CB-2034 User Guide

September 2016

Version 1.1

Copyright Information

Trademark Credits

ARM® is a registered trademark of ARM Limited, UK

Wi-Fi® is a registered trademark of the Wi-Fi Alliance

i.MX6 is a product of Freescale Semiconductor, Inc.

Document Revisions

Date	Version N	Document Changes
09/30/15	0.1	Initial Draft
12/12/15	0.8	First Release
12/14/15	0.9	Error corrections
03/21/16	1.0	Moved vendor specific information to separate AppNote
09/09/16	1.1	Preliminary for HW version 2.5

Table of Contents

1Introduction....6
1.1System Overview....6
1.2Hardware Features....7
1.3Software Features....7
2CB-2034 Specification....8
2.1Power supply....9
2.2LED Indicators....9
2.3Hardware Interfaces....10
2.3.1Interfaces with external control devices....10
2.3.2Interfaces for connecting Devices Under Test....11
2.4Software Interfaces....16
2.4.1CB-2034 System Interface....16
2.4.2Script based Controls....16
2.4.2.1gw_shut_down....17
2.4.2.2gw_set_factory_defaults....17
2.4.2.3gw_cfg_gpio....17
2.4.2.4gw_read_gpio....18
2.4.2.5gw_write_gpio....18
2.4.2.6gw_read_vcc_status....18
2.4.2.7gw_led_port....19
2.4.2.8gw_uart_port....19
2.4.2.9gw_sdio_port....20
2.4.2.10gw_sdio_card_detect....20
2.4.2.11gw_sdio_card_remove....20
2.4.2.12dut_load_drv....21
2.4.2.13dut_unload_drv....21
2.4.2.14dut_config....21
2.4.2.15dut_on....22
2.4.2.16dut_off....22
2.4.2.17dut_reset....22
3CB-2034 Setup....24
3.1Simple setup for manual DUT operation....24
3.2Manufacturing setup for automated DUT operation....25

4Quick start guide for CB-2034 operation....26
4.1Initial setup....26
4.2Configuration of IP network parameters....26
4.3Controlling the DUT....26
4.3.1Manual control via a terminal program....26
4.3.2Automated, remote control via SSH....27
4.4Updating files on CB-2034....28

1 Introduction

This document provides a detailed description of the CB-2034 and all aspects of its use as host platform for testing wireless devices. It is intended as product manual for development, test and manufacturing engineers and technical staff working with this product.

1.1 System Overview

The CB-2034 is a Linux based host platform for wireless Devices Under Test (DUT). It integrates device drivers for the tested devices and provides mechanisms to control the DUT directly, respectively for sending vendor specific DUT commands to the appropriate device drivers.

CB-2034 can be easily controlled and configured via a serial interface, or, a Telnet/SSH connection over the network interface. The user has access to a Linux command shell, enabling low level access to all control box features. DUT functions can be controlled script based via the command shell. The following script based functions are available to remote control the operation of CB-2034, respectively the DUT:

- switching CB-2034 off
- switching CB-2034 control LEDs on/off
- configure, read or write GPIO signals provided at the DUT interface connector
- switching DUT power on/off
- activating/deactivating digital DUT control & input signals
- loading/unloading of DUT device drivers
- transmitting command sequences to the DUT

In addition, CB-2034 integrates an application for transparent bridging of DUT commands between its network interface and DUT device drivers. This function enables direct DUT remote control from an external, DUT vendor specific software application.

DUT specific test & manufacturing software is also available from several RF test equipment vendors. CB-2034 enables integration of such dedicated test equipment in the manufacturing line as shown in this user guide.

•

1.2 Hardware Features

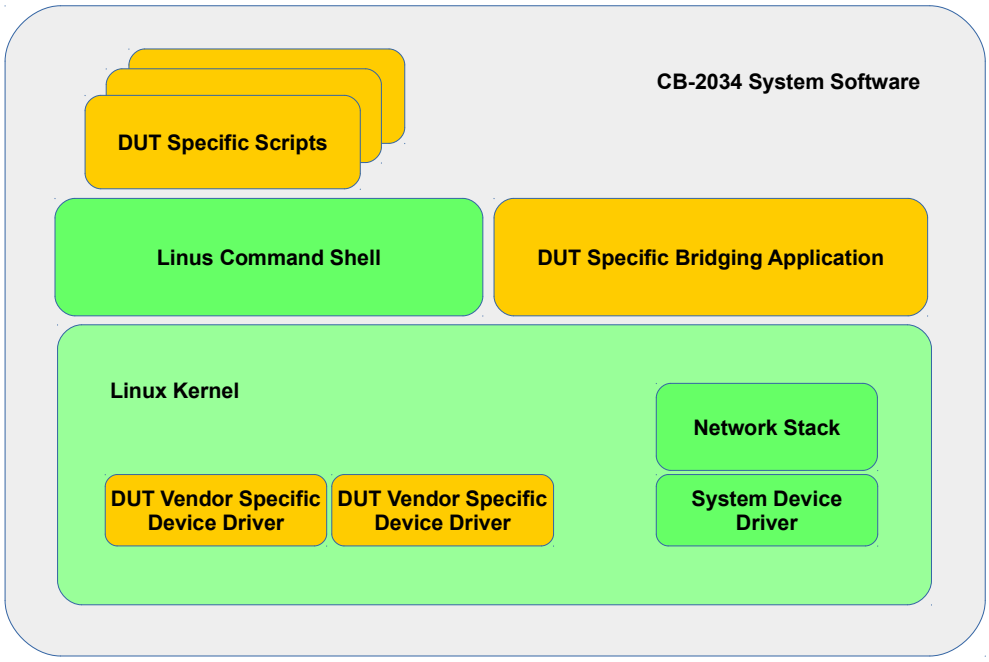
CB-2034 is an ARM based CPU host platform for DUTs, which has specifically been developed for use in production line test and manufacturing environments, as well as under laboratory conditions. The system is build into a ruggedized chassis and supports a wide range of ambient temperatures. That makes it suitable for operation in harsh conditions like manufacturing, screening or burn-in of wireless devices.

