

LTE eNodeB

AZCMCK2KBXB1 Baseband Unit BBU Platform SW Installation Manual

Document Revision: 1.2
Date: June, 13th 2019

Copyright © Azcom Technology srl 2019. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Azcom Technology srl.

Trademarks and Permissions



and other Azcom trademarks are trademarks of Azcom Technology srl.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Azcom and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute the warranty of any kind, express or implied.

Azcom Technology srl

Headquarter:

Centro Direzionale Milanofiori

Strada 6, Palazzo N/2

20089 Rozzano (MI) – Italy

Tel.: +39 02 82450311

Website: www.azcom.it

E-mail: info@azcom.it

REVISION HISTORY

Rev.	Date	Description
1.0	25-Mar-2016	Changed template
1.1	29-Jun-2016	New platform 2.0
1.2	13-Jun-2019	Chapter 5 added, describing recovery system upgrade procedure. Chapter 6 added, describing new automatic exit from recovery feature.

ABBREVIATIONS AND ACRONYMS

ARM	Advanced RISC Machine
BBU	Base Band Unit
CCS	Code Composer Studio
ETH	Ethernet
EVM	Evaluation Module
FPGA	Field Programmable Gate Array
PC	Personal Computer
SSH	Secure Shell
SW	Software
TFTP	Trivial File Transfer Protocol
UBIFS	Unsorted Block Image File System
USB	Universal Serial Bus

SUMMARY

1 Introduction	6
2 Platform distribution package content	7
3 Host configuration.....	8
3.1 Serial port configuration	8
3.2 Network configuration.....	8
3.3 File transfer	8
4 Update of BBU Platform Software.....	9
4.1 Platform command.....	9
4.2 Platform Information	10
4.3 Platform Status	10
4.4 Platform Type.....	11
5 Update of BBU Recovery Platform Software	12
5.1 Recovery command.....	12
5.2 Recovery Information	13
6 Automatic Reboot from Recovery Mode.....	14
6.1 Stop Automatic Reboot.....	14
7 Update device to New Filesystem Layout (Platform 2.0)	15
7.1 Setup	15
7.1.1 Hardware.....	15
7.1.2 Software.....	15
7.1.3 Configuration	16
7.2 Procedure.....	17
8 Dip-switch configuration	25
9 Update PIC firmware	26

1 Introduction

This document describes the procedure of Platform SW installation for the following boards:

- Azcom Kepler BBU (AZCMCK2KBXB1), hereinafter **<soc>** substituted with **bb-kepler**
- Texas Instruments EVM-Kepler, hereinafter **<soc>** substituted with **evmk2k**

2 Platform distribution package content

Distribution Package is organized as follow:

platform_<version>.tar	
platform_<version>.txt	Release manifest
images/bootcore-bb-kepler.img	Bootcore Image
images/u-boot-bb-kepler.img	Operative Bootloader
images/u-boot-recovery-bb-kepler.img	Recovery Bootloader
images/boot-bb-kepler.azubi	UBIFS Main boot partition (Device Tree, Kernel, RootFS)
images/pers-bb-kepler.azubi	UBIFS persistent partition (Application packages, configuration)
images/drec-bb-kepler.azubi	UBIFS Disaster Recovery partition(Device Tree, Recovery kernel, Recovery RootFS)
packages/platform-tws8300bbu_2.2.3.azpkg	Platform package (Bootloader, FPGA bitstream, Device Tree, Kernel, RootFS)
packages/recovery-tws8300bbu_2.2.3.azpkg	Recovery package (Bootloader, FPGA bitstream, Device Tree, Recovery Kernel, Recovery Rootfs)

3 Host configuration

Windows 7 (or higher) PC or Ubuntu 16.04 (or higher) is required.

3.1 Serial port configuration

The following parameters are needed to connect to the serial console:

- Speed (baud rate): 115200
- Bits: 8
- Parity: None
- Stop Bits: 1
- Flow Control: None

3.2 Network configuration

To execute the update procedure is necessary that Host Computer and Kepler BBU Ethernet port are reachable via IP protocol.

