PEAK-gridARM Eval Board
Linux-based Development Platform for the gridARM™ Microcontroller

Hardware Manual
# Products taken into account

<table>
<thead>
<tr>
<th>Product Name</th>
<th>Model</th>
<th>Part number</th>
</tr>
</thead>
<tbody>
<tr>
<td>PEAK-gridARM Evaluation Board</td>
<td></td>
<td>IPEH-004051</td>
</tr>
</tbody>
</table>

CANopen® and CiA® are registered community trade marks of CAN in Automation e.V. gridARM™ is a trademark of Grid Connect, Inc. SD™ and microSD™ are trademarks or registered trademarks of SD-3C in the United States, other countries, or both. ARM® and ARM7TDMI® are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or in other countries.

All other product names mentioned in this document may be the trademarks or registered trademarks of their respective companies. They are not explicitly marked by “™” and “®”.

© 2014 PEAK-System Technik GmbH

PEAK-System Technik GmbH
Otto-Roehm-Strasse 69
64293 Darmstadt
Germany

Phone: +49 (0)6151 8173-20
Fax: +49 (0)6151 8173-29

www.peak-system.com
info@peak-system.com

Document version 1.1.1 (2014-03-03)
Contents

1 Introduction
1.1 Properties at a Glance 6
1.2 Prerequisites for Operation 7
1.3 Scope of Supply 7

2 Startup Procedure
2.1 Starting the Board 9
2.2 Setting Date and Time 13
2.3 Demo Scripts 14
   2.3.1 LEDs at the Digital Outputs (led_demo, led_switch) 14
   2.3.2 Push Buttons at the Digital Inputs (btn_demo) 15
   2.3.3 CAN Communication (can_demo) 15

3 Elements on the Board
3.1 Supply Socket 16
3.2 USB 17
3.3 Gigabit Ethernet 17
3.4 CAN 18
3.5 RS-232 Terminal 20
3.6 RS-232 Debug 21
3.7 JTAG 22
3.8 microSD™ Slot 23
3.9 Text Display Connector 24
3.10 I²C/SPI 26
3.11 Digital inputs and outputs 27
3.12 Analog Inputs 28
3.13 Status LED Microcontroller 29
3.14 Real-Time Clock 30
3.15 Reset Push Button 30
4 Configuration

4.1 Selecting the Boot Flash 31
4.2 Activating the Write Protection for the System EEPROM 32
4.3 Using I/O Interrupt or microSD™ Card Detection 33
4.4 Using Microcontroller LED or USB Detection 34
4.5 Assigning Traffic LEDs 35
4.6 Replacing the Button Cell for the Real-Time Clock 37

5 Technical Specifications 38

Appendix A Dimension Drawing 40
Appendix B Jumpers Overview 41
Appendix C Circuit Diagrams 45
C.1 Microcontroller 46
C.2 Memory 47
C.3 Ethernet 48
C.4 I/O 49
C.5 Serial 50
C.6 Power Supply 51
1 Introduction

Grid Connect\(^1\) has developed the ARM7\(^\text{TM}\) microcontroller gridARM\(^\text{TM}\) as SoC solution aiming for embedded applications in the field of industrial communication. The PEAK-gridARM evaluation board is a Linux-based development platform for the gridARM microcontroller. It has connectors for Gigabit Ethernet, High-speed CAN, USB 2.0, RS-232, SPI, and I\(^2\)C. Digital and analog inputs of the evaluation board are manipulated by push buttons and potentiometers. The microcontroller state as well as the states for supply and data traffic are indicated by LEDs. A Board Support Package (BSP) for Linux gives access to the hardware resources of the PEAK-gridARM evaluation board.

The gridARM\(^\text{TM}\) microcontroller by the U.S. company Grid Connect is distributed in Europe exclusively by PEAK-System.

This manual describes the evaluation board’s hardware and first steps with it. Please refer to the following documents for further information:


- gridARM - Hardware Data Sheet\(^2\): documentation about the microcontroller

\(^1\) http://gridconnect.com
\(^2\) http://gridconnect.com/media/documentation/grid_connect/gridARM_Hardware_DS.pdf
1.1 Properties at a Glance

- ARM7 gridARM microcontroller (80 MHz)
- Linux operating system (version 2.6.36)
- 4 MByte NOR flash with 16-bit data bus access
- 8 MByte NOR flash with 32-bit data bus access
- 64 MByte SDRAM with 32-bit data bus access
- I²C EEPROM for device configuration
- 4 MByte SPI flash
- Memory card slot for additional memory capacity
- Real-time clock (RTC) with battery
- Two alternative JTAG sockets
- Push button for hardware reset
- Voltage supply via USB or an external power supply unit (8 - 30 V)
- Dimensions: 110 x 110 mm
- Additional HD44780-compatible text display can be connected, availability on request