The system is powered by an external power supply, which is connected via a screw locking power inlet.

1.3 Software Features

As mentioned before, CB-2034 is a Linux based platform, running with kernel version 3.16. It integrates vendor specific Linux drivers for the DUTs, as well as control functions. Available features can be listed as:

- Linux 3.16 operating system
- integrated vendor specific DUT device drivers
- functions for CB-2034 remote control
  - script based via UART or Telnet/SSH session
- DUT remote control
  - script based via UART or Telnet/SSH session
  - by bridging vendor specific test commands between network interface and DUT device drivers



2 CB-2034 Specification

CB-2034 is a stand-alone Linux platform, specifically designed as host system for testing wireless devices. The system is based on a Freescale i.MX6 CPU. It is build into a small form factor, ruggedized case, which features power supply, LAN and UART connectors at the front side. A 50 pin flat cable connector with all signals for DUT connectivity is provided at the rear side. For signaling system states, respectively the state of DUT interface signals, 3 rows a 10 LEDs are shown at the top side of the case.

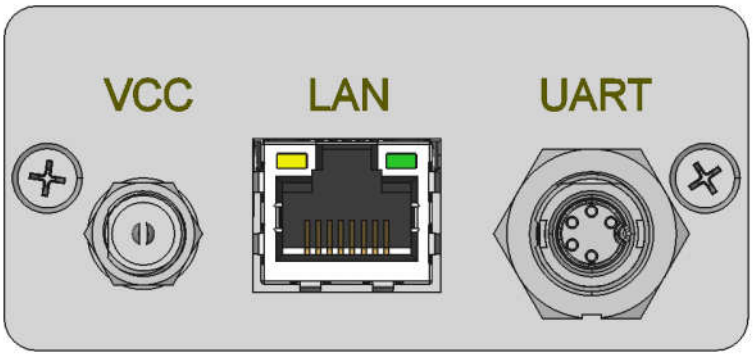


Figure 1: Front side view and connectors

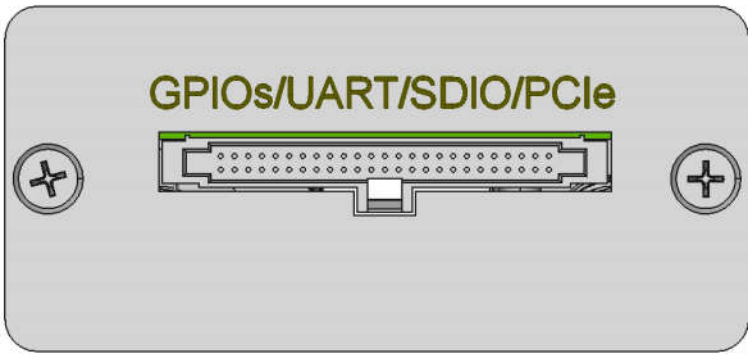


Figure 2: Rear side view with DUT connector

2.1 Power supply

The power supply of CB-2034 is via an external 3,2-5,1V power supply. The power inlet has an inverse-polarity, under/over voltage and overload protections. Physically, the external power supply is connected via a Switchkraft S761K screw locking connector. The polarity of the connector is as shown below.

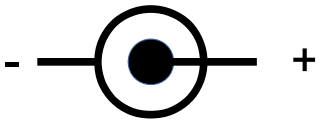


Figure 3: Polarity of the power supply connector

2.2 LED Indicators

CB-2034 features 10 LEDs to indicate different device states, and further 20 LEDs that show the states of the GPIO lines available for DUT connection. LEDs showing GPIO status can be switched on or off all together via a dedicated script command.

