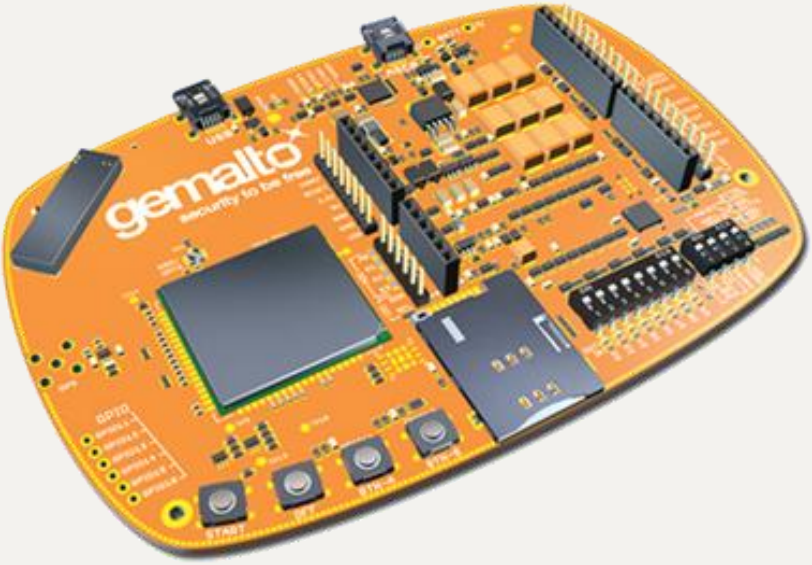
Cinterion® Concept Board

Hardware Interface Description

Version: 01a
DocID: concept_board_hd_v01a



Document Name: **Cinterion® Concept Board Hardware Interface Description**

Version: **01a**

Date: **December 20, 2013**

DocId: **concept_board_hd_v01a**

Status: **Confidential / Draft**

Contents

Tables

Figures

1 Introduction

The Cinterion® Concept Board is a development tool engineered by Gemalto which enables programmers worldwide to quickly build M2M and Internet of Things (IoT) prototypes.

Thanks to its Shield Interface (following Arduino), little to no hardware expertise is needed to connect all kinds of sensors and actuators. Wireless connectivity is provided by the Cinterion® EHS6 module and the on-board penta-band antenna.

Applications are written in Java using widely spread tools such as Eclipse and NetBeans.

This document describes the hardware of the Cinterion® Concept Board. It helps you quickly retrieve interface specifications, electrical and mechanical details.

1.1 Key features at a Glance

Figure 1 shows the main Concept Board features:

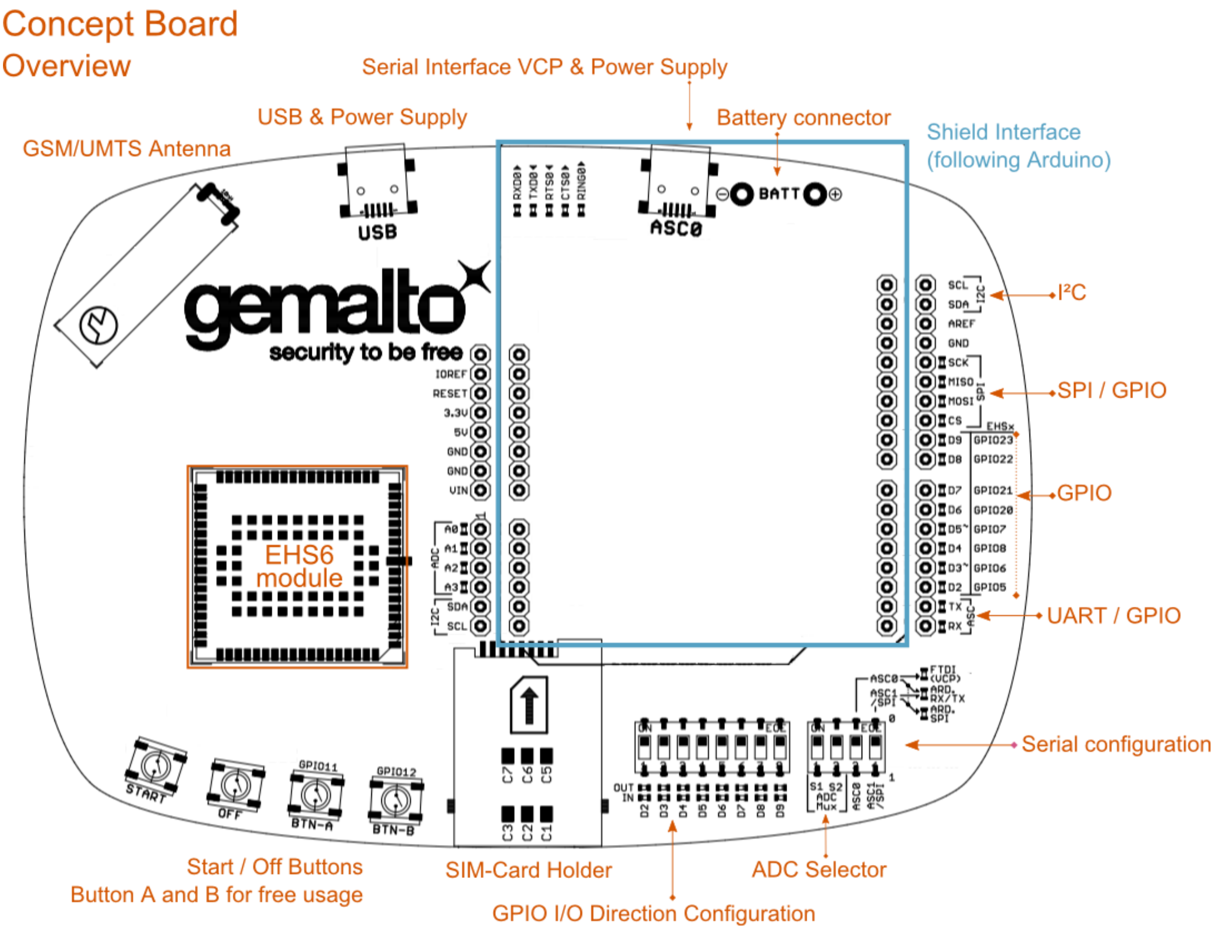


Figure 1: Concept Board feature overview

Table 1: Key features at a glance

Feature	Implementation
Frequency bands	UMTS/HSPA+: Five band 800/850/900/1900/2100MHz GSM/GPRS/EDGE: Quad band 850/900/1800/1900MHz For further EHS6 Module details see [1]
SIM interface	Supported SIM cards: 3.0V, 1.8V
Antenna interface	Penta-band GSM/UMTS RF antenna
Extension interface	Extension of Concept Board hardware through the Shield Interface
Power supply	USB interface and Supports Li+ battery pack through 2-pin connector on board
Serial interface	EHS6 ASC0 via Virtual COM Port (VCP) Maximum baud rate 921kbps
USB interface	Supported
User buttons	4 buttons: - START - OFF - User button BTN-A - User button BTN-B
LEDs	Current configuration is displayed via on board LEDs.
Configuration	Two switch banks allow manual configuration of GPIO signal direction, ATC multiplexing and SPI/ASC sharing configuration. SW controlled configuration via CCU (Configuration Control Unit).

1.2 System Overview

At the core of the Concept Board there is a Cinterion® EHS6 module. The module acts both as a GSM/UMTS modem, and as Java™ application processor running a J2ME Java Virtual Machine.

Due to its low power architecture, the EHS6 module interfaces (such as GPIOs, SPI, etc.) operate at 1.8V. On the other hand, the Shield Interface operates at 5V. A level adaptation layer shifts the different operating voltage levels. These level shifters have to be set in the direction needed for correct shield operation which can be done via the Control Switch Bank or via I2C command by a Configuration and Control Unit (CCU).

Figure 2 shows the Concept Board system architecture.

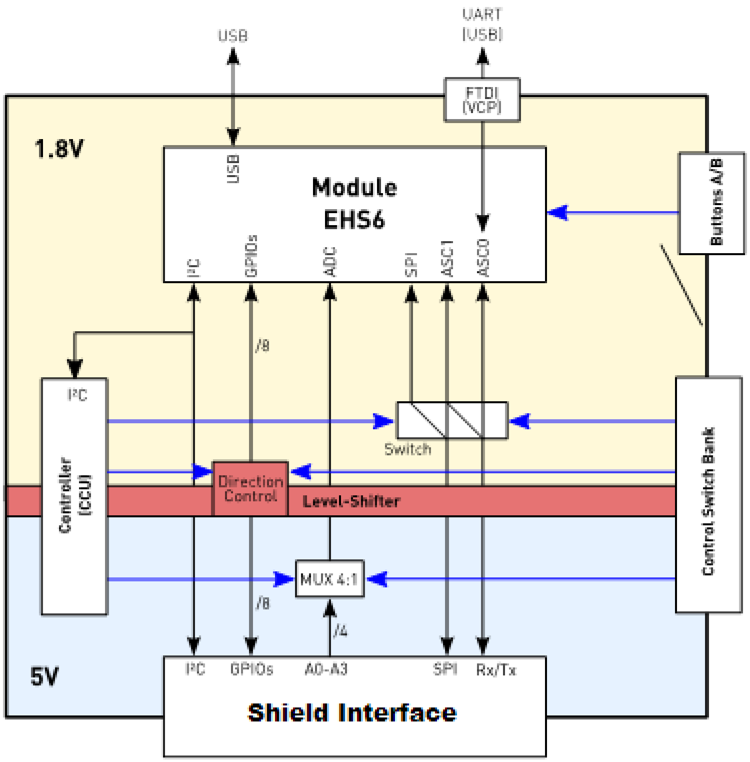


Figure 2: Concept Board architecture overview

2 Interface Characteristics

2.1 Shield Interface

The Concept Board supports a 32 pin extension interface called the Shield Interface. The Shield Interface allows the user to conveniently connect external hardware components (such as sensors, actuators, shields, etc.).