Communication

- Gigabit Ethernet (10/100/1000 Mbit/s)
- High-speed CAN channel (ISO 11898-2) with bit rates up to 1 Mbit/s
- Connection via D-Sub, 9-pin (in accordance with CiA® 102)
- NXP CAN transceiver PCA82C251
- USB 2.0 Full-speed (USB Device Port)
- Two RS-232 ports for debugging and terminal access
SPI bus (1 MHz) for two external extensions
I²C bus (400 kHz) for external extensions

Inputs and outputs
- 8 digital inputs, occupied with 6 push buttons, an RTC alarm, and memory card detection
- 8 digital outputs, occupied with 7 LEDs and a buzzer
- 5 analog inputs, occupied with 4 potentiometers
- Optional use of I/O ports with external wiring via jumpers
- LEDs for μC status, supply, CAN, and RS-232

1.2 Prerequisites for Operation
- PC with USB port for the power supply, alternatively, DC source 8 to 30 V with barrel connector

Optionally for development:
- RS-232 port at the PC and null-modem cable (D-Sub 9 f-f) for terminal access
- Linux operating system recommended for the PC

1.3 Scope of Supply
- PEAK-gridARM Evaluation Board
- USB connection cable (type B)
- Flat ribbon cable for RS-232 debugging port to D-Sub 9 m
Software and documentation

- μClinux-dist development environment and toolchain
- Board Support Package (BSP) for Linux with sources and binaries
- Hardware manual with circuit diagrams, BSP user guide (both PDF)
2 Startup Procedure

The chapter describes the startup of the PEAK-gridARM Evaluation Board by applying a supply voltage. The startup process can be observed on a PC via a serial connection.

Afterwards, the setup of the date and time is described.

**Note:** This startup procedure is referring to the Linux image being present in the memory at delivery of the evaluation board.

In addition to the scope of supply you need:
- PC with USB port for the voltage supply
- RS-232 port at the PC
- null-modem cable (D-Sub 9 f-f)
- terminal emulation program

2.1 Starting the Board

**Attention!** Electrostatic discharge (ESD) can damage or destroy components on the PEAK-gridARM Evaluation Board. Take precautions to avoid ESD when handling the board.

Do the following to start the evaluation board and to observe the startup process:

1. Place the board on a flat, clean, and dry surface.
2. Plug the supplied flat ribbon cable onto the double pin header “J501 Debug” on the board. The strand marked red
must be on the side of pins 1 and 2 of the double pin header.

RS-232 Debug (J501, left), USB (J401, right)

3. Use a null-modem cable (not included in the scope of supply) in order to connect the RS-232 port of the PC with the flat ribbon cable.

4. On your PC, start a terminal emulation program that establishes a connection to the evaluation board via the RS-232 interface (e.g. COM1 in Windows or ttyS0 in Linux). For this use the following connection parameters:
   - 38,400 bit/s
   - 8 data bits, no parity, 1 stop bit (8N1)
   - No hardware or software handshake

5. Use the supplied USB cable to make a connection between the powered-on PC and the “J401 USB” connector on the board.
   
   This establishes the voltage supply of the board.

Note: If you use another USB cable instead of the supplied one, it may happen that a cheap version doesn't have an appropriate wire cross section for proper voltage supply. Possibly, the evaluation board does not start then.
**Startup Sequence**

At delivery, the PEAK-gridARM Evaluation Board is supplied with the U-Boot bootloader and with a Linux image with Kernel 2.6.36, both being started in succession.

U-Boot bootloader with 3-second countdown until the automatic Linux boot sequence
Start of the Linux boot sequence

Bottom: Linux prompt after the finished boot sequence
After a successful boot sequence, the μC status LED on the eval-
atation board blinks with the rhythm of a heartbeat signal (blink -
blink - pause).

**Tip:** The pre-installed Linux image is provided with several
demo scripts. One starts automatically after the boot sequence
and activates the seven LEDs at the digital outputs one after
another. More information about the scripts in section 2.3 on
page 14.

At the Linux prompt (current path with number sign #), you can now
enter and execute commands.

### 2.2 Setting Date and Time

We recommend to check the system date and time and to correct it
if needed after the first start.

Check the date and time by entering the following command at the
Linux prompt:

```bash
date
```

Note the time zone in the output. The default is UTC (Coordinated
Universal Time). Accordingly, you must then specify the new time
for this time zone.

If required, enter a new time with the following two commands.
This sets the system time in a first step and then applies the set
time to the real-time clock.

```bash
date YYYY-MM-DD hh:mm
hwclock --systohc
```

3 **YYYY:** year, **MM:** month, **DD:** day, **hh:** hour, **mm:** minute
Alternatively, you can enter only the time with **hh:mm.**
Even after disconnecting the supply voltage, the real-time clock stays in function, because it is supplied by the button cell.