LED assignment is shown in the table below:

LED	Description
System Power	Indicates whether CB-2034 is on or off
UART_RX	Receive activity at the host UART interface
UART_TX	Transmit activity at the host UART interface
TEST/ERR	Test is running, or, internal error condition has occurred
VCC_GROUP_ST1…6	State of external power supply groups 1… 6
LED0 … 15	State of the DUT control signals DATA0 to DATA15.
DUT UART_RX	Receive activity at the DUT UART interface
DUT UART_TX	Transmit activity at the DUT UART interface
DUT UART_RTS	Status of DUT UART interface RTS signal line
DUT UART_CTS	Status of DUT UART interface CTS signal line

2.3 Hardware Interfaces

The following sections describe the interfaces of CB-2034 accessible to the user. These interfaces are used to connect the product to external control devices, like computers or computer networks, or, to connect a Device Under test (DUT) to the CB-2034. Interfaces to external control devices are provided at the front side of the system case, while all DUT connections are made via the rear side connector.

2.3.1 Interfaces with external control devices

For connections with external devices the control box provides a 10/100 Mbps Ethernet LAN interface. Alternatively, a UART serial interface supporting a maximum baud rate of 921600 bps can be used as well.

Note
The Ethernet port, is not ESD protected!

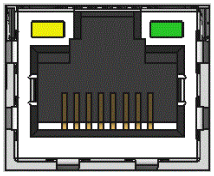


Figure 4: Ethernet LAN connector

The Ethernet connector of CB-2034 is an RJ-45 connector with two integrated LEDs, which show the status of the Ethernet connection with an external device or network.

LED	State	Description
yellow	off on	no physical link available (cable not plugged in) link detected
green	off on blinking	10 Mbps link speed 100 Mbps link speed data transfer in progress

The system provides a connector for the serial interface, which can be used with a Switchkraft EN2C5M26G1 plug. Signal assignments of the connector are shown below.

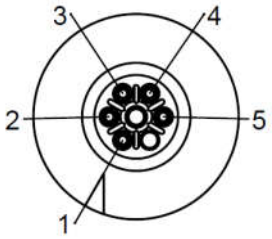


Figure 5: UART connector

Pin Number	Signal Name	Description
1	UART_CTS	UART Clear-to-send signal
2	UART_TX	UART transmit signal
3	UART_RTS	UART Request-to-send signal
4	UART_RX	UART receive signal
5	GND	Signal ground

2.3.2 Interfaces for connecting Devices Under Test

As a DUT host the CB-2034 is supporting different host interface options:

- SDIO3.0
- PCIexpress
- UART

To control other specific digital hardware signals of the DUT, like RESET, Power Down or Wake Up signals, the CB-2034 supports several GPIO signals. The use of this GPIO signals can be defined by the user according to the needs of the DUT.

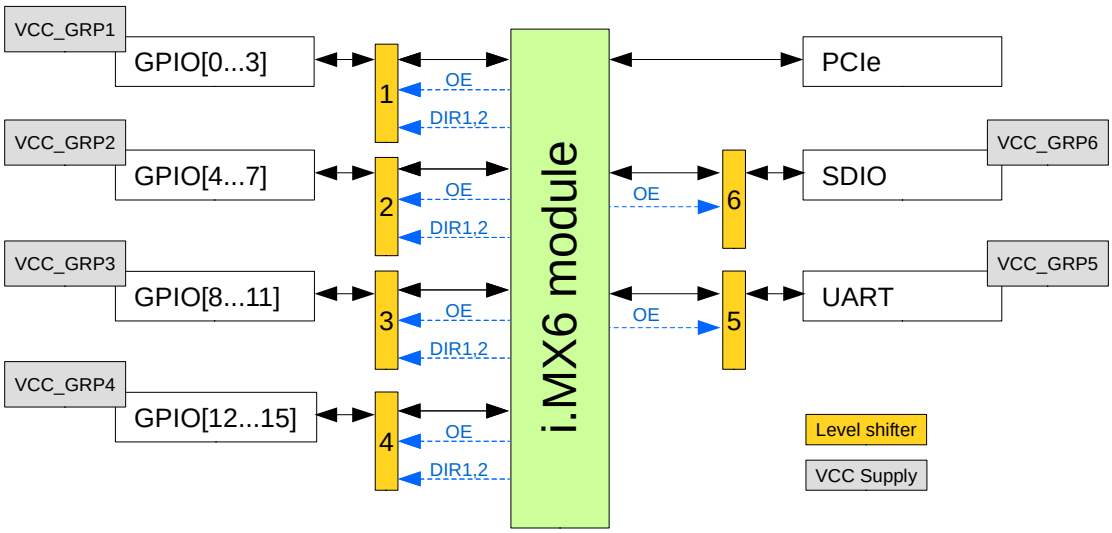


Figure 6: Block diagram of DUT signal interfaces with host CPU

Note
UART and SDIO are connected through level shifters, which allow to select the IO voltage level of the DUT. PCIe is connected directly. That exposes this interface to the risk of ESD damage!

The system provides a 50 pin, dual row flat cable connector for DUT connection (TE 5-104892-5). It is used to implement a short interconnect with the DUT test fixture. All host interfaces, power supplies and control signals are exposed at that connector. The pin-out of this connector is given in the table below.