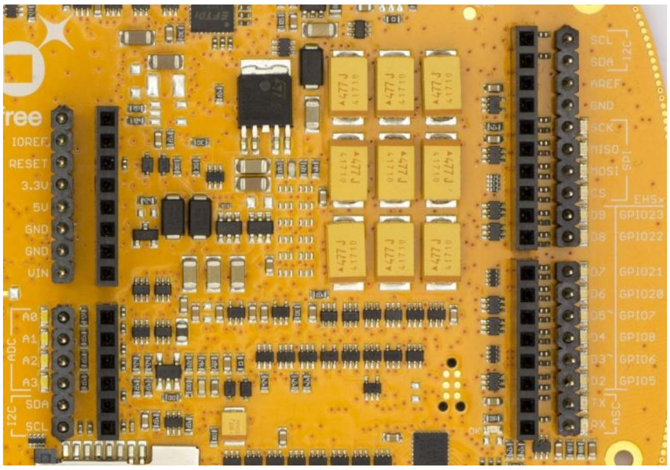


Figure 3: Shield Interface (top view)

Table 2 shows the pin assignment of the Shield Interface, as well as the corresponding pin of the EHS6 module where it is connected to. The DIR column shows whether this pin is input, output or bidirectional.

Table 2: Shield Interface pin assignments and their properties

Name	EHS6	DIR	Description	Properties
NC	-		Do not use	
IOREF	-	O	5V	V_{usb} – 0.25V
RESET	EMERG_RST	I	Reset signal for EHS6	Active low >10ms reset EHS6 $V_I$max = 0.5V; 47K pull up to 5V
3.3V	-	O	Shield supply 3.3V	3.3V / 200mAmax
5V	-	O	Shield supply 5V	V_{usb} – 0.25V
GND	-		Main Ground	
Vin	-	O	Connected to 5V	
A0	ADC1	I	ADC multiplexed 1:4	$V_I$max = 5V; R_i = 120kΩ, t_{sw} = 30ms, t_{conv} = TBD
A1	ADC1	I	ADC multiplexed 1:4	$V_I$max = 5V; R_i = 120kΩ, t_{sw} = 30ms, t_{conv} = TBD
A2	ADC1	I	ADC multiplexed 1:4	$V_I$max = 5V; Ri = 120kΩ, t_{sw} = 30ms, t_{conv} = TBD
A3	ADC1	I	ADC multiplexed 1:4	$V_I$max = 5V; Ri = 120kΩ, t_{sw} = 30ms, t_{conv} = TBD
SDA	SDA	IO	I^2C data signal	R_{pu} = 4.7KΩ to 5V
SCL	SCL	O	I^2C clock signal	R_{pu} = 4.7KΩ to 5V
SCL	SCL	O	I^2C clock signal	R_{pu} = 4.7KΩ to 5V
SDA	SDA	IO	I^2C data signal	R_{pu} = 4.7KΩ to 5V
AREF	-		Not connected	-
SCK	SCK / DSR0	O	SPI Clock out, Alternate UART flow control	5V Push pull; I_O = +-50mA
MISO	MISO / GPI17	I	SPI Data input Alternate GPI	I_{in}max = +-2µA

MOSI	MOSI / GPO16	O	SPI Data onput Alternate GPO	5V Push pull; I_O = +-50mA; t_{osw} = 35µs
CS	CS / GPO19	O	SPI CS Alternate GPO	5V Push pull; I_O =+-50mA; t_{osw} = 35µs
D9	GPIO23	IO	GPIO	OUT: 5V, I_o= +-50mA; IN: $I_{in}max$ = +-2µA t_{osw} = 35µs, t_{dsw} = 10ms
D8	GPIO22	IO	GPIO	OUT: 5V, I_o= +-50mA; IN: $I_{in}max$ = +-2µA t_{osw} = 35µs, t_{dsw} = 10ms
D7	GPIO21	IO	GPIO	OUT: 5V, I_o = +-50mA; IN: $I_{in}max$ = +-2µA t_{osw} = 35µs, t_{dsw} = 10ms
D6	GPIO20	IO	GPIO	OUT: 5V, I_o = +-50mA; IN: $I_{in}max$ = +-2µA t_{osw} = 35µs, t_{dsw} = 10ms
D5	GPIO7	IO	GPIO	OUT: 5V, I_o = +-50mA; IN: $I_{in}max$ = +-2µA t_{osw} = 35µs, t_{dsw} = 10ms
D4	GPIO8	IO	GPIO	OUT: 5V, I_o = +-50mA; IN: $I_{in}max$ = +-2µA t_{osw} = 35µs, t_{dsw} = 10ms
D3	GPIO6	IO	GPIO	OUT: 5V, I_o = +-50mA; IN: $I_{in}max$ = +-2µA t_{osw} = 35µs, t_{dsw} = 10ms
D2	GPIO5	IO	GPIO	OUT: 5V, I_o = +-50mA; IN: $I_{in}max$ = +-2µA t_{osw} = 35µs, t_{dsw} = 10ms
TX	RxD0 / RxD1	O	UART ASC0 / UART ASC1	5V Push pull; I_o = +-50mA t_{osw} = 35µs
RX	TxD0 / TxD1	I	UART ASC0 UART ASC1	$I_{in}max$ = +-2µA