2.3 Demo Scripts

At delivery, the Linux image is provided with several demo scripts that illustrate the access possibilities to the periphery.

The scripts are located in the /home/peak/demo/ directory.

2.3.1 LEDs at the Digital Outputs (led_demo, led_switch)

During boot, the led_demo script is started automatically. It activates the seven LEDs at the digital outputs one after another.

Do the following to stop the script:

1. Enter the following command:
   
   ```
   ps
   ```

   All currently running processes are listed.

2. Note the PID (process number) of the /bin/sh demo/led_demo entry.

3. Enter the following command:

   ```
   kill -9 PID
   ```

   Replace PID with the previously noted number.

Using the led_switch script, you can influence each LED. It is called with parameters:

```
led_switch led [on|off]
```

led: cpu, yellow:1, yellow:2, yellow:3, yellow:4, green:5, green:6, red:7
Example (activating the yellow LED 3):

/home/peak/demo # ./led_switch yellow:3 on

2.3.2 Push Buttons at the Digital Inputs (btn_demo)

During boot, the btn_demo script is started automatically. On pressing button 1 or 2, it reacts with the transmission of a CAN message onto a connected CAN bus.

In the demo script, no function is assigned to the other four buttons.

You can stop the execution of the script according to the instructions in the previous subsection 2.3.1.

2.3.3 CAN Communication (can_demo)

During boot, the can_demo script is started automatically. It shows incoming CAN messages from a connected CAN bus in the debug terminal.

You can stop the execution of the script according to the instructions in the previous subsection 2.3.1.

To transmit single CAN messages manually, use the following command:

cansend can0 ID#Data

Example (CAN ID 12F, data bytes 45 67 89 AB):

/ # cansend can0 12F#456789AB

The Data Length Code (DLC, here: 4) is automatically determined from the number of given data bytes.
3 Elements on the Board

**Attention!** Electrostatic discharge (ESD) can damage or destroy components on the PEAK-gridARM Evaluation Board. Take precautions to avoid ESD when handling the board.

3.1 Supply Socket

The PEAK-gridARM Evaluation Board can either be supplied via the USB port (setting at delivery) or the supply socket. The supply path is determined by the jumper JP600.

<table>
<thead>
<tr>
<th>Jumper position JP600</th>
<th>Supply Eval Board via…</th>
</tr>
</thead>
<tbody>
<tr>
<td>JP600 2-1*</td>
<td>USB port</td>
</tr>
<tr>
<td>JP600 3-2</td>
<td>Supply socket</td>
</tr>
</tbody>
</table>

* Setting at delivery
Important note: Never connect all three pins of the JP600 jumper together. The microcontroller and the USB interface of a connected PC can otherwise be destroyed by overvoltage.

For the supply socket, a DC source with barrel connector is needed (not part of the scope of supply), e.g. an AC adaptor.

Supply voltage:
12 V DC (8 - 30 V possible)

Diameter of barrel connector:
a = 5.5 mm, b = 2.1 mm;
minimum length: 11 mm

3.2 USB

The USB interface supports the USB 2.0 standard (UDP) with bit rates up to 12 Mbit/s (Full Speed).

With a USB connection to a PC, the power supply of the evaluation board can be taken on (setting at delivery). Alternatively, it is possible to use the separate supply socket (see section 3.1 on page 16).

3.3 Gigabit Ethernet

The Ethernet interface can be operated with 10 Mbit/s, 100 Mbit/s, and 1 Gbit/s. On the connector housing, there are two LEDs that indicate an established connection and transfer activity.
Besides basic communication purposes, the Ethernet interface can also be used for a firmware update.

3.4 CAN

The CAN interface supports the two CAN formats 2.0A (Standard) and 2.0B (Extended). The NXP PCA82C251 transceiver leads out a High-speed CAN bus (ISO 11898-2), that allows bit rates up to 1 Mbit/s.
The bus is connected to the 9-pin **D-Sub** port. The pin assignment for CAN corresponds to the specification CiA® 102.

If the evaluation board is connected to the end of a CAN bus and this end isn’t yet terminated, the **CAN termination** (120 Ω) available on the board can be activated. This is done by setting both switches of the switch block J500 to ON.
3.5 RS-232 Terminal

Via RS-232 (V.24 levels), the D-Sub port J502 provides a terminal access for administrative purposes. With a D-Sub extension cable (9-pin m-f, 1:1), a connection to a PC can be established. Alternatively, the RS-232 signals can be accessed via the pin header field J503 (not equipped).

RS-232 port J502 (D-Sub 9-pin), J503 (pin header field not equipped) alternatively

Assignment of the D-Sub connector for RS-232 Terminal
### 3.6 RS-232 Debug

Via RS-232 (V.24 levels), the double pin header J501 provides debugging access. At delivery, the evaluation board is configured to route output of the bootloader and of the Linux environment and to receive input via this interface (see also 2.1 *Starting the Board* on page 9). For further information about debugging, see the Hardware Data Sheet of the gridARM microcontroller.

