OpenOCD primer – K3 Devices

Date: 2023-09-22

Nishanth Menon

Texas Instruments

Overview

- What