t_{msw} = multiplexer switching time; including I^2C Java connection setup
t_{dsw} = direction change switching time; toggling in/out, including I^2C Java connection setup
t_{osw} = output change switching time; toggling L/H
t_{conv} = ADC conversion time

2.1.1 UART

The Shield Interface provides a UART port (RxD/TxD) which supports baudrates up to 921kbit/s. The UART port is connected to the EHS6 ASC1 serial port by default. Depending on the user requirements (for instance, if SPI functionality is required), the Control Switch Bank can be used to route ASC0 to the UART port instead, allowing the SPI functionality to be available at the Shield Interface.

For further details please refer to section 3.1.

2.1.2 I^2C

This interface is always active; both CCU and the Shield Interface are connected to through the I^2C bus to the EHS6 module.

When using the CCU, the I^2C data traffic is also present at the Shield Interface. For details on the I^2C protocol used to control the CCU refer to section 0.

2.1.3 SPI

A four line SPI Master interface is supported by the Concept Board. By default, the SPI port of the EHS6 module is deactivated.

In order to use SPI functionality, the SPI port has to be activated by your Java program, or through the following AT command:

```
AT^SSPI=1,0,[speed],0,0,[mode]
```

Table 3: SPI settings

Speed setting	**Data transfer rate**
0000	100 kbps
0001	250 kbps
0010	500 kbps
0011	1.083 Mbps
0100	3.25 Mbps
0101	6.5 Mbps
Mode setting	**Description**
0000	Mode 0; CPOL=0, CPHA=0
0001	Mode 1; CPOL=0, CPHA=1
0010	Mode 2; CPOL=1, CPHA=0
0011	Mode 3; CPOL=1, CPHA=1

The level shifters for the SPI interface are already set in right direction and cannot be changed.

2.1.4 GPIO

The Concept Board supports up to 11 GPIOs whereas 3 GPIOs have fixed direction which cannot be changed (shared with the SPI port).

Out of the 11 GPIOs, 8 are input/output, 2 are fixed outputs, and 1 is a fixed input.

Table 4: GPIO pin assignment

Shield Interface GPIO	**EHS6 GPIO**
D2	GPIO5
D3	GPIO6
D4	GPIO8
D5	GPIO7
D6	GPIO20
D7	GPIO21
D8	GPIO22
D9	GPIO23
CS	GPO19
MOSI	GPO16
MISO	GPI17

2.1.5 ADC

The 4 ADC channels ADC0…ADC3 are multiplexed to one EHS6 ADC channel. Choosing one channel out of four can be done either by the Control Switch Bank or through the CCU.

The EHS6 conversion result is given in mV with a limit of 1200mV. An adaptation between the 5V Shield Interface and the 1.2V EHS6 is implemented which means the full scale 1200mV equals the full scale 5V on the Shield Interface. Therefore, the **division factor 4.17** has to be used when programming ADC-based applications in Java.

2.1.6 Power supply

A chargeable Lithium+ battery pack can be connected / soldered on the BATT pads. The on-board circuit charges the battery pack up to 4.1V with max 250mA. This battery pack can operate as the main power source when the USB power is not connected. In such case, the 5V Shield Interface supply is drawn from the battery pack, and is therefore equal to the battery voltage (allowed range: 4.1…3.3V).

The user is ultimately responsible to dimension the battery. Due to the current consumption during GSM operation, we recommend to use a battery of capacity >800mAh.

External battery shields may be used. In that case, the Diode D10 on the Concept Board has to be replaced by a 0Ω resistor.

2.2 Serial communication interfaces

Two USB connectors are present on the board. The connector named “USB” connects directly to the EHS6 USB port.

EHS6 enumerates, i.e., registers with the USB host, as a CDC-ACM Composite USB device supporting eight separate interfaces. This is the *Composite Communication* enumeration.