For a connection, the provided flat cable can be used. The connection to the PC is then done with a null modem cable (D-Sub 9 f-f).

Two LEDs can be used for the traffic indication for RxD and TxD. For the corresponding configuration, see 4.5 *Assigning Traffic LEDs* on page 35.
RS-232 port for debugging purposes.

<table>
<thead>
<tr>
<th>Function</th>
<th>Pin at J501</th>
</tr>
</thead>
<tbody>
<tr>
<td>RxD</td>
<td>3</td>
</tr>
<tr>
<td>TxD</td>
<td>5</td>
</tr>
<tr>
<td>GND</td>
<td>9</td>
</tr>
</tbody>
</table>

3.7 JTAG

For the development at the microcontroller, a JTAG interface with two connection possibilities is available. J100 is a 10-pin double pin header, J101 a 10-pin header socket with 1.27-mm grid.
JTAG ports J100 (double pin header) and J101 (header socket)

<table>
<thead>
<tr>
<th>Function</th>
<th>Pin at J100</th>
<th>Pin at J101</th>
</tr>
</thead>
<tbody>
<tr>
<td>GND</td>
<td>1, 2</td>
<td>3, 5, 9</td>
</tr>
<tr>
<td>μC Reset</td>
<td>3</td>
<td>none</td>
</tr>
<tr>
<td>3.3 V</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>TCK</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>TMS</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>TDO</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>TDI</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>RTCK</td>
<td>9</td>
<td>none</td>
</tr>
<tr>
<td>TRST</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

### 3.8 microSD™ Slot

Among others, the microSD card can be used for system updates. It is connected to the gridARM microcontroller via the SPI bus.
Note: A plugged-in microSD card blocks the communication with further devices that are connected to the SPI bus (e.g. internally connected: 4 MB Flash-ROM). Remove the microSD card in order to allow the communication with further devices again.

3.9 Text Display Connector

At P400 and P401, a HD44780-compatible LC text display can be connected. The pin assignment of such a display fits to the one on the evaluation board.

The LCD contrast can be adjusted with the corresponding potentiometer.
Pin headers P400 and P401 for a text display and potentiometer for contrast adjustment

<table>
<thead>
<tr>
<th>Pin at text display</th>
<th>Pin at P400/P401</th>
<th>Name</th>
<th>Level</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>P401-1</td>
<td>Vss</td>
<td>GND</td>
<td>Ground</td>
</tr>
<tr>
<td>2</td>
<td>P401-2</td>
<td>Vdd</td>
<td>5 V or 3.3 V</td>
<td>Supply</td>
</tr>
<tr>
<td>3</td>
<td>P401-3</td>
<td>Vo</td>
<td>0 - 1.5 V</td>
<td>Contrast voltage</td>
</tr>
<tr>
<td>4</td>
<td>P401-4</td>
<td>RS</td>
<td>High/Low</td>
<td>Register Select</td>
</tr>
<tr>
<td>5</td>
<td>P401-5</td>
<td>R/W</td>
<td>High/Low</td>
<td>High: read, Low: write</td>
</tr>
<tr>
<td>6</td>
<td>P401-6</td>
<td>E</td>
<td>High</td>
<td>Enable</td>
</tr>
<tr>
<td>7</td>
<td>P401-7</td>
<td>D0</td>
<td>High/Low</td>
<td>Data 0 (LSB)</td>
</tr>
<tr>
<td>8</td>
<td>P401-8</td>
<td>D1</td>
<td>High/Low</td>
<td>Data 1</td>
</tr>
<tr>
<td>9</td>
<td>P401-9</td>
<td>D2</td>
<td>High/Low</td>
<td>Data 2</td>
</tr>
<tr>
<td>10</td>
<td>P401-10</td>
<td>D3</td>
<td>High/Low</td>
<td>Data 3</td>
</tr>
<tr>
<td>11</td>
<td>P401-11</td>
<td>D4</td>
<td>High/Low</td>
<td>Data 4</td>
</tr>
<tr>
<td>12</td>
<td>P401-12</td>
<td>D5</td>
<td>High/Low</td>
<td>Data 5</td>
</tr>
<tr>
<td>13</td>
<td>P401-13</td>
<td>D6</td>
<td>High/Low</td>
<td>Data 6</td>
</tr>
<tr>
<td>14</td>
<td>P401-14</td>
<td>D7</td>
<td>High/Low</td>
<td>Data 7 (MSB)</td>
</tr>
<tr>
<td>15</td>
<td>P400-1</td>
<td>LED+</td>
<td>5 V or 3.3 V</td>
<td>LED background light +</td>
</tr>
<tr>
<td>16</td>
<td>P400-2</td>
<td>LED-</td>
<td>GND</td>
<td>LED background light -</td>
</tr>
</tbody>
</table>

Connector pins of a text display
3.10 I²C/SPI

Via the pin header H400, the signals of the communication busses I²C and SPI can be accessed directly. The pin header is not equipped at delivery. The pins have the following functions:

![Diagram showing I²C and SPI pins](image)

Attention! The pins are directly connected to the microcontroller and not protected. Incorrect use can cause damage to the microcontroller.