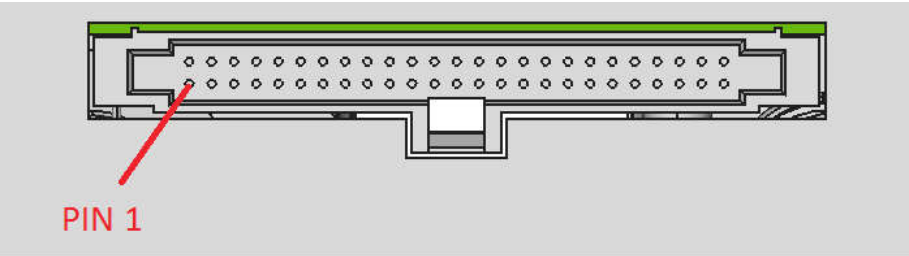


Figure 7: 50 pin DUT connector

The color coding shows how the different signal groups are supplied by external power supplies. DATA[0..15] are general GPIOs, which can be defined as either input or output signals to control digital control signals of the DUT.

Function	Pin	Pin	Function
GND	1	2	+3V3
PCI_RXP	3	4	DATA3
PCI_RXN	5	6	DATA2
GND	7	8	DATA1
PCI_CLKN	9	10	DATA0
PCI_CLKP	11	12	GND
GND	13	14	VCC_GROUP1
PCI_TXP	15	16	DATA4
PCI_TXN	17	18	DATA5
GND	19	20	DATA6
NC	21	22	DATA7
SDIO DATA2	23	24	GND
SDIO DATA3	25	26	VCC_GROUP2
SDIO CMD	27	28	DATA8
GND	29	30	DATA9
VCC GROUP6	31	32	DATA10
SDIO CLK	33	34	DATA11
SDIO DATA0	35	36	GND
SDIO DATA1	37	38	VCC_GROUP3
VCC_GROUP5	39	40	DATA12
GND	41	42	DATA13
UART TX	43	44	DATA14
UART CTS	45	46	DATA15
UART RX	47	48	GND
UART RTS	49	50	VCC_GROUP4

Figure 8: DUT connector pin assignments (connector front view)

The table below provides a more detailed description of each connector pin and shows, how the digital input/outputs are controlled via the CB-2034 level shifter logic. Level shifters can control the direction of signal pairs only, output enable is applied on groups of 4 signals. Refer figure 6 above for the assignments of level shifters versus DUT connector signals.

DATA[0..15] can be configured as either input, or output. Signal direction can be defined in pairs only (e.g. DATA0 and DATA1) due to the operation of the level shifters. The level shifter output enable signal is applicable to all DATA[n] signals belonging to a VCC_GROUP.

For the SDIO signals, as well as for the DUT UART signals only output enable can be controlled. The direction of these interface signals is fixed by the CB-2034 implementation.

Pin Number	Signal Name	VCC Group	LS DIR	LS OE	Description
1	GND	-	-	-	Signal ground
2	+3,3 V	-	-	-	Auxiliary supply, output max 400 mA
3	PCIE_RXP	-	-	-	PCIe RX differential signal pair 1, +
4	DATA_3	1	1.2	1	GPIO[3]
5	PCIE_RXN	-	-	-	PCIe RX differential signal pair 1, -
6	DATA_2	1	1.2	1	GPIO[2]
7	GND	-	-	-	Signal ground
8	DATA_1	1	1.1	1	GPIO[1]
9	PCIE_CLKN	-	-	-	PCIe clock differential signal pair, -
10	DATA_0	1	1.1	1	GPIO[0]
11	PCIE_CLKP	-	-	-	PCIe clock differential signal pair, +
12	GND	-	-	-	Signal ground
13	GND	-	-	-	Signal ground
14	VCC_GRP1	1	-	-	External VIO supply 1
15	PCIE_TXP	-	-	-	PCIe TX differential signal pair 1, +
16	DATA_4	2	2.1	2	GPIO[4]
17	PCIE_TXN	-	-	-	PCIe TX differential signal pair 1, -
18	DATA_5	2	2.1	2	GPIO[5]
19	GND	-	-	-	Signal ground
20	DATA_6	2	2.2	2	GPIO[6]
21	-	-	-	-	reserved
22	DATA_7	2	2.2	2	GPIO[7]
23	SDIO_DATA2	6	-	-	SDIO data signal 2
24	GND	-	-	-	Signal ground
25	SDIO_DATA3	6	-	-	SDIO data signal 3
26	VCC_GRP2	2	-	-	External VIO supply 2