Under Microsoft® Windows XP™, Microsoft® Windows Vista® and Microsoft® Windows 7 this USB device implements a Modem port (a virtual USB Modem port) plus six further communication ports as well as a reserved port. The Modem port as well as the ports enumerated 3 to 5 are AT command interfaces, i.e., these ports are accessible by AT command and may be used as control and data interfaces.

The interfaces - with the exception of the reserved port - are controlled by separate device drivers running on the Windows host. These drivers may be standard USB drivers integrated in Windows. The driver configuration files for Windows (.inf files) are supplied by Gemalto and need to be installed before the device can be used.

The second one named “ASC0” serves a virtual COM port chipset (FTDI FT232R). Depending on the status of the Configuration Switch Bank, this ASC0 VCP is active or mapped to the Shield Interface. The red LED under “ASC0” displays the current configuration. When turned on, ASC0 is active on USB VCP; when off, ASC0 is disconnected from the VCP while the VCP itself is still activated on PC side.

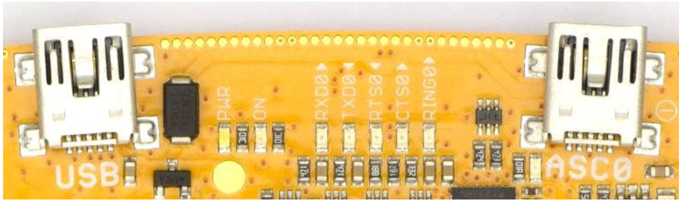


Figure 4: USB and ASC0 connection

2.3 Configuration Control

2.3.1 Manual control of SPI/ASC1 sharing and ADC multiplexing

The Configuration Switch Bank controls the SPI/ASC1 sharing and the ADC multiplexing. The settings are displayed via LEDs. The settings can be overruled by the Configuration Control Unit (CCU) via I^2C commands. Refer to chapter 0 for details.

In order to get manual control over the configuration, the CCU has to release the switch settings via an I^2C command.

Important note: If the CCU is used to control the interface configuration, the LED display may not fit to the actual Configuration Switch Bank settings. This happens because the CCU overrules any switch settings. In any case, the LEDs always show the correct settings.

On the right switch bank (1-4)
- the first two switches (1 and 2) are selecting the active Shield Interface ADC channel which is displayed by the LEDs A0…A3
- the last two switches (3 and 4) are selecting the ASC0, ASC1, SPI configuration, see table below

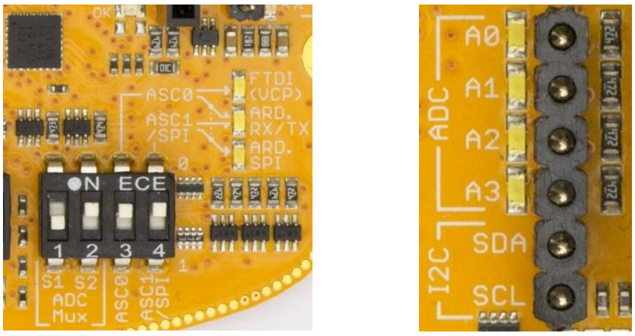
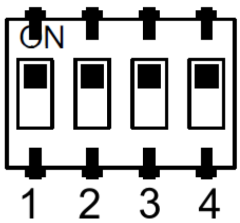


Figure 5: configuration switch bank and corresponding LEDs

Table 5: Concept Board configuration for ASC/SPI and ADC

<table>
<tr><th colspan="5">ADC</th></tr>
<tr><td>Switch 1 & 2 up</td><td colspan="3">A0</td><td rowspan="9">
</td></tr>
<tr><td>Switch 1 down / 2 up</td><td colspan="3">A1</td></tr>
<tr><td>Switch 1 up / 2 down</td><td colspan="3">A2</td></tr>
<tr><td>Switch 1 & 2 down</td><td colspan="3">A3</td></tr>
<tr><th></th><th>ASC0</th><th>ASC1</th><th>SPI</th></tr>
<tr><td>Switch 3 & 4 up</td><td>USB VCP</td><td>to Shield I/F</td><td>Not used</td></tr>
<tr><td>Switch 3 down / 4 up</td><td>to Shield I/F</td><td>Not used</td><td>to Shield I/F</td></tr>
<tr><td>Switch 3 up / 4 down</td><td>USB VCP</td><td>Not used</td><td>to Shield I/F</td></tr>
<tr><td>Switch 3 & 4 down</td><td colspan="3">Reserved for future use</td></tr>
</table>

2.3.2 Configuration and control via CCU by I^{2}C command

The same manual configuration can be selected via the CCU I^{2}C commands. Refer to section 0 for the CCU I^{2}C command-set.

2.3.3 GPIO direction control via switch bank

The Configuration Switch Bank also controls the direction of the 8 GPIO lines D2~D9. Setting the switches up configures the corresponding GPIO direction as an output towards the Shield Interface, while setting them down configures the GPIO direction as an input.