On the evaluation board, the I²C bus internally connects the following peripherals with 400 kbit/s:

- System EEPROM (256 byte)
- Buzzer
- microSD card detection
- Real-time clock (RTC)
- RTC alarm input
- 7 freely available LEDs (see 3.11 on page 27)
- 6 freely available push buttons (see 3.11 on page 27)
On the evaluation board, the SPI bus internally connects the following peripherals:

- microSD card
- 4 MByte flash ROM

### 3.11 Digital inputs and outputs

The I²C bus connects six push buttons, the detection of a microSD card, and the RTC alarm as digital inputs, and furthermore, seven LEDs and a buzzer as digital outputs. They can be disconnected separately from the inputs and outputs by opening the interconnected jumpers.

The inputs can be directly accessed via the pin header field H402, the outputs via H401.
3.12 Analog Inputs

Five analog inputs are directly led out from the gridARM microcontroller. The reference voltage is 3.3 V. Four of the analog inputs are equipped with potentiometers.

A direct access to the analog inputs is provided by the pin header field H403. To do this, the potentiometers must be separated from the inputs by opening the interconnected jumpers.

Analog input 5 can optionally be used for measuring the VddBU voltage (button cell). To do this, the solder field R454 (see figure below) must be shorted by a solder jumper.

<table>
<thead>
<tr>
<th>Pin at H403</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Analog input 1 (μC AD0)</td>
</tr>
<tr>
<td>2</td>
<td>Analog input 2 (μC AD1)</td>
</tr>
<tr>
<td>3</td>
<td>Analog input 3 (μC AD2)</td>
</tr>
<tr>
<td>4</td>
<td>Analog input 4 (μC AD3)</td>
</tr>
<tr>
<td>5</td>
<td>Analog input 5 (μC AD4)</td>
</tr>
<tr>
<td>6</td>
<td>Not used</td>
</tr>
<tr>
<td>7</td>
<td>Not used</td>
</tr>
<tr>
<td>8</td>
<td>Not used</td>
</tr>
<tr>
<td>Pin at H403</td>
<td>Function</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>9</td>
<td>GND analog (at delivery connected to GND)</td>
</tr>
<tr>
<td>10</td>
<td>3.3 V analog</td>
</tr>
</tbody>
</table>

Attention! The pins are directly connected to the microcontroller and not protected. Incorrect use can cause damage to the microcontroller.

Note: The gridARM microcontroller is also available in a BGA package with eight analog inputs.

3.13 Status LED Microcontroller

At delivery, the “μC status” LED is connected to port PA15 of the gridARM microcontroller and it is freely programmable. For example, it can be used as heartbeat indicator.

If the LED doesn't react when controlled by Port PA15, this may be due to the jumper configuration (see 4.4 Using Microcontroller LED or USB Detection on page 34).
3.14 Real-Time Clock

The real-time clock is connected to the gridARM microcontroller via the I²C bus. It is buffered by the button cell if no supply voltage is applied (JP429 set to 3-2).

The alarm signal of the real-time clock can be read via the I/O module at the I²C bus (IO_B7).

3.15 Reset Push Button

Pressing the reset push button initiates a reboot of the PEAK-gridARM Evaluation Board.
4 Configuration

On the PEAK-gridARM Evaluation Board, several hardware functions can be configured:

- Selecting the Boot Flash (below)
- Activating the Write Protection for the System EEPROM (on page 32)
- Using I/O Interrupt or microSD™ Card Detection (on page 33)
- Using Microcontroller LED or USB Detection (on page 34)
- Assigning Traffic LEDs (on page 35)
- Replacing the Button Cell for the Real-Time Clock (on page 37)

4.1 Selecting the Boot Flash

On start of the microcontroller a bootloader is loaded from either the 16-bit or the 32-bit NOR flash. This is determined by the jumpers JP200 and JP201.

The U-Boot bootloader can only be loaded from the 16-bit NOR flash.