27	SDIO_CMD	6	-	-	SDIO command signal
28	DATA_8	3	3.1	3	GPIO[8]
29	GND	-	-	-	Signal ground
30	DATA_9	3	3.1	3	GPIO[9]
31	VCC_GRP6	6	-	-	External VIO supply 6, VCC_SDIO
32	DATA_10	3	3.2	3	GPIO[10]
33	SDIO_CLK	6	-	-	SDIO clock signal
34	DATA_11	3	3.2	3	GPIO[11]
35	SDIO_DATA0	6	-	-	SDIO data signal 0
36	GND	-	-	-	Signal ground
37	SDIO_DATA3	6	-	-	SDIO data signal 3
38	VCC_GRP3	3	-	-	External VCC supply 3
39	VCC_GRP5	5	-	-	External VIO supply 5
40	DATA_12	4	4.1	4	GPIO[12]
41	GND	-	-	-	Signal ground
42	DATA_13	4	4.1	4	GPIO[13]
43	UART_TX	5	5.1	5	UART_TX_DATA (output)
44	DATA_14	4	4.2	4	GPIO[14]
45	UART_CTS	5	5.1	5	UART_CTS (output)
46	DATA_15	4	4.2	4	GPIO[15]
47	UART_RX	5	5.2	5	UART_RX_DATA (input)
48	GND	-	-	-	Signal ground
49	UART_RTS	5	5.2	5	UART_RTS (input)
50	VCC_GRP4	4	-	-	External VIO supply 4

2.4 Software Interfaces

Main interface of the CB-2034 is the Linux command shell accessible via Telnet/SSH. Users have full access to the system via this interface. Shell access can be achieved via the Ethernet LAN interface, or, via the UART interface of the device.

2.4.1 CB-2034 System Interface

The CB-2034 Ethernet LAN interface is configured in auto-detect 10/100 Mbps mode. When connecting it to an external network, the interface will auto-detect the proper interface speed and connect to the network accordingly. By default, the system is configured with a static IP address of 192.168.178.99 and will use it as long as the user does not configure it differently. Use of DHCP to obtain an IP address automatically from the network can be configured as well. However, for the application in a manufacturing environment dedicated, fixed IP addresses may be of advantage.

The IP address CB-2034 comes up with can be set in a configuration file called ***initETH.sh***, which is located in the /media/fs/autorun directory.

As an alternative to access the system Linux command shell a UART interface can be used as well. The default configuration of the UART interface out-of the-box is 115200 baud, 8 bits, no parity and one stop bit (8N1). Similar to the LAN interface the user may reconfigure these settings if needed.

2.4.2 Script based Controls

On system shell level several predefined scripts are available to control functions of the CB-2034 and/or the DUT. The following sections describe all script commands. Generic, DUT independent script commands and executables are located in /media/fs/bin directory, while DUT specific scripts are stored under /media/fs/<vendor>/<device>/scripts.

The /media/fs/bin path is added to the PATH system variable. That allows to call generic scripts like system commands directly from any location. DUT specific commands are called via symbolic links in the /sbin directory. These symbolic links are created as function of the selected DUT type during initialization of CB-2034, when ***jobs.sh*** located in /media/fs/autorun is executed.

2.4.2.1 gw_shut_down

This command is used to switch off CB-2034. Before the system is shut down the DUT will be switched off and all DUT controls are disabled.

#gw_shut_down

Parameters

none

Returns

n/a

2.4.2.2 gw_set_factory_defaults

This command is typically called once, right after the system has been started and Linux is booted. It assures proper initialization of all DUT interface signals. Host interfaces to the DUT are initialized and disabled. GPIO signals DATA[0...15] at the DUT interface connector are all set into input mode.

#gw_set_factory_defaults

Parameters

none

Returns

n/a

2.4.2.3 gw_cfg_gpio

This command can be called if GPIO interface lines have to be configured for a specific use with the DUT as either input or output. The appropriate level shifter controlling the affected GPIO pins will be re-configured as well. Since the GPIO level shifters have explicit controls for the signal direction (A ⟶ B, or, B ⟶ A), and those controls are applied pairwise on level shifter signals (e.g. DATA[0] and DATA[1], DATA[2] and DATA[3], etc.), the command is always configuring pairs of signals. If a signal pair is configured as output, the appropriate output data value can be specified optionally. Further, there is an optional parameter, specifying whether the output shall be enabled immediately, or, whether the appropriate level shifter enable signal shall not be changed.

#gw_cfg_gpio data_line direction [value] [enable]

Parameters

data_line number of the upper or lower signal of a GPIO data signal pair (0...15)

direction	direction to be configured, 'in' or 'out'
value	optional output value to be configured on the signal pair (0...3)
enable	optional level shifter output enable control: 0 - don't enable level shifter (default), 1 - enable it after setting output values

Returns

error code

2.4.2.4 gw_read_gpio

This command allows to read a specific GPIO data signal pin, if the pin is configured as input and the level shifter is configured as input and enabled. Otherwise an error code will be returned.

#gw_read_gpio data_line

Parameters

data_line	number of the GPIO data signal to be read (0...15)

Returns

result	signal state 0 or 1, or error code

2.4.2.5 gw_write_gpio

With this command a specific GPIO pin can be set, provided the pin is configured as output and the level shifter is configured as output, too. After setting the specified value to the GPIO pin, the level shifter output enable pin will be set.