For example, we describe a direct connection between Host Computer and Kepler BBU:

- Kepler BBU IP address: 192.168.1.1
- Host IP address: 192.168.1.254
- Network netmask 255.255.255.0

Connect the Ethernet cable to **ETH RJ-45** connector.

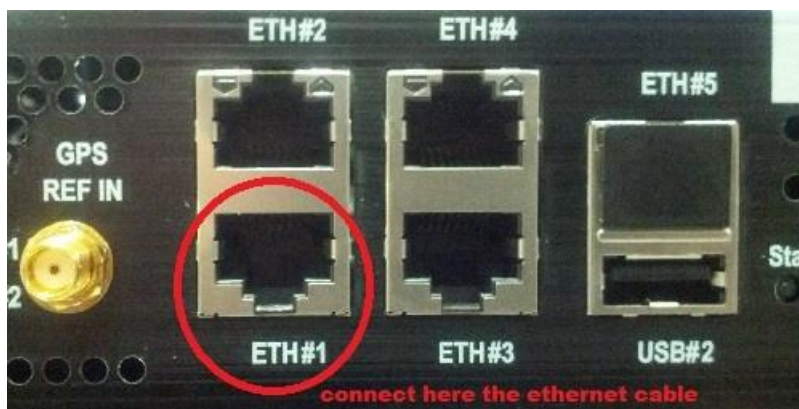


Figura 1 Kepler BBU ETH port

3.3 File transfer

The preferred method to transport packages files to Kepler BBU is Secure Copy protocol (scp). This copy protocol is native on Linux PC and is available on Windows PC via WinSCP application, freely downloadable at <https://winscp.net/eng/index.php>

4 Update of BBU Platform Software

From Platform 2.0.0 and successive there is a new method to upgrade whole system without affecting applications (eNB and EPC) and configurations.

The system is now composed by different packages:

- Platform
- eNB
- EPC

There is also a separate layer to make local configuration persistent between upgrades.

These packages are updatable independently and are decoupled each other: this means that updating platform packages do not affect other packages and configuration.

4.1 Platform command

Platform package deployment is executed in a two-step procedure using 'platform' command after a valid package is uploaded in machine /tmp directory.

These are the steps to complete the platform update operation:

1. Upload a valid platform package in product /tmp directory via scp command or WinSCP application
2. From Kepler BBU command line execute 'platform retrieve' command.

This should be the console output:

```
$ platform retrieve
Serch for platform [PRESENT]
Clean deploy directory [DONE]
Get configuration file [DONE]
Parse OS release [DONE]
Check tws8300bbu compatibility [VALID]
Parse OS release [DONE]
Check platform version [2.0.0]
Deploying skern [DONE]
Deploying kernel [DONE]
Deploying devicetree [DONE]
Deploying initrd [DONE]
Deploying rootfs [DONE]
Deploying fpga [DONE]
Deploying uboot [DONE]
Deploying configuration [DONE]
Deploying configuration file [DONE]
Sync filesystem [DONE]
Mark retrieve as ready to upgrade [DONE]

Platform is now ready to be upgraded
```

3. At this point is possible to upgrade the system platform issuing 'platform upgrade' command.

This is the expected output if no u-boot or no FPGA needs update:

```
$ platform upgrade

Check for platform status      [VALID]
Verifying skern               [DONE]
Verifying kernel              [DONE]
Verifying devicetree           [DONE]
Verifying initrd               [DONE]
Verifying rootfs              [DONE]
Verifying fpga                 [DONE]
Verifying uboot                [DONE]
Verifying configuration        [DONE]
Check u-boot version           [SAME VERSION]
Check fpga version             [SAME VERSION]
Mark system to be upgraded     [DONE]

Platform is now rebooted to be upgraded
```

4. Now the machine will reboot and initial ram-disk will complete the procedure. This means that a second reboot after initial ram-disk update is expected, but there is no need for any user interaction.