The settings are displayed through the colour of the LED: yellow means output, while orange means input. These settings can also be overruled by the CCU. Once activated, the CCU has to release the switch settings to allow manual control.

Important note: if the CCU is active, the LED display may not fit to the actual switch bank settings, as the switches were overruled by the CCU. In any case, the LEDs always show the correct settings.

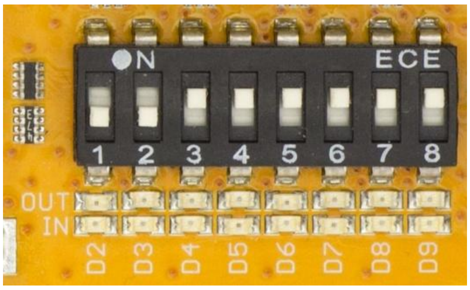
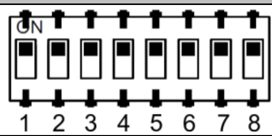


Figure 6: GPIO direction control switch bank and LED

Table 6: Concept Board configuration for GPIO direction

	GPIO1..8	LED	
Switch 1…8 up	Output	*Yellow*	
Switch 1…8 down	Input	*Orange*	

2.4 User buttons

The START and OFF buttons control the operation of the Concept Board.

The START button triggers an ignition signal for the EHS6 module.

The OFF button completely switches the Concept Board off, while the Shield Interface power supply remains active.

BTN-A and BTN-B are connected to EHS6 and can be used as general purpose user buttons.

3 Shield compatibility aspects

There are a number of compatible extension boards (or shields) available in the market. In order to ensure the compatibility of the Concept Board with the shield of your choice, we have listed the aspects that should to be checked beforehand.

3.1 SPI, ASC1, DSR0 sharing

SPI and ASC1 functionality is shared in EHS6. That means, ASC1 is not active when using SPI on the Concept Board and therefore it is disconnected from the Shield Interface.

Alternatively, the ASC0 port of EHS6 can be used in parallel with the SPI port. ASC0 (TxD/RxD) can be mapped to the Shield Interface (instead of the default, ASC1) by simply configuring the switch banks appropriately.

When using SPI on the Concept Board, the flow control signal DSR0 (USB VCP) is not active and can't be used.

3.2 D10~D13 and SPI sharing

As the GPIOs D10~D13 on the Shield Interface are shared with the SPI interface, the D10~D13 signal directions are fixed corresponding to the SPI direction settings.

D10 (CS), D11 (MOSI) and D13 (SCK) can only be used as an OUTPUT, whereas D12 (MISO) can only be used as an INPUT.

3.3 ADC limitations

EHS6 supports one ADC channel. To have four ADC available on the Shield Interface, the EHS6 ADC can be multiplexed (controlled manually or via CCU). This may result in a speed limitation when switching between the channels. Switching and conversion timing are given in Table 2.

3.4 GPIO switching speed limitation

The GPIO switching speed is limited, which should be taken into account when choosing sensor shields. E.g. when using a GPIO signal as a command/data decider, a SPI data throughput might be reduced because of slower command/data transition. Switch timings are given in Table 2.

3.5 GPIO direction speed switching

As the automatic GPIO direction setting is controlled by the CCU, the speed at which a GPIO can change its direction is limited. Refer to Table 2 for the detailed timing.

3.6 Interrupt signals

EHS6 doesn't support interrupt signals, incoming messages / signals have to be polled.

4 Specifications

4.1 Limiting values

Table 7: Absolute maximum ratings

Parameter	Min	Max	Unit
Supply voltage on USB ports	-0.3	5.75	V
Supply voltage on BATT terminal	-0.3	5.5	V
Voltage at 3.3V application supply interface (Shield I/F)	-0.3	3.3	V
Voltage at 5V application supply interface (Shield I/F)	-0.3	5	V
Voltage at application signal interface (Shield I/F)	-0.3	6	V
Sink / source current application signal interface (Shield I/F)	-50	+50	mA
Overall source current application signal interface (Shield I/F)		+150	mA
Environmental temperature	0	35	°C

4.2 Recommended operating conditions

Table 8: recommended operating conditions

Parameter	Min	Max	Unit
Supply voltage on USB ports	4.75	5.25	V
Supply voltage on BATT terminal	3.5	5.5	V

4.3 Static characteristics

Table 9: static characteristics

Parameter	Max	Unit
High level input voltage on application interface @ V_{usb} = 5V	3.36	V
Low level input voltage on application interface @ V_{usb} = 5V	1.44	V
High level output voltage on application interface @ V_{usb} = 5V; I_o = -32mA	4.1	V
Low level output voltage on application interface @ V_{usb} = 5V; I_o = 32mA	0.55	V

5 Regulatory compliance information

This Cinterion® Concept Board is intended is intended for use only in a laboratory test environment. All persons handling the Cinterion® Concept Board must be properly trained in electronics and observe good engineering practice standards.

