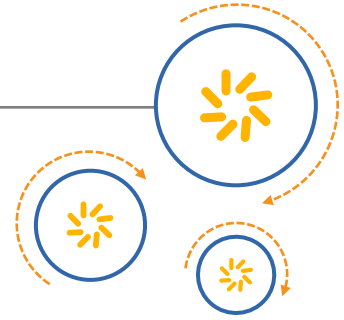




Qualcomm Technologies, Inc.



# Lauterbach Debugger for IPQ4018/IPQ4019/IPQ4028/IPQ4029

## Application Note

80-Y9571-5 Rev. F

August 25, 2016

**Confidential and Proprietary – Qualcomm Technologies, Inc.**

**NO PUBLIC DISCLOSURE PERMITTED:** Please report postings of this document on public servers or websites to:

[DocCtrlAgent@qualcomm.com](mailto:DocCtrlAgent@qualcomm.com).

**Restricted Distribution:** Not to be distributed to anyone who is not an employee of either Qualcomm Technologies, Inc. or its affiliated companies without the express approval of Qualcomm Configuration Management.

Not to be used, copied, reproduced, or modified in whole or in part, nor its contents revealed in any manner to others without the express written permission of Qualcomm Technologies, Inc.

Qualcomm Internet Processor is a product of Qualcomm Technologies, Inc. Other Qualcomm products referenced herein are products of Qualcomm Technologies, Inc., or its subsidiaries.

Qualcomm is a trademark of Qualcomm Incorporated, registered in the United States and other countries. Other product and brand names may be trademarks or registered trademarks of their respective owners.

This technical data may be subject to U.S. and international export, re-export, or transfer ("export") laws. Diversion contrary to U.S. and international law is strictly prohibited.

Qualcomm Technologies, Inc.  
5775 Morehouse Drive  
San Diego, CA 92121  
U.S.A.

## Revision history

Revision	Date	Description
A	July 2015	Initial release
B	September 2015	Updated Section <a href="#">6.2</a>
C	November 2015	Updated Section <a href="#">8</a>
D	April 2016	Updated Section <a href="#">6.2</a>
E	May 2016	Updated Section <a href="#">6.2</a> and <a href="#">Figure 6-2</a>
F	August 2016	Updated Section <a href="#">6.2</a>

QUALCOMM  
2016-10-12 19:15:44 PDT  
quanhai.zhang@arrowasia.com

# Contents

---

<b>1 Purpose</b>	<b>4</b>
<b>2 Introduction</b>	<b>5</b>
<b>3 Topology</b>	<b>6</b>
<b>4 T32 installation procedure</b>	<b>8</b>
<b>5 DK_T32 Package</b>	<b>9</b>
<b>6 Image flashing procedure</b>	<b>10</b>
6.1 Starting T32	10
6.1.1 Starting T32 Cortex-A7 window	11
6.2 Flashing the images	11
<b>7 Reboot via T32</b>	<b>14</b>
<b>8 T32 Linux menu for debugging</b>	<b>15</b>
<b>9 Troubleshooting</b>	<b>16</b>

## Figures

Figure 3-1 Lauterbach connect position on APXXX reference platform	6
Figure 3-2 Lauterbach on DK.01 reference platform	7
Figure 5-1 DK_T32 package directory structure	9
Figure 6-1 T32 start window	10
Figure 6-2 ARM Core-0 T32 start window	11
Figure 6-3 ARM Core-0 T32 start window after a successful u-boot	13

# 1 Purpose

---

This document describes the usage of the Lauterbach debugger (Trace32-ARM) for the IPQ4018/IPQ4019/IPQ4028/IPQ4029 reference platform.

