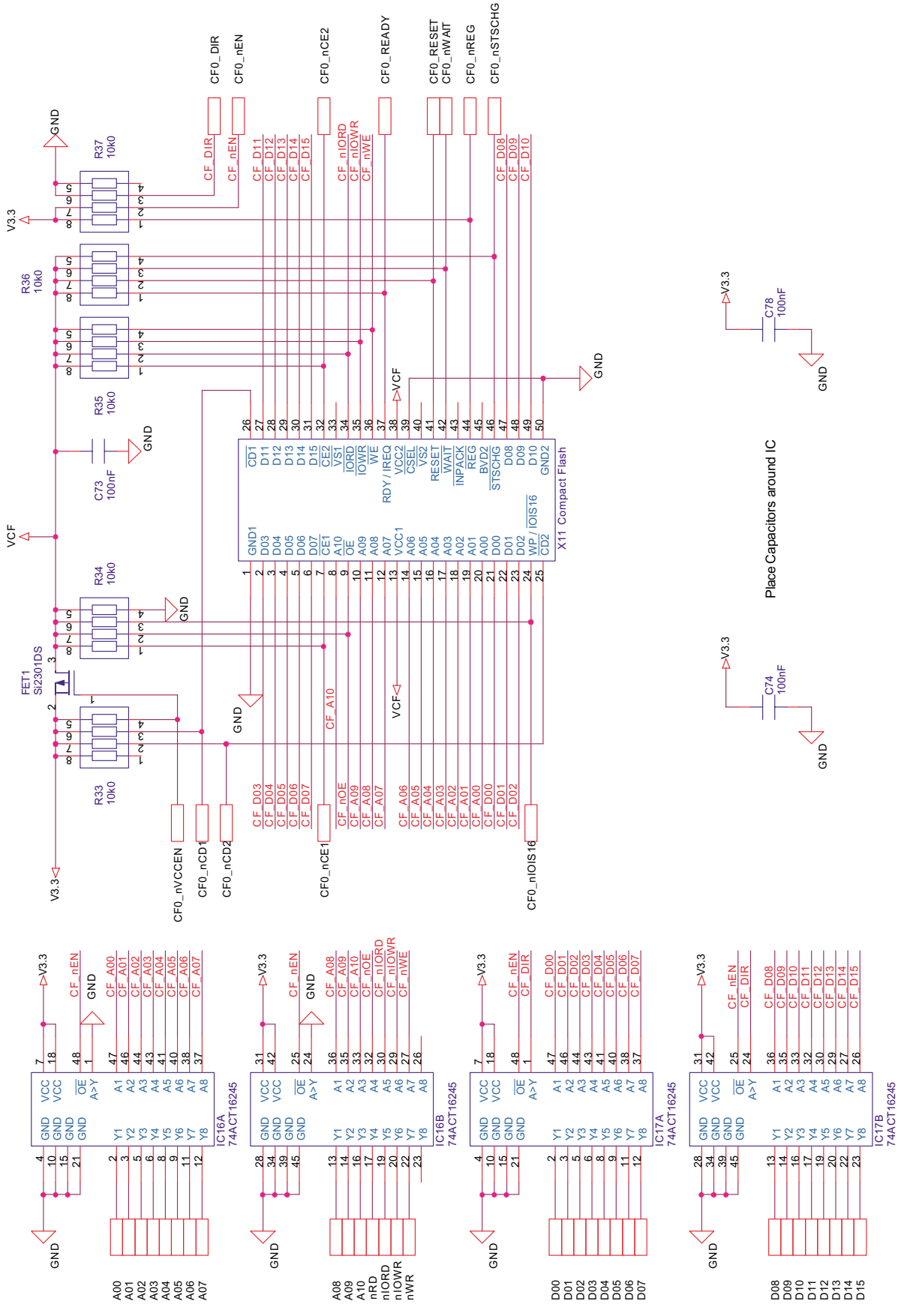
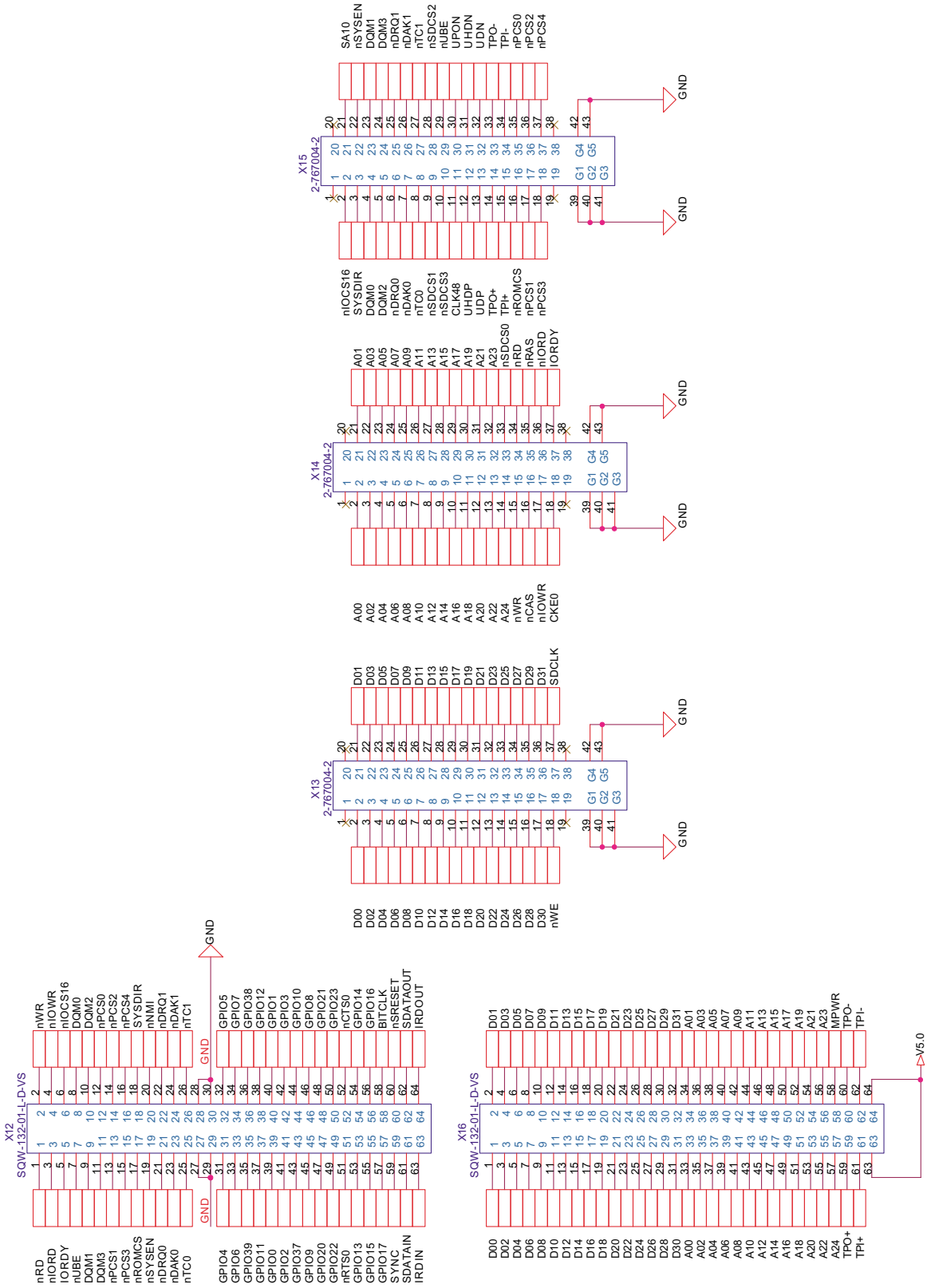