After second reboot the machine is updated as expected.

In case also FPGA needs to be updated 'platform upgrade' command will prompt to power cycle the board.

4.2 Platform Information

From Platform 2.0.0 and successive there is a method to get information about the current platform installed. Executing the command '*platform info*' will report this information.

```
$ platform info
Codename:          tws8300bbu
Version:           2.2.3
Build:             6
Platform FPGA:     tws8300bbu-001c
Actual FPGA:       v001c
Bootloader:        2013.01.14.r51
```

4.3 Platform Status

From Platform 2.2.3 there is a method to know if the current system is in Normal Mode or Recovery Mode. Executing the command '*platform status*' you will get the following information

- If executed in Normal Mode:

```
$ platform status
Type:              main
```

- If executed in Recovery Mode:

```
$ platform status
Type: recovery
```

4.4 Platform Type

From Platform 2.2.3 there is a method to switch from main and recovery system with a user friendly interface.

The Normal Mode System can switch to Recovery System executing:

```
$ platform type recovery
Set Next Boot on Recovery System: [DONE]
$ reboot

The system is going down for reboot NOW!
```

The Recovery System can switch to Normal Mode System executing:

```
$ platform type main
Setting boot as safe [DONE]
root@AZB_NCOM_4300:~# reboot

The system is going down for reboot NOW!
```

5 Update of BBU Recovery Platform Software

From Platform 2.2.3 there is a procedure to upgrade the recovery system.

Warning: The update of the recovery partition is a critical operation. Please be sure that the system is connected to a stable power supply to avoid unrecoverable system corruption.

5.1 Recovery command

Recovery package deployment is executed using 'recovery' command after a valid package is uploaded in machine /tmp directory.

These are the steps to complete the recovery update operation:

1. Upload a valid recovery package in product /tmp directory via scp command or WinSCP application
2. From Kepler BBU command line execute '*recovery upgrade*' command.

This should be the console output:

```
$ recovery upgrade
CAUTION: this upgrade needs to be connected to a stable power supply
If power goes down during upgrade the system may be corrupted
Continue upgrade? [y/N]
y
Serch for recovery package [PRESENT]
Extract recovery package [DONE]
Get configuration file [DONE]
Check package files integrity [DONE]
Mount recovery filesystem (read-only) [DONE]
Get Actual system information [DONE]
Check os compatibility [VALID]
Check platform version [2.2.3]
Check u-boot version [2013.01.14.r6.6]
Umount recovery filesystem (ro) [DONE]
Mount recovery filesystem (read-write) [DONE]
Remove old recovery [DONE]
Sync filesystem [DONE]
Copy new recovery [DONE]
Sync filesystem [DONE]
Verifying skern [DONE]
Verifying kernel [DONE]
Verifying devicetree [DONE]
Verifying initrd [DONE]
Verifying rootfs [DONE]
Verifying uboot [DONE]
Verifying configuration [DONE]
Erase U-Boot flash partition:100% |#####| Time: 0:00:00 105.14 MB/s
Write U-Boot flash partition:100% |#####| Time: 0:00:00 4.09 MB/s
Sync filesystem [DONE]
Umount recovery filesystem (rw) [DONE]

Upgrade ended successfully
```

5.2 Recovery Information

From Platform 2.2.3 there is a method to get information about the current recovery system installed. Executing the command *'recovery info'* will report this information.

```
$ recovery info
Codename:          tws8300bbu
Version:           2.2.3
Bootloader:        2013.01.14.r6.6
Build:             6
```

6 Automatic Reboot from Recovery Mode

From Platform 2.2.3 release, an automatic procedure to restart the machine when falling in Recovery Mode has been introduced. The feature behaviour is the following:

1. When the system starts in Recovery Mode, a timer is armed (60 secs).
2. In case the timer expires, system restart is triggered to end-up in Normal Mode.
3. The command '*backtomainstop*' is available for operator to stop the automatic restart (see next paragraph); in case the operator interrupts the countdown, system remains in recovery mode.
4. A counter is maintained to prevent an infinite loop of restart in recovery (in case the system is not able to restore normal operation after the restart); after 3 restarts the system remains in Recovery Mode.
5. When the system starts in Normal Mode, recovery loop counter is reset.