QUALCOMM  
2016-10-12 19:15:44 PDT  
quanhai.zhang@arrowasia.com

## 2 Introduction

---

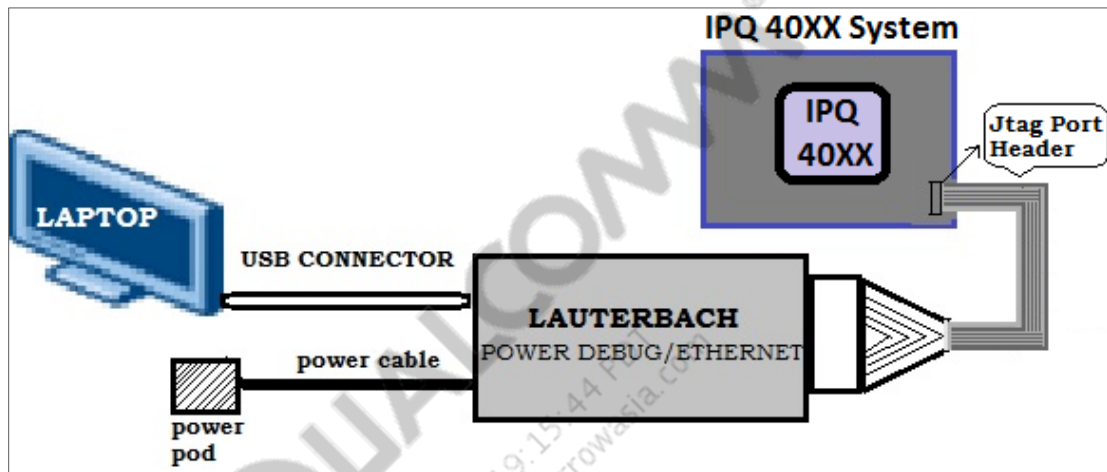
The Lauterbach debugger is the advanced tool for debugging and flashing image on the IPQ40xx reference platform. It provides detailed information about CPU register dump and symbol details in line with the source code. The 20-pin JTAG header on IPQ40xx platform is mounted to connect the Lauterbach debugger externally.

<File>.cmm is the script file with cmm- style command; the script file uses one command on each line.

1. In TRACE32, choose  
File → Run Batchfile as <file.cmm>.
2. Enter `DO file.cmm` or `CD.D0 file.cmm` in the trace32 command line.

### 3 Topology

---



**Figure 3-1 Lauterbach connect position on APXXX reference platform**

NOTE: This document is based on the following details:

- Lauterbach LA-7705 – Power debug module ethernet
- Lauterbach LA-7746 – JTAG debugger for ARM7 20-pin connector (ICD)
- Lauterbach LA-7843X –JTAG debugger extension for CORTEX-A/-RDB147-010
- Lauterbach T32 software version should be greater than 2.2.35



**Figure 3-2 Lauterbach on DK.01 reference platform**

## 4 T32 installation procedure

---

TRACE32 supports many operating systems. While installing TRACE32, specify the processor to be used (MIPS, XTENSA, or ARM).

During the installation process, use the following settings:

	Select
<b>Setup type</b>	New installation
<b>Product type</b>	ICD In-Circuit Debugger
<b>Interface type</b>	USB interface
<b>Development tools</b>	License key not necessary
<b>OS selection</b>	PC Windows 2000/XP/VISTA/7
<b>CPU selection</b>	ICD XTENSA, ICD MIPS32, ICD ARM (ARM 7,9,10,11, Cortex, Janus, XScale)
<b>Screen configuration</b>	Multiple Document Interface (MDI)
<b>Prepare TRACE32 for integrations with other products</b>	No integration
<b>Folder program group type</b>	Common
<b>How to submit registration</b>	Register later

**NOTE:** See the *IPQ4018/IPQ4028 AP.DK01 Setup Guide* (80-Y9700-2) for the Lauterbach connector and its placement.

For a license key or to register during installation, check with the Lauterbach vendor for detailed information. Use Trace32 of version P.2013.02.000044559 or greater for better results.



## 5 DK\_T32 Package

---

Figure 5-1 shows the directory structure of the delivered DK\_T32 package.



```
Dakota_start.ts2
└─cmm_scripts
    AR_DAP_CortexA7_Core0_5TAP.cmm
    ddr_init_16bit.cmm
    gcc_init.cmm
    semihost_setup.cmm
```

**Figure 5-1 DK\_T32 package directory structure**

# 6 Image flashing procedure

Image flashing procedure is for IPQ4018/IPQ4019/IPQ4028/IPQ4029 reference platform.

## 6.1 Starting T32

Use an available PC COM port with the following settings:

- Speed 115200 bits per second
- 8 data bits, 1 stop bit, 0 parity bit
- No flow control

To start T32, run **DK\_cmm\_scripts\Dakota\_start.ts2** to open the T32 start window, as shown in [Figure 6-1](#).

NOTE: Ignore warning messages, if any.