If jumper JP104 is open (port BMS receives High signal), the bootloader contained in the gridARM microcontroller is started instead of an external one.
Jumpers JP200, JP201, and JP104

<table>
<thead>
<tr>
<th>Jumper setting JP200 and JP201</th>
<th>NOR flash selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>closed* 3-2*</td>
<td>16-bit at CS0 and 32-bit at CS2</td>
</tr>
<tr>
<td>open 2-1</td>
<td>32-bit at CS0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Jumper setting JP104</th>
<th>Bootloader</th>
</tr>
</thead>
<tbody>
<tr>
<td>open</td>
<td>Internal of the gridARM µC</td>
</tr>
<tr>
<td>closed*</td>
<td>From external flash (CS0)</td>
</tr>
</tbody>
</table>

* Setting at delivery

4.2 Activating the Write Protection for the System EEPROM

The system EEPROM (256 Byte, I²C connection) can be write-protected with jumper JP430. In case of a microcontroller-internal boot, it contains the boot order in the first 128 bytes. If using the U-
Boot bootloader (setting at delivery), the system EEPROM is not needed and therefore completely available to the user.

<table>
<thead>
<tr>
<th>Jumper setting</th>
<th>Write protection</th>
</tr>
</thead>
<tbody>
<tr>
<td>open*</td>
<td>deactivated</td>
</tr>
<tr>
<td>closed</td>
<td>activated</td>
</tr>
</tbody>
</table>

* Setting at delivery

4.3 Using I/O Interrupt or microSD™ Card Detection

Usually, the detection of a inserted microSD card is done via the I/O module at the I²C bus (IO_B6). If required, the PA02 port at the gridARM microcontroller can be used alternatively for this purpose. To do so, jumper JP431 must be set over. In that case, the interrupt
line is disconnected from the I/O module and an interrupt signal is not available anymore.

<table>
<thead>
<tr>
<th>Jumper setting</th>
<th>Port PA02 connected to</th>
</tr>
</thead>
<tbody>
<tr>
<td>JP431</td>
<td>1-2* Interrupt I²C I/O</td>
</tr>
<tr>
<td></td>
<td>2-3 microSD card detection</td>
</tr>
</tbody>
</table>

* Setting at delivery

4.4 Using Microcontroller LED or USB Detection

Port PA15 of the gridARM microcontroller is assigned to the “μC Status” LED (D407) at delivery. Alternatively, port PA15 can be used as input for detection of a plugged USB cable at the evaluation board (not useful if supply is done via the USB port). To do so, jumper JP432 must be set over.
Jumper JP432

<table>
<thead>
<tr>
<th>Jumper setting</th>
<th>Port PA15 connected to</th>
</tr>
</thead>
<tbody>
<tr>
<td>JP432 1-2*</td>
<td>“μC Status” LED</td>
</tr>
<tr>
<td>JP432 2-3</td>
<td>USB detection</td>
</tr>
</tbody>
</table>

* Setting at delivery

### 4.5 Assigning Traffic LEDs

Two of the four traffic LEDs can be assigned either to the CAN communication or to the communication via RS-232 Debug. This is done with the jumpers JP505 and JP506.

The remaining two traffic LEDs are permanently assigned to the communication via RS-232 Terminal.
Jumpers JP505 and JP506 and traffic LEDs

<table>
<thead>
<tr>
<th>Jumper setting</th>
<th>Traffic indication for…</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-1*</td>
<td>CAN Tx</td>
</tr>
<tr>
<td>2-1*</td>
<td>CAN Rx</td>
</tr>
<tr>
<td>3-2</td>
<td>RS-232 Debug TxD</td>
</tr>
<tr>
<td>3-2</td>
<td>RS-232 Debug RxD</td>
</tr>
</tbody>
</table>

* Setting at delivery
4.6 Replacing the Button Cell for the Real-Time Clock

The real-time clock (RTC) used by the evaluation board or the timer as part of the gridARM microcontroller are supplied by a button cell of the IEC type CR 2032 (3 V), as long as the evaluation board is turned off.

A new button cell lasts several years. If the real-time clock indicates an unexpected time, take out the button cell and measure its voltage. This should be around the nominal 3.0 Volts. If the measured voltage is lower than 2.5 Volts, you should replace the button cell with a fresh one.

Afterwards, the real-time clock must be set again.
# 5 Technical Specifications

## Power supply

<table>
<thead>
<tr>
<th>Description</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supply voltage</td>
<td>12 V DC nominal, 8 - 30 V possible</td>
</tr>
<tr>
<td>Current consumption (at 12 V)</td>
<td>U-Boot idling: 100 mA</td>
</tr>
<tr>
<td></td>
<td>Linux kernel idling: 150 - 180 mA</td>
</tr>
<tr>
<td>Buffer battery for the real-time clock or VddBU</td>
<td>Button cell CR 2032, 3 V</td>
</tr>
</tbody>
</table>