6.1 Stop Automatic Reboot

In order to stop the automatic reboot feature and remain in Recovery Mode an operator can execute the *'backtomainstop'* command.

```
AZB_NCOM_4300 login: root

          \      _     /   -    _   |   \       `--'   \
         _-/_\_-/_/_(C)(C)|_/_|_|_|_|
        _/_/_\_/_\_|\_\_|_\_/___|__|_|_|

recovery-tws8300bbu

!!!! RECOVERY !!!!!

The last boot was unsafe. A recovery image is running now.

root@AZB_NCOM_4300:~# backtomainstop
Automatic restart stopped
root@AZB_NCOM_4300:~#
```

7 Update device to New Filesystem Layout (Platform 2.0)

7.1 Setup

7.1.1 Hardware

Updating device from Platform version 1 to Platform version 2 needs the following hardware:

- PC running Linux or Windows with Ethernet and a serial port or a serial adapter (es. USB to Serial)
- Straight pin-to-pin serial cable with D-Sub 9pin female connectors on both ends
- Cat5e Ethernet cable with RJ45 connectors

7.1.2 Software

To execute the update from Platform version 1 to Platform version 2 is necessary the following software:

- Serial port communication program
 - Windows: Teraterm or Putty
 - Linux: minicom (in command line)
- TFTP server:
 - Windows: TFTPd (64 or 32 bit version)
 - Linux: xinetd TFTPd service or tftpd-hpa daemon

7.1.3 Configuration

In following points there will be the configuration of the PC needed to complete the update:

- Serial Port
 - Speed: 115200
 - Data: 8
 - Parity: None
 - Stop bits: 1
- TFTP Server Ethernet Port
 - Address: 192.168.1.254
 - Netmask: 255.255.255.0
 - Gateway: None
- TFTP Server directory structure:
 - / (tftp root directory)
 - platform_2.0
 - boot-bb-kepler.azubi
 - bootcore-bb-kepler.img
 - drec-bb-kepler.azubi
 - pers-bb-kepler.azubi
 - u-boot-bb-kepler.img
 - u-boot-recovery-bb-kepler.img
 - platform-tws8300bbu_2.0.0.azpkg

7.2 Procedure

1) Enter in recovery bootloader:

a) Power up the machine

b) Enter in u-boot console by pressing a key during the countdown:

```
Hit any key to stop autoboot: 0
AZB_NCOM_4300 #
```

c) Reset 'safeboot' variable and restart:

```
AZB_NCOM_4300 # safeboot reset
Safeboot successfully reset
AZB_NCOM_4300 # reset
resetting ...
```

d) Enter in u-boot recovery console by pressing a key during the countdown:

```
Hit any key to stop autoboot: 0
AZB_NCOM_4300 #
```

e) Verify u-boot recovery console finding 'U-Boot 2013.01.14-RECOVERY' in version command:

```
AZB_NCOM_4300 # ver

U-Boot 2013.01.14-RECOVERY-r18-git+552e236 (May 14 2016 - 03:10:23)
arm-azcom-linux-gnueabi-gcc (GCC) 4.9.1
GNU ld (GNU Binutils) 2.24
AZB_NCOM_4300 #
```

If 'RECOVERY' is not in bootloader version please repeat steps from 1.c to 1.e

2) Set u-boot IP networking:

a) Set machine IP address to 192.168.1.1:

```
AZB_NCOM_4300 # setenv ipaddr 192.168.1.1
```

b) Set TFTP server IP address:

```
AZB_NCOM_4300 # setenv serverip 192.168.1.254
```

c) Set IP gateway (if direct connection use same IP address of TFTP server):

```
AZB_NCOM_4300 # setenv gatewayip 192.168.1.254
```

d) Set netmask:

```
AZB_NCOM_4300 # netmask=255.255.255.0
```

e) Set transfer type:

```
AZB_NCOM_4300 # setenv tftp_cmd tftp
```

f) Set TFTP root directory:

```
AZB_NCOM_4300 # setenv tftp_root "platform_2.0"
```