Figure 6-1 T32 start window

### 6.1.1 Starting T32 Cortex-A7 window

In the T32Start V2.2.35 window, do the following:

- Select **5TAP\_A7\_and\_Tensilica** > 1: Podbus Device Chain > 1: Power Trace Ethernet
- Select **A7\_Core0**
- Click on the **Start** button to open the ARM Core-0 T32 start window, as shown in [Figure 6-2](#).

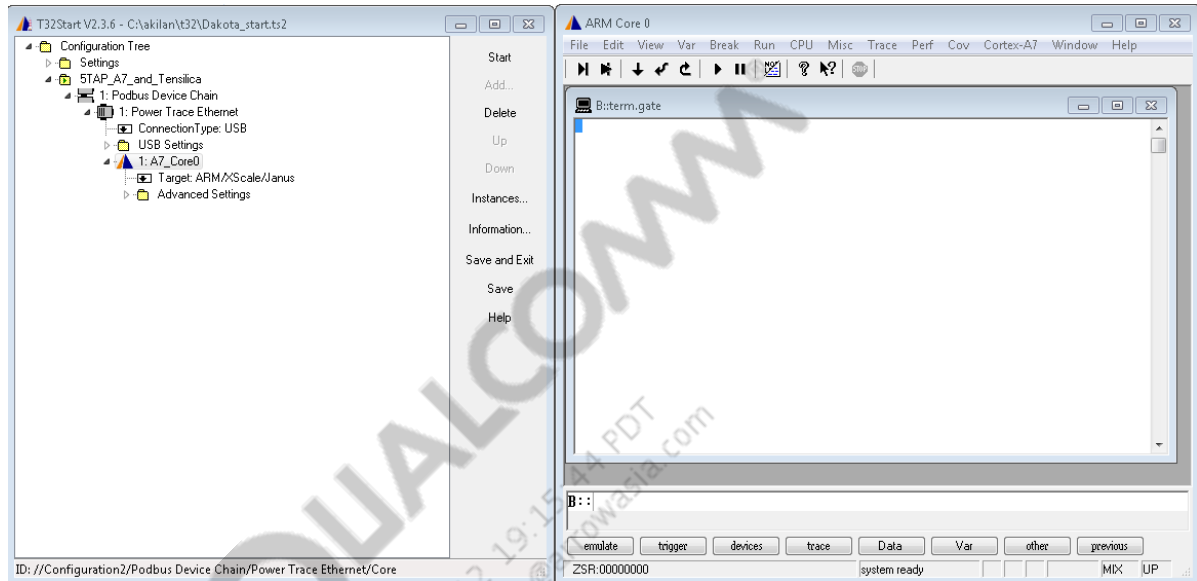


Figure 6-2 ARM Core-0 T32 start window

## 6.2 Flashing the images

In the DAP (ARM Core-0) ARM7 window, run the following commands:

- Initialize all clocks:
 

```
B::winclear
B::sys.d
B::sys.u
CD.DO gcc_init.cmm
```
- Initialize DDR:
 

```
CD.DO ddr_init_16bit.cmm
```
- Load the IPQ40xx (NAND/NOR/eMMC) single image to flash the new image:
 

```
data.load.binary $IMAGE_DIR<nand/nor/emmc>-ipq40xx-single.img
0x84000000
```
- Load SMEM:
 

```
data.load.binary $IMAGE_DIR\smem-AP.DK01.1-C1.bin 0x87e00000
```
- Load U-Boot:
 

```
data.load.elf $IMAGE_DIR\openwrt-ipq40xx-u-boot.elf
```
- Start execute U-Boot:
 