6 Safety precaution notes

The common safety precautions that apply to mobile phones must also be observed at all times when using this Concept Board. Failure to comply with these precautions violates safety standards. Gemalto M2M assumes no liability for customer's failure to comply with these precautions.

The following is a non-extensive list of the mobile phone and Concept Board usage restrictions:

- Pacemaker patients are advised to keep their hand-held mobile away from the pacemaker while it is on.
- Mobile phones must be switched off before boarding an aircraft.
- Mobile phones may not be operated in the presence of flammable gases or fumes
- Interference can occur if mobile phones are used close to TV sets, radios, computers or inadequately shielded equipment
- Do not use your mobile while driving a vehicle
- SOS You should never rely solely upon any wireless device for essential communications, for example for emergency calls

The power supply connected to the Concept Board shall be in compliance with the SELV requirements defined in EN 60950-1.

7 CCU I²C protocol description

The CCU implements an I²C compatible interface that allows changing the terminal configuration during runtime. The I²C signals are accessible via the module, and also via the extension header (SCL/SDA).

The I²C interface implements the write and the read protocol. The 7-bit device address is 0x69 (binary: 1101001).

7.1 WRITE command

S	Slave Address	W	A	Register Address	A	Data Byte	A	P

Example (Set GPIO8 direction to “output”):

<table>
<tr><td rowspan="2">S</td><td>0x69</td><td>0</td><td rowspan="2">A</td><td rowspan="2">0x12</td><td rowspan="2">A</td><td rowspan="2">0x01</td><td rowspan="2">A</td><td rowspan="2">P</td></tr>
<tr><td colspan="2">0xD2</td></tr>
</table>

7.2 READ command

S	Slave Address	R	A	Data Byte	N	P

Example (Read last status = OK):

<table>
<tr><td rowspan="2">S</td><td>0x69</td><td>1</td><td rowspan="2">A</td><td rowspan="2">0x00</td><td rowspan="2">N</td><td rowspan="2">P</td></tr>
<tr><td colspan="2">0xD3</td></tr>
</table>

(Key: S: Start Condition, W: Write-bit=0, R: Read-bit=1, A: Acknowledge, N: Not Acknowledge, P: Stop Condition).

7.3 I²C Protocol detailed description

In **write-mode**, one address-byte and one data byte is sent to the device. The address-byte specifies a register in which to write the data-byte. The value is only written if it is valid, i.e. in the specified range. After a write attempt, the status code of the operation is saved and the read-address-register (RAR) is automatically set to the status-register address (SR). A subsequent read command will then result the latest status code (see status codes table). Only when the address-byte is the RAR, i.e. another register is selected to be read, the RAR is not automatically set to the SR register.

In **read-mode**, one data bye can be read from the device. Attempts to read more bytes will result in undefined values to be returned by the device. The device will always return the value that is addressed by the RAR register. To read a specific register, a Write-command with RAR as the address-byte and the register-to-be-read as the data-byte has to be issued first. The next read will return the value at this address. Note that there are only a few registers that can be read (see register table). When the RAR is written with a non-read address, the RAR is set to the SR, and the status code ILLEGAL_ARGUMENT is saved. Note that the next read will not be valid, as the return value will be ILLEGAL_ARGUMENT, but the caller cannot determine if the result is the value at the faulty address or an error status code.

7.4 Register Table

Address	Read/ Write	Description	Name	Non-Volatile	Default	Value Range
0x00	R	Status	SR	-	OK	See result codes table
0x10	W	GPIO5	GPIOxR	No	F	0: Input 1: Output 0xFF: Release for manual configuration
0x11	W	GPIO6			F	
0x12	W	GPIO8			F	
0x13	W	GPIO7			F	
0x14	W	GPIO20			F	
0x15	W	GPIO21			F	
0x16	W	GPIO22			F	
0x17	W	GPIO23			F	
0x30	R	Read GPIO manual switch configuration. [23 22 21 20 7 8 6 5]. Switches the processor pin to input	GPIORR	-	0xFF	[0 .. 0xFF]
0x51	R/W	ADC-Channel select. Read command reads the manual switch configuration by changing the direction to input, reading it, and changing back to the defined setting.	ADCCHR	No	0xFF	0..3 0xFF: Release for manual configuration
0x52	R/W	ASC0/ASC1/SPI config. Read command reads the manual switch configuration by changing the direction to input, reading it, and changing back to the defined setting.	ASCSPIR	No	0xFF	0: ASC0→VCP, ASC1→RXTX 1: ASC0→VCP, ASC1→SPI 2: ASC0→RXTX, ASC1→SPI 0xFF: Release for manual configuration
0xB0	W	Set I2CBL bootloader I2C-Address. (default is the same as device I2C-Address)	BLADDR	Yes	Device I2C-Address	1..127
0xBB	W	Enter I2CBL bootloader. Password is the device I2C-Address, XOR-ed with this register address.	BLCMD	No	0x00	BLCMD xor I2CADDR (password) (=0xBB ^ 0x69 = D2)
0xFD	R	Firmware Version	VER	-		[0 .. 0xFF] [MAJ MIN] 4:MSB: MAJ 4:LSB: MIN MAJ: Major version number