## gridARM microcontroller

<table>
<thead>
<tr>
<th>Description</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor</td>
<td>ARM7TDMI® 32-Bit</td>
</tr>
<tr>
<td>Volatges</td>
<td>Core: 1.3 V; I/O: 3.0 V</td>
</tr>
<tr>
<td>Clock frequency</td>
<td>80 MHz</td>
</tr>
<tr>
<td>On-chip memory</td>
<td>256 KByte masked ROM (Grid BIOS, boot code)</td>
</tr>
<tr>
<td></td>
<td>160 KByte SRAM</td>
</tr>
<tr>
<td>Interrupt handling</td>
<td>Advanced Interrupt Controller (AIC)</td>
</tr>
<tr>
<td>Manufacturing</td>
<td>130-nm technology</td>
</tr>
<tr>
<td>Packages</td>
<td>QFP 208-pin or BGA 225-pin, RoHS</td>
</tr>
<tr>
<td>Further technical specifications</td>
<td>See data sheets for the gridARM microcontroller&lt;sup&gt;4&lt;/sup&gt;</td>
</tr>
</tbody>
</table>

## Gigabit Ethernet

<table>
<thead>
<tr>
<th>Description</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit rates</td>
<td>10 Mbit/s, 100 Mbit/s, 1 Gbit/s</td>
</tr>
<tr>
<td>Phy</td>
<td>Micrel KSZ9021RLI (RGMII, MDIO/MDC)</td>
</tr>
</tbody>
</table>

## CAN

<table>
<thead>
<tr>
<th>Description</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specification</td>
<td>ISO 11898-2, High-speed CAN</td>
</tr>
<tr>
<td></td>
<td>2.0A (Standard format) and 2.0B (Extended format)</td>
</tr>
<tr>
<td>Bit rates</td>
<td>up to 1 Mbit/s</td>
</tr>
<tr>
<td>Transceiver</td>
<td>NXP PCA82C251</td>
</tr>
<tr>
<td>Termination</td>
<td>120 Ω, switchable on board</td>
</tr>
</tbody>
</table>


### SPI Serial Flash ROM
- **Size**: 4 MByte
- **SPI modes**: 0 and 3
- **Chip Erase Time**: 35 ms (typical)
- **Sector/Block Erase Time**: 18 ms (typical)
- **Byte Program Time**: 7 μs (typical)

### Parallel Flash ROM
- **Size**: 4 MByte, 16-bit
  8 MByte, 32-bit
- **Interface**: Common Flash Interface (CFI)
- **Access time**: 70 ns

### SDRAM
- **Size**: 64 MByte, 32-bit
- **Organization**: 4 Mbit x 16 I/O x 4 bank
- **Timing**: 166 MHz 3-3-3 PC166
- **Cycle**: 6 ns

### I/O
- **Digital inputs (via I²C)**: 8: 6 push buttons, microSD card detection, RTC alarm
- **Digital outputs (via I²C)**: 8: 7 LEDs, 1 buzzer
- **Analog inputs**: 5: 5 potentiometers, 1 alternative for voltage measurement of the button cell
- **All I/Os**: Also direct access possible

### Measures
- **Size**: 110 x 110 mm (L x W)
  See also dimension drawing Appendix A on page 40
- **Weight**: 125 g

### Environment
- **Operating temperature**: -40 - +85 °C (-40 - +185 °F)
- **Temperature for storage and transport**: -40 - +100 °C (-40 - +212 °F)
- **Relative humidity**: 15 - 90 %, not condensing
Appendix A Dimension Drawing