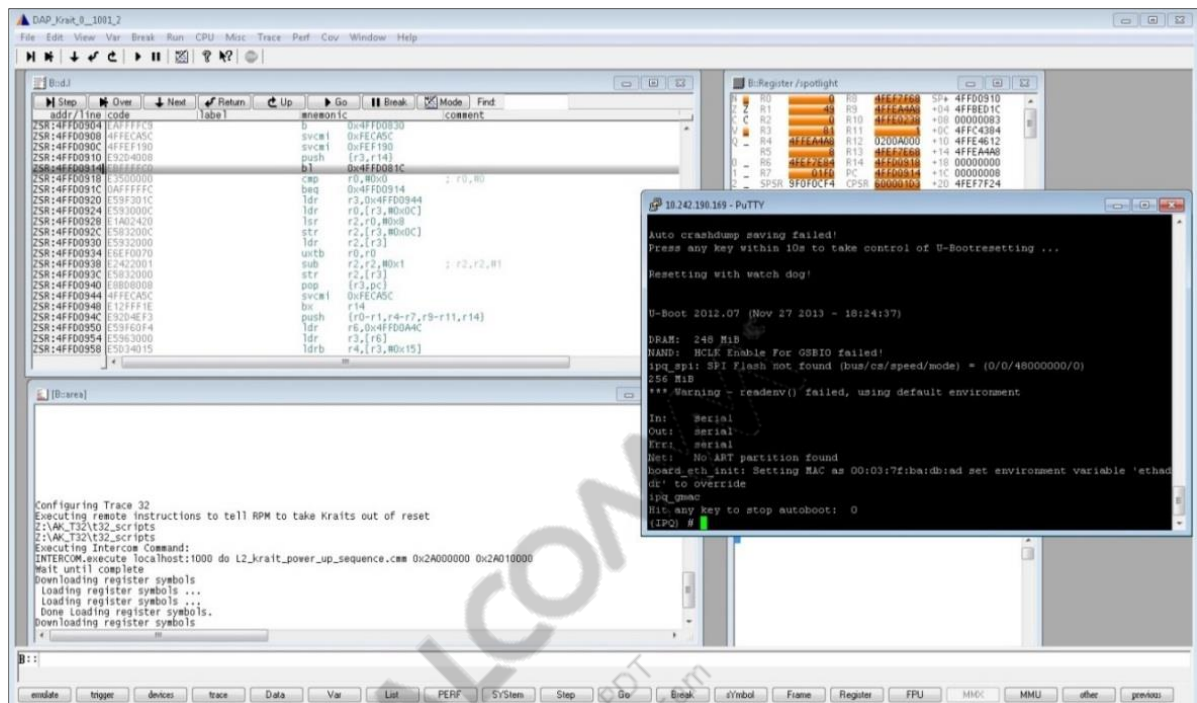
```
Go
```

NOTE: The single.img and smem.bin/U-Boot are compiled from the software releases.

Recommended build version required		
Pre-Release	Software version	P.2013.02.000044559
	Build	42354-44559
	Compiled	May 14 2013 (02:54:09)
T32 scripts	\$MetaFolder\common\t32\cmm_scripts	
U-Boot path	\$MetaFolder\common\build\ipq\openwrt-ipq40xx-u-boot.elf	
smem path	\$MetaFolder\common\build\ipq\smem-AP.DK01.1-C1.bin \$MetaFolder\common\build\ipq\smem-AP.DK01.1-C2.bin (NOR + NAND) \$MetaFolder\common\build\ipq\smem-AP.DK04.1-C1.bin \$MetaFolder\common\build\ipq\smem-AP.DK04.1-C2.bin \$MetaFolder\common\build\ipq\smem-AP.DK04.1-C3.bin \$MetaFolder\common\build\ipq\smem-AP.DK04.1-C5.bin \$MetaFolder\common\build\ipq\smem-AP.DK01.1-S1.bin (NOR-16MB) \$MetaFolder\common\build\ipq\smem-AP.DK05.1-C1.bin \$MetaFolder\common\build\ipq\smem-AP.DK06.1-C1.bin \$MetaFolder\common\build\ipq\smem-AP.DK07.1-C1.bin	
single.img path	\$MetaFolder\common\build\bin\nor-ipq40xx-single.img \$MetaFolder\common\build\bin\nand-ipq40xx-single.img \$MetaFolder\common\build\bin\emmc-single.img \$MetaFolder\common\build\bin\nornand-ipq40xx-single.img \$MetaFolder\common\build\bin\norplusemmc-ipq40xx-single.img \$ StandardProfileMetaFolder\common\build\bin\ nor-ipq40xx-standard-single.img	

NOTE: For DK03, corresponding smem binaries of DK01 can be used.

The U-Boot prompt appears as shown in [Figure 6-3](#). Stop the U-Boot before it goes for auto boot.