						MIN: Minor version number
0xFE	W	Blink n-times	BLINK	Yes	-	[1 .. 255] Number of times to blink
0xFF	R/W	Read Address Register (RAR)	RAR	No	0x00	[0 .. 0xFF] Only valid addresses contain valid values

Example of AT input via the module

AT^SSPI= // Open EHS6 I2C-data connection. Expect CONNECT
<aD21401> // Write “1” in register “GPIO20R” (set GPIO20 to output). Expect {a+}
<bD30001> // Read 1 byte: expect {b+00}, means last command was successful
<cD2FF30> // Set next read address to GPIORR (GPIO direction low byte). Expect {c+}
<dD30001> // Read 1 byte: expect something like {d+1F} (bit 4 is set)
// Close data connection. Expect OK

Result codes

Result	Code	
OK	0x00	Last command was executed successfully
PROTOCOLL_ERROR	0x01	Protocol Error, i.e. wrong number of bytes
ILLEGAL_ADDRESS	0x02	Illegal register address
ILLEGAL_ARGUMENT	0x03	Illegal argument. Argument is out of allowed range.
UNDEFINED	0xFF	

8 New and improved features

8.1 HW Revision 5.1

PCB marking “L30960-N0050-A100”

Table 10: New features HW Rel 5.1

Item	Description
automated switching and configuration	An automated GPIO switching and configuration via I^2C bus was introduced in HW Rel. 5.1. The Concept Board firmware can be updated to get same I^2C functionality as Revision 5.1. The update procedure can be found in the Gemalto Developer Zone.

9 Known issues

9.1 HW Revision 4.0

PCB marking “javakit_a4”

Table 11: known issues HW Release 4

Item	Description	Workaround / Solution
High GPIO current	Increased current up to 80mA / line will be drawn by the GPIO Level shifter / EHS6 when GPIO is configured to wrong direction e.g. GPIO from EHS6 = Out-H ⇔ GPIO from Shield Interface = In.L In that case EHS6 will draw increased current into the level shifter. Level shifters and / or EHS6 might be damaged if that wrong setting persists over a long period of time.	Always be sure that the GPIOs are configured in the right direction. This issue is solved in HW Release 5.1.

9.2 HW Revision 5.1

PCB marking “L30960-N0050-A100”

Table 12: known issues HW Rel 5.1

Item	Description	Solution
Wrong ASC0 state while battery operation	When powering the Concept Board through the battery terminals, the ASC0 TxD and RTS state may change to active. This is caused by the FTDI	Set switch 3 down to configure ASC0 to be connected to the Shield Interface.

	FT232R chipset powering down when the operating voltage is missing. This active state may interfere with EHS6 operation. This does not happen when using battery shields.	
Concept Board fails to switch ON	When the OFF button was pressed shorter than 1s the CB may fail to switch on the next cycle.	Press the OFF button at least 1s to switch off the CB. If the CB already gets in the failed state, press the OFF button >5s to recover normal operation.

10 Related Documents

[1] EHS6 Hardware Interface Description, v02.000a
[2] EHS6 AT Command Set, v02.000a
[3] Getting Started with EHS6, v01
[4] EHSx Java User's Guide, v05
[5] Concept Board Start-up Guide, v01

To visit the Gemalto M2M Website you can use the following link:
http://m2m.gemalto.com/

About Gemalto

Gemalto (Euronext NL0000400653 GTO) is the world leader in digital security with 2011 annual revenues of €2 billion and more than 10,000 employees operating out of 74 offices and 14 Research & Development centers, located in 43 countries.

We are at the heart of the rapidly evolving digital society. Billions of people worldwide increasingly want the freedom to communicate, travel, shop, bank, entertain and work - anytime, everywhere - in ways that are enjoyable and safe. Gemalto delivers on their expanding needs for personal mobile services, payment security, authenticated cloud access, identity and privacy protection, eHealthcare and eGovernment efficiency, convenient ticketing and dependable machine-to-machine (M2M) applications.

Gemalto develops secure embedded software and secure products which we design and personalize. Our platforms and services manage these secure products, the confidential data they contain and the trusted end-user services they enable. Our inovations enable our clients to offer trusted and convenient digital services to billions of individuals.

Gemalto thrives with the growing number of people using its solutions to interact with the digital and wireless world.

For more information please visit
m2m.gemalto.com, www.facebook.com/gemalto, or Follow@gemaltom2m on twitter.

Gemalto M2M GmbH
St.-Martin-Str. 60
81541 Munich
Germany

gemalto
security to be free