3) Flash new bootcore and recovery bootloader:

a) Update bootcore:

```
AZB_NCOM_4300 # run get_bootcore_net burn_bootcore
Using BB_KEPLER_EMAC device
TFTP from server 192.168.1.254; our IP address is 192.168.1.1
Filename 'platform_2.0/bootcore-bb-kepler.img'.
Load address: 0x86000000
Loading: #####
          878.9 KiB/s

done
Bytes transferred = 262144 (40000 hex)

NAND erase.part: device 0 offset 0x0, size 0x40000
Erasing at 0x20000 -- 100% complete.
OK

NAND write: device 0 offset 0x0, size 0x40000
          262144 bytes written: OK
AZB_NCOM_4300 #
```

b) Update recovery bootloader:

```
AZB_NCOM_4300 # run get_uboot_r_net burn_uboot_r
Using BB_KEPLER_EMAC device
TFTP from server 172.25.10.5; our IP address is 172.25.22.103
Filename 'platform_2.0/u-boot-recovery-bb-kepler.img'.
Load address: 0x87000000
Loading: #####
          1.4 MiB/s

done
Bytes transferred = 361416 (583c8 hex)

NAND erase.part: device 0 offset 0x40000, size 0x100000
Erasing at 0x120000 -- 100% complete.
OK

NAND write: device 0 offset 0x40000, size 0x583c8
          361416 bytes written: OK
AZB_NCOM_4300 #
```

c) Restart the machine:

```
AZB_NCOM_4300 # reset
resetting ...
```

d) Enter in u-boot console by pressing a key during the countdown:

```
Hit any key to stop autoboot: 0
AZB_NCOM_4300 #
```

4) Reset environment parameters to default

a) Reset environment variables:

```
AZB_NCOM_4300 # env default -a
## Resetting to default environment
AZB_NCOM_4300 #
```

5) Set u-boot IP networking:

a) Set machine IP address:

```
AZB_NCOM_4300 # setenv ipaddr 192.168.1.1
```

b) Set TFTP server IP address:

```
AZB_NCOM_4300 # setenv serverip 192.168.1.254
```

c) Set IP gateway (if direct connection use same IP address of TFTP server):

```
AZB_NCOM_4300 # setenv gatewayip 192.168.1.254
```

d) Set netmask:

```
AZB_NCOM_4300 # netmask=255.255.255.0
```

e) Set transfer type:

```
AZB_NCOM_4300 # setenv tftp_cmd tftp
```

f) Set TFTP root directory:

```
AZB_NCOM_4300 # setenv tftp_root "platform_2.0"
```

6) Re-Flash filesystem with New Filesystem Layout:

a) Erase whole NAND flash:

```
AZB_NCOM_4300 # nand erase.chip

NAND erase.chip: device 0 whole chip
Skipping bad block at 0x0ff00000
Skipping bad block at 0x2ff00000
Erasing at 0x3ffe0000 -- 100% complete.
OK
AZB_NCOM_4300 #
```

b) Update bootcore:

```
AZB_NCOM_4300 # run get_bootcore_net burn_bootcore
Using BB_KEPLER_EMAC device
TFTP from server 192.168.1.254; our IP address is 192.168.1.1
Filename 'platform_2.0/bootcore-bb-kepler.img'.
Load address: 0x86000000
Loading: #####
      878.9 KiB/s

done
Bytes transferred = 262144 (40000 hex)

NAND erase.part: device 0 offset 0x0, size 0x40000
Erasing at 0x20000 -- 100% complete.
OK

NAND write: device 0 offset 0x0, size 0x40000
 262144 bytes written: OK
AZB_NCOM_4300 #
```