Figure A-10: VR4181A Connectors



NEC

Facsimile Message

From:

Name

Company

Tel. FAX

Address

Although NEC has taken all possible steps to ensure that the documentation supplied to our customers is complete, bug free and up-to-date, we readily accept that errors may occur. Despite all the care and precautions we've taken, you may encounter problems in the documentation. Please complete this form whenever you'd like to report errors or suggest improvements to us.

Thank you for your kind support.

North America
NEC Electronics Inc.
Corporate Communications Dept.
Fax: +1-800-729-9288
+1-408-588-6130

Europe
NEC Electronics (Europe) GmbH
Market Communication Dept.
Fax: +49-211-6503-274

South America
NEC do Brasil S.A.
Fax: +55-11-6462-6829

Hong Kong, Philippines, Oceania
NEC Electronics Hong Kong Ltd.
Fax: +852-2886-9022/9044

Korea
NEC Electronics Hong Kong Ltd.
Seoul Branch
Fax: +82-2-528-4411

Taiwan
NEC Electronics Taiwan Ltd.
Fax: +886-2-2719-5951

Asian Nations except Philippines
NEC Electronics Singapore Pte. Ltd.
Fax: +65-250-3583

Japan
NEC Semiconductor Technical Hotline
Fax: +81- 44-435-9608

I would like to report the following error/make the following suggestion:

Document title: _____

Document number: _____ Page number: _____

If possible, please fax the referenced page or drawing.

Document Rating	Excellent	Good	Acceptable	Poor
Clarity	☐	☐	☐	☐
Technical Accuracy	☐	☐	☐	☐
Organization	☐	☐	☐	☐