#gw_write_gpio data_line value

Parameters

data_line	number of the GPIO data signal to be read (0...15)
value	signal value to be written (0 or 1)

Returns

error code

2.4.2.6 gw_read_vcc_status

This command can be used, if the status of a specific VCC input line, or, all VCC lines shall be checked. A return value of 1 means the an external supply voltage is provided for the specified VCC group, while a status of 0 means there is no VCC supply voltage present for that group.

#gw_read_vcc_status [vcc_line]

Parameters

vcc_line	optional parameter indicating VCC line or VCC group, for which the status is checked (1...6); if this parameter is not specified, a bit mask with the status of all VCC inputs will be returned

Returns

result	VCC status (0 or 1), or error code

2.4.2.7 gw_led_port

This command enables or disables the 20 LEDs showing the signal status of the GPIO lines DATA[0...19].

#gw_led_port status

Parameters

status	port status to be set, 'enable' or 'disable'

Returns

error code

2.4.2.8 gw_uart_port

With this command the DUT UART port can be enabled, or, disabled.

#gw_uart_port status

Parameters

status	port status to be set, 'enable' or 'disable'

Returns

error code

2.4.2.9 gw_sdio_port

This command enables or disables the SDIO port used to connect the DUT.

#gw_sdio_port status

Parameters

status port status to be set, 'enable' or 'disable'

Returns

error code

2.4.2.10 gw_sdio_card_detect

This command asserts the SDIO card detect signal, triggering the Linux mmc driver to initiate a card (DUT device) detection cycle. It must be called, if the DUT is connected via SDIO and the OS shall be made aware of it's presence. By calling this command SDIO hot plug functionality will be activated.

#gw_sdio_card_detect

Parameters

none

Returns

none

2.4.2.11 gw_sdio_card_remove

This command de-asserts the SDIO card detect signal. The Linux mmc driver is detecting a card (DUT device) removal and will deactivate all device SDIO functions.

#gw_sdio_card_remove

Parameters

none

Returns

none

2.4.2.12 dut_load_drv

With this command all required DUT driver modules are loaded into the kernel. The DUT does not need to be present if this is done.

#dut_load_drv [no-mfg]

Parameters

no-mfg	optional parameter to start the device in normal operation mode; without this parameter the DUT is started and initialized in test mode

Returns

none

2.4.2.13 dut_unload_drv

This function allows to unload all driver modules, e.g. to reload the driver with additional parameters for debugging or other DUT specific configurations.

#dut_unload_drv

Parameters

none

Returns

none

2.4.2.14 dut_config

In order to configure the CB-2034 for a specific DUT this function has to be called once, before each other DUT specific function can be executed. Note that for each DUT type that is operated via the CB-2034, individual dut_xxx functions are implemented.

#dut_config

Parameters

none

Returns

none

2.4.2.15 dut_on

With this function the DUT will be enabled for testing. It enables the DUT specific host interfaces of CB-2034, switches on DUT power supplies and activates the SDIO card detect signal to start the DUT driver if required. Once the devices are up and running a DUT specific bridge application is started, which enables exchange of DUT test command and response frames with an external PC.

#dut_on

Parameters

none

Returns

none

2.4.2.16 dut_off

The DUT is switched off by this function. This is done by disconnecting all DUT power supplies and disabling the host interfaces. After switching off DUT, it is disconnected and isolated from the test fixture. Changing DUTs is possible without the risk of electrical damage of the tested devices. The CB-2034 can be left powered and fully functional during this state.

#dut_off

Parameters

none

Returns

none

2.4.2.17 dut_reset

The DUT reset function is called to carry out a full power cycle of the DUT, while explicitly asserting the DUT reset signal.

Note that this function has to be used with care, since it does not take care of the device driver states. In some cases it might therefore be more appropriate to call the dut_off / dut_on commands instead to make sure device drivers are in proper state after the DUT reset.

#dut_reset

Parameters

none

Returns

none

3 CB-2034 Setup

For operating the DUT under control of a vendor specific SW application, or the application provided by a wireless test system equipment vendor, CB-2034 has to be interconnected with external devices as shown below.

3.1 Simple setup for manual DUT operation

The following setup can be used under laboratory conditions, if DUT and measurement equipment are operated manually. CB-2034 is connected to a PC running vendor specific test software, enabling manual DUT control. Host interfaces of the DUT are connected to the CB-2034, while its RF ports are accessible to various RF test equipment.