**Figure 6-3 ARM Core-0 T32 start window after a successful u-boot**

After the successful U-Boot, run the following commands at the U-Boot prompt to flash the new image:

(Qualcomm® Internet Processor (IPQ)) # imgaddr=0x84000000

```
(IPQ) # source $imgaddr:script
```

## 7 Reboot via T32

---

In the DAP (ARM Core-0) Cortex-A7 window, click:

1. “**Break**” to stop ARM Core-0
2. “CPU->In Target Reset”
3. "CPU->Reset CPU Register"
4. “**Go**” to start Cortex A7

Watch the console for U-Boot.

QUALCOMM®  
2016-10-12 19:15:44 PDT  
quanhai.zhang@arrowasia.com

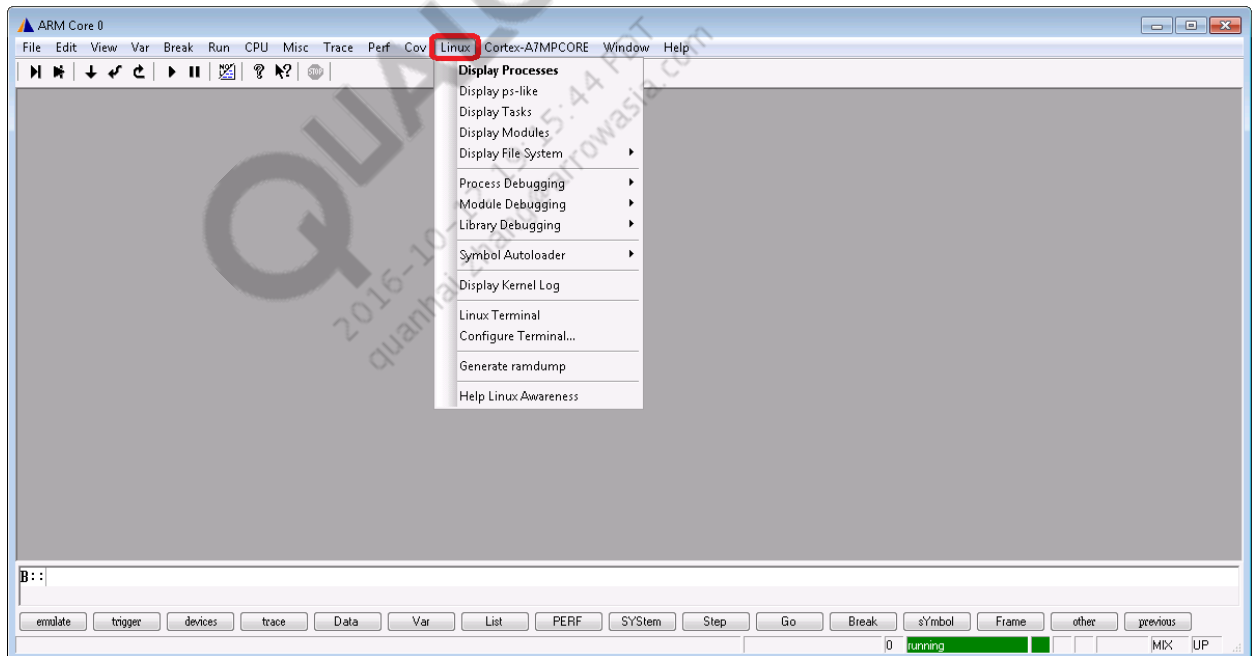
## 8 T32 Linux menu for debugging

---

In the DAP (ARM Core-0) Cortex-A7 window, do the following:

1. Load the u-image after getting kernel prompt.  
`d.load.elf <u-image> /nocode`
2. Execute the following scripts:  
`task.config ~/demo/arm/kernel/linux/linux-3.x/linux3.t32`  
`menu.rp ~/demo/arm/kernel/linux/linux-3.x/linux.men`

See the “Linux” menu option in the T32 window as shown in the figure.



## 9 Troubleshooting

---

1. What if ARM Cortex-A7 does not respond during out of reset procedure?
  - Power recycle the board and T32, and redo the steps again.
2. What if the U-Boot console does not appear?
  - Ensure that correct smem.bin is loaded in DDR at location 0x87e00000 prior to starting U-Boot
  - Ensure that U-Boot is error-free
3. What if U-Boot is not appearing in the AP.DK01/ AP.DK.04 board?

The smem.bin is different for different boards. Ensure that the correct smem.bin is loaded for the AP.DK01/AP.DK.04 boards.