c) Update recovery bootloader:

```
AZB_NCOM_4300 # run get_uboot_r_net burn_uboot_r
Using BB_KEPLER_EMAC device
TFTP from server 172.25.10.5; our IP address is 172.25.22.103
Filename 'platform_2.0/u-boot-recovery-bb-kepler.img'.
Load address: 0x87000000
Loading: #####
          1.4 MiB/s
done
Bytes transferred = 361416 (583c8 hex)

NAND erase.part: device 0 offset 0x40000, size 0x100000
Erasing at 0x120000 -- 100% complete.
OK

NAND write: device 0 offset 0x40000, size 0x583c8
          361416 bytes written: OK
AZB_NCOM_4300 #
```

d) Update main bootloader:

```
AZB_NCOM_4300 # run get_uboot_net burn_uboot
Using BB_KEPLER_EMAC device
TFTP from server 192.168.1.254; our IP address is 192.168.1.1
Filename 'platform_2.0/u-boot-bb-kepler.img'.
Load address: 0x87000000
Loading: #####
          1.5 MiB/s
done
Bytes transferred = 361424 (583d0 hex)

NAND erase.part: device 0 offset 0x2180000, size 0x100000
Erasing at 0x2260000 -- 100% complete.
OK

NAND write: device 0 offset 0x2180000, size 0x583d0
          361424 bytes written: OK
AZB_NCOM_4300 #
```

e) Update boot partition:

[illegible]

f) Update Disaster Recovery (drec) partition:

[illegible]

g) Update persistent partition:

```
AZB_NCOM_4300 # run get_ubipers_net burn_ubipers
Using BB_KEPLER_EMAC device
TFTP from server 172.25.10.5; our IP address is 172.25.22.103
Filename 'tw8300bbu_#11903/pers-bb-kepler.azubi'.
Load address: 0x82000000
Loading: #####
#####
#####
#####
#####
#####
#####
1.7 MiB/s
done
Bytes transferred = 3932160 (3c0000 hex)

NAND erase.part: device 0 offset 0x12340000, size 0x2dcc0000
Skipping bad block at 0x2ff00000
Erasing at 0x3ffe0000 -- 100% complete.
OK

NAND write: device 0 offset 0x12340000, size 0x3c0000
3932160 bytes written: OK
AZB NCOM 4300 #
```

h) Set 'safeboot' variable and restart:

```
AZB_NCOM_4300 # safeboot set
Safeboot successfully set
AZB_NCOM_4300 # reset
resetting ...
```

7) Now the machine is update to platform 2.0. We need now to upgrade the system using new platform packaging, so if FPGA needs to be updated it will be done automatically.

This is the procedure:

a) Login to main system:

```
Azcom (Azcom Technology Reference Distro) 2.0.0rc15 AZB_NCOM_4300 /dev/ttyS0
AZB_NCOM_4300 login: root
```

b) Download platform package to machine /tmp directory, for example using scp command.

Remember to substitute '<user>' with the correct TFTP Server user name

```
root@AZB_NCOM_4300:~# scp <user>@192.168.1.254:/tftpboot/platform_2.0/platform-
tw8300bbu_2.0.0.azpkg /tmp/
<user>@192.168.1.254's password:
platform-tw8300bbu_2.0.0.azpkg          100%   32MB   8.0MB/s   00:04
root@AZB_NCOM_4300:~#
```

c) Retrieve platform with 'force' optional argument:

```
root@AZB_NCOM_4300:~# platform retrieve --force
Serch for platform [PRESENT]
Clean deploy directory [DONE]
Get configuration file [DONE]
Parse OS release [DONE]
Check tw8300bbu compatibility [VALID]
Parse OS release [DONE]
Check platform version [2.0.0]
Deploying skern [DONE]
Deploying kernel [DONE]
Deploying devicetree [DONE]
Deploying initrd [DONE]
Deploying rootfs [DONE]
Deploying fpga [DONE]
Deploying uboot [DONE]
Deploying configuration [DONE]
Deploying configuration file [DONE]
Sync filesystem [DONE]
Mark retrieve as ready to upgrade [DONE]

Platform is now ready to be upgraded

root@AZB_NCOM_4300:~#
```