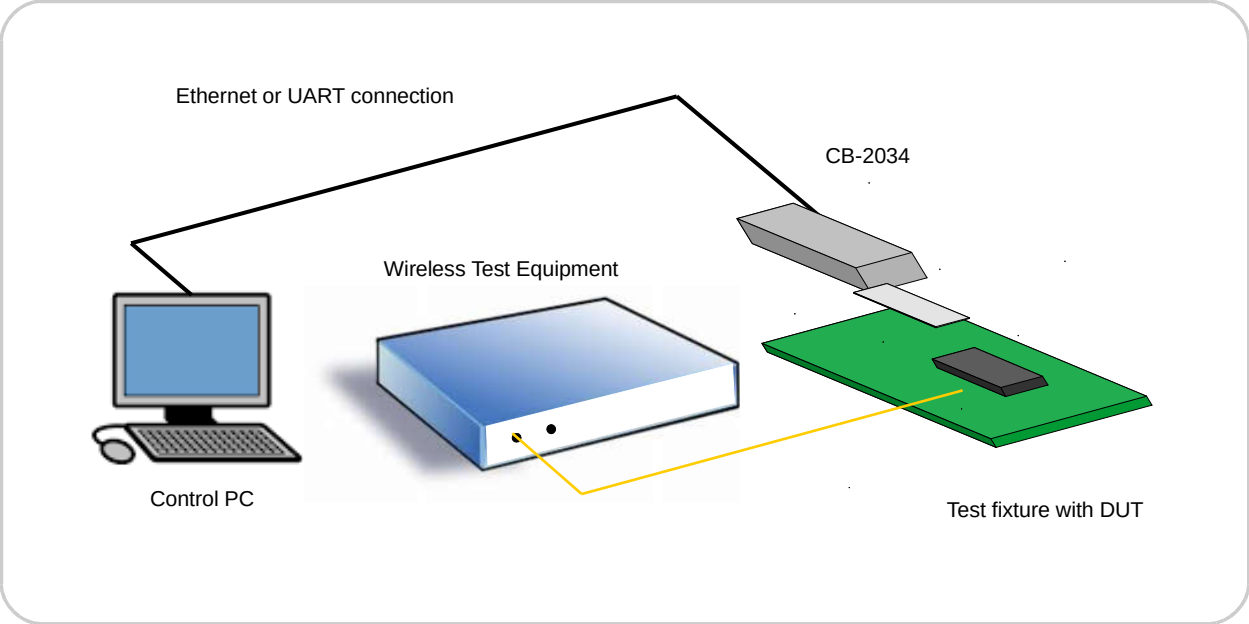


Figure 9: Setup for manual tests with CB-2034

3.2 Manufacturing setup for automated DUT operation

Under automated operating conditions, CB-2034, the DUT connected to it, as well as the RF test equipment are controlled by a centralized software function running on an external PC. Manual interaction is minimized, typically reduced to changing DUTs (connected with CB-2034 via a dedicated test fixture) and monitoring test results displayed by the software.

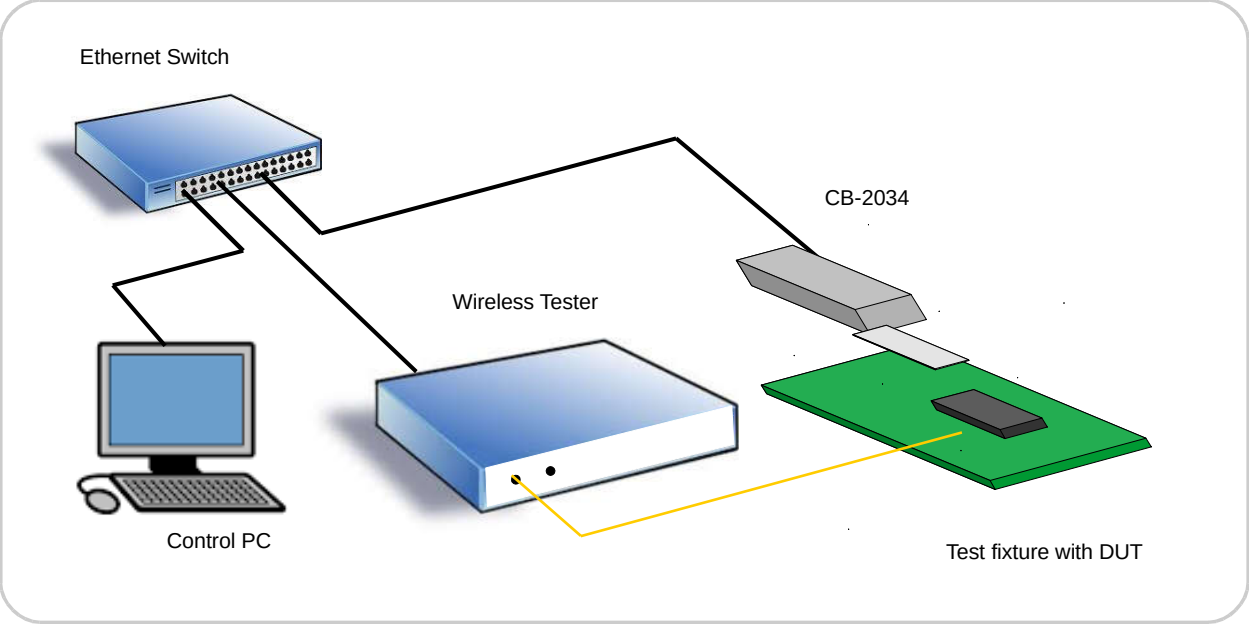


Figure 10: Setup for automated tests with CB-2034

4 Quick start guide for CB-2034 operation

4.1 Initial setup

For the first setup of CB-2034 it is suggested to connect a serial cable to an external control PC and to use a terminal program to access and configure the device. The following steps are required:

- connect CB-2034 to an external power supply
- connect the serial cable to the control PC and start a terminal program with serial port parameters 115200 baud, 8 bit, no parity, one stop bit (8N1)
- switch on CB-2034 power supply, the terminal window should show the log messages from booting up the Linux kernel
- at the log-in prompt enter username=root, password=root to get access to the Linux command shell of CB-2034

Now you are ready to check the system configuration and to make modifications to it if necessary.