The figure does not show the original size.
Appendix B Jumpers Overview

Jumpers on the top side
Jumpers on the bottom side
<table>
<thead>
<tr>
<th>Jumper</th>
<th>Setting</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>JP100</td>
<td>1-2*</td>
<td>VddBU 1.2 V from the supply</td>
</tr>
<tr>
<td></td>
<td>2-3</td>
<td>VddBU 1.2 V from the LDO/button cell (see JP429)</td>
</tr>
<tr>
<td>JP101</td>
<td>open*</td>
<td>Wake-up by switching between the two states</td>
</tr>
<tr>
<td></td>
<td>closed</td>
<td></td>
</tr>
<tr>
<td>JP104</td>
<td>open</td>
<td>Internal bootloader of the gridARM µC</td>
</tr>
<tr>
<td></td>
<td>closed*</td>
<td>Boot from external flash (CS0)</td>
</tr>
<tr>
<td>JP200 +</td>
<td>open + 1-</td>
<td>32-bit NOR flash at CS0</td>
</tr>
<tr>
<td>JP201</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>closed +</td>
<td>16-bit NOR flash at CS0 and 32 bit NOR flash at CS2</td>
</tr>
<tr>
<td></td>
<td>2-3*</td>
<td></td>
</tr>
<tr>
<td>JP202</td>
<td>open*</td>
<td>Boot block of 16-bit NOR flash writable</td>
</tr>
<tr>
<td></td>
<td>closed</td>
<td>Boot block of 16-bit NOR flash write-protected</td>
</tr>
<tr>
<td>JP203</td>
<td>open*</td>
<td>Boot block of 32-bit NOR flash writable</td>
</tr>
<tr>
<td></td>
<td>closed</td>
<td>Boot block of 32-bit NOR flash write-protected</td>
</tr>
<tr>
<td>JP400 -</td>
<td>open</td>
<td>Analog inputs: external voltages on H403 pins 1 - 5 (0 - 3.3 V)</td>
</tr>
<tr>
<td>JP404</td>
<td>closed*</td>
<td>Analog inputs: pots RV401 - RV404 (Do not apply external voltages!)</td>
</tr>
<tr>
<td>JP411</td>
<td>closed*</td>
<td>Digital outputs A1 - A7: LEDs D400 - D406 (Do not apply external voltages!)</td>
</tr>
<tr>
<td>JP412</td>
<td>open</td>
<td>Digital output A8: forwarding to H401 pin 8</td>
</tr>
<tr>
<td></td>
<td>closed*</td>
<td>Digital output A8: beeper SP400 (Do not apply external voltages!)</td>
</tr>
<tr>
<td>JP413 -</td>
<td>open</td>
<td>Digital inputs B1 - B6: external voltages on H403 pins 1 - 6 (0 - 3.3 V)</td>
</tr>
<tr>
<td>JP418</td>
<td>closed*</td>
<td>Digital inputs B1 - B6: push buttons PB400 - PB405 (Do not apply external voltages!)</td>
</tr>
<tr>
<td>JP419</td>
<td>open</td>
<td>Digital input B7: external voltage on H402 pin 7 (0 - 3.3 V); tip: set JP431 to 2-3</td>
</tr>
<tr>
<td></td>
<td>closed*</td>
<td>Digital input B7: microSD card detection via I²C I/O module</td>
</tr>
<tr>
<td>JP420</td>
<td>open</td>
<td>Digital input B8: RTC cannot trigger alarm</td>
</tr>
<tr>
<td></td>
<td>closed*</td>
<td>Digital input B8: RTC alarm via I²C I/O module</td>
</tr>
<tr>
<td>JP427</td>
<td>1-2</td>
<td>Supply LC display 3.3 V</td>
</tr>
<tr>
<td></td>
<td>2-3*</td>
<td>Supply LC display 5 V</td>
</tr>
<tr>
<td>Jumper</td>
<td>Setting</td>
<td>Function</td>
</tr>
<tr>
<td>---------</td>
<td>---------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td>JP429</td>
<td>1-2</td>
<td>Real-time counter in the microcontroller is supplied by the button cell (only useful with JP100 on 2-3)</td>
</tr>
<tr>
<td></td>
<td>2-3*</td>
<td>RTC at the I²C bus is supplied by the button cell</td>
</tr>
<tr>
<td>JP430</td>
<td>open*</td>
<td>System EEPROM writable</td>
</tr>
<tr>
<td></td>
<td>closed</td>
<td>System EEPROM write-protected</td>
</tr>
<tr>
<td>JP431</td>
<td>1-2*</td>
<td>PA02: interrupt from I²C I/O module</td>
</tr>
<tr>
<td></td>
<td>2-3</td>
<td>PA02: microSD card detection</td>
</tr>
<tr>
<td>JP432</td>
<td>1-2*</td>
<td>PA15: output for μC status LED</td>
</tr>
<tr>
<td></td>
<td>2-3</td>
<td>PA15: input for USB detection</td>
</tr>
<tr>
<td>JP500</td>
<td>1-2</td>
<td>CAN transceiver supply 3.3 V (e.g. TI SN56HVD230)</td>
</tr>
<tr>
<td></td>
<td>2-3*</td>
<td>CAN transceiver supply 5 V (e.g. NXP 82C251)</td>
</tr>
<tr>
<td>JP505 +</td>
<td>1-2 + 1-2</td>
<td>LEDs: CAN bus</td>
</tr>
<tr>
<td>JP506</td>
<td></td>
<td>2-3 + 2-3* LEDs: RS-232 Debug</td>
</tr>
<tr>
<td>JP600</td>
<td>1-2*</td>
<td>Board supply from USB</td>
</tr>
<tr>
<td></td>
<td>2-3</td>
<td>Board supply via supply socket (8 - 30 V DC)</td>
</tr>
</tbody>
</table>

* Setting at delivery
Appendix C Circuit Diagrams

On the following pages, the circuit diagrams of the PEAK-gridARM Evaluation Board are enlisted, split up in the following sections:

- Microcontroller
- Memory
- Ethernet
- I/O
- Serial
- Power Supply
Set jumper JP500 to pos 1-2 (=3.3V supply)
Using a 3.3V CAN transceiver (U500) like TI's SN65HVD230:

Remark: CAN levels might be slightly reduced.

Set jumper JP500 to pos 2-3 (=5V supply)
Using a 5V CAN transceiver (U500) like NXP's 82C251:

(These parts optional in own designs)