d) Upgrade platform:

```
root@AZB_NCOM_4300:~# platform upgrade
Check for platform status      [VALID]
Verifying skern                [DONE]
Verifying kernel              [DONE]
Verifying devicetree          [DONE]
Verifying initrd              [DONE]
Verifying rootfs              [DONE]
Verifying fpga                [DONE]
Verifying uboot               [DONE]
Verifying configuration        [DONE]
Check u-boot version          [SAME VERSION]
Check fpga version            [SAME VERSION]
Mark system to be upgraded     [DONE]

Platform is now rebooted to be upgraded
```

After upgrade procedure (see chapter 4.1) the platform is now updated to version 2.0.0

8 Dip-switch configuration

Condition	Dip #4	Dip #5	Dip #6
No Boot	OFF	OFF	OFF
Nand Boot	OFF	OFF	ON

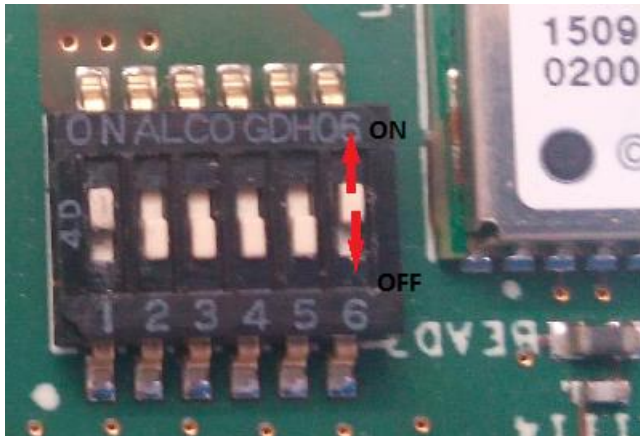


Figure 2 Kepler BBU DIP switch

9 Update PIC firmware

It is possible to update the firmware of the Microchip PIC.

For this updating you need:

- The Microchip software “AN1310 bootloader”
<http://ww1.microchip.com/downloads/en/AppNotes/Serial%20Bootloader%20AN1310%20v1.05r.zip> installed on a PC windows 7 or later
- SSH client for windows (Putty, WinSCP or equivalent)
- A serial cable connected between PC window and the serial line of the Azcom Kepler BBU
- An Ethernet cable connected between PC windows and the ETH0 of
- The new binary file of the PIC application (**picappl-bb-kepler_XXX.hex**)

After installing of the AN1310 software please follows the following steps:

- Connect to SOC via SSH
- At the linux prompt executes the following commands
 - ***fpga -w 0x86 1***
 - ***sync***
- SSH disconnect
- Start **Teraterm** application (or equivalent) and select the serial line connected to the board:
 - press the key **<R>** to send this character to the PIC
 - if the PC receives this character, the PICLED lights up in **RED color**: this means that PIC will reboot in 10 seconds
- Within 10 seconds, close the Teraterm and starts **AN1310 v1.5r**
- From this app:
 - Press **<F3>** key (= Break/Reset mode)
 - Wait for the PIC rebooting: when the PICLED will light up **WHITE**
 - Press **<F4>** key (= Bootloader mode): it will be show the bootloader version of the PIC (=1.05)
 - Press **<F7>** key (= Erase Device): PIC will erase the application flash partition
 - Press **<F6>** key (= Write Device): the application will ask the filename of the new binary application **picappl-bb-kepler_XXX.hex**
 - At the end of the transfer, press the **<F2>** key (= Run Mode) and wait the PICLED will light up **YELLOW**
 - Close the AN1310 app.
- Turn off the board (PIC included)
- Start again the Teraterm application selecting the same serial line
- Turn on the board
- Press the PIC button (if it is not already pressed) and wait for the linux “login”
- At the prompt executes the command ***picinfo*** to verify the new version of the PIC.