4.2 Configuration of IP network parameters

In order to run CB-2034 in an IP network, the device IP address must be configured accordingly. By default the system is configured with a static IP address of **192.168.178.99**. The Ethernet port eth0 is configured with this address during the initialization of the device right after starting it up. If necessary, the used IP address can be reconfigured by changing the appropriate entry in **/media/fs/autorun/initETH.sh**.

4.3 Controlling the DUT

Once the device is up and running DUT commands can be sent manually by a terminal program connected either via UART, or TCP/IP. For automated control of CB-2034, it is suggested to use a secure remote connection via SSH.

4.3.1 Manual control via a terminal program

Via a terminal program CB-2034 control commands and DUT control commands as described in section 2.4.2 can be entered after successful log-in. In addition, all Linux command shell (ash) commands can be executed as well.

After starting up CB-2034 the following DUT control commands must be entered:

- **dut_config**, for basic configuration of CB-2034 for operating a specific DUT

- **dut_on**, to switch on the DUT for operation
- **dut_off**, for switching off the DUT

4.3.2 Automated, remote control via SSH

For controlling the DUT (respectively CB-2034) via SSH it is recommended to install an SSH client software on the controlling PC. OpenSSH is a good candidate for control PC running Windows operating system. It also includes an SCP client, which can be used to read/write files from/to CB-2034 /media/fs file system.

The command to be called on the control PC for running a script or command on the remote CB-2034 device is the following:

```
ssh root@<CB-2034_IP_address> "source /etc/profile; <command>"
```

<CB-2034_IP_address> must be replaced with the IP address of CB-2034 used in your setup, and <command> is the particular script or Linux command that is to be executed on it. Executing the above SSH command will prompt for the CB-2034 password. This is a problem, if an automated control software running on the PC needs to execute commands on CB-2034 as part of the manufacturing test sequence. In such an environment, manual entry of passwords is not possible. An easy, yet extremely secure alternative for entering a password is using SSH with authentication keys.

SSH key pairs are two cryptographically secure keys that can be used to authenticate a client to an SSH server. Each key pair consists of a public key and a private key. The private key is retained by the client and should be kept absolutely secret. The public key can be used to encrypt messages that only the private key can decrypt. This property is employed as a way of authenticating using the key pair.

The following section describes the steps to generate an SSH key pair on your control PC and to install your public key on CB-2034. The OpenSSH software package provides all necessary tools for this operation.

1. On he control PC open a Windows command shell and enter the following command to generate a private and public key pair for being used with SSH.

   ```
   ssh-keygen -t rsa
   ```

2. The program will ask for a target file location and name to store the keys. Just press enter to use the default name. Further it will ask for a passphrase, which can be used to encrypt the stored key files. We don't use a passphrase, therefore just hit enter.

3. The keys are now generated in the C:\Users\<user_name>\.ssh directory, with the private key stored in a file called ***id_rsa***, the public key stored in ***id_rsa.pub***.

4. For SSH to work properly with the key files specific access rights need to be set to the private key file. Appropriate attributes cannot be set with the normal Windows file tools. Therefore the following command must be executed at the Windows command shell prompt, after changing into the C:\Users\<user_name>\.ssh directory:

   ```
   chmod -vR 600 id_rsa
   ```

5. Now the public key needs to be uploaded to CB-2034. For that, the key is added to a special file within the user account (root) you will be logging into called ***authorized_keys***. This file is located in the /root/.ssh directory. A template of this file is copied from /media/fs/authorized_keys to the /root/.ssh directory during system start-up. To make your changes to ***authorized_keys*** persistent over CB-2034 reboots, please enter the key in the appropriate file in /media/fs, as well as in the currently used copy in /root/.ssh.

After setting up the authentication keys on your control PC and CB-2034, the above mentioned SSH command is executed without prompting for a password anymore.

4.4 Updating files on CB-2034

The kernel file system of CB-2034 is read-only. A working copy of it is stored in a SRAM, which makes it read/write during operation. All changes to it are temporary only, i.e. will be lost when the system is rebooted or powered off.

Permanent file storage is provided in the /media/fs file system only. All permanent changes to the files stored on CB-2034 should therefore be made on this file system.

For uploading/downloading files to/from /media/fs an SCP client installed with the OpenSSH package can be used. The command to store files on CB-2034 is

```
scp <source_file> root@<CB-2034_IP_address>:/media/fs/<target_file>
```

While copying files from CB-2034 to the control PC can be done with

```
scp root@<CB-2034_IP_address>:/media/fs/<source_file> <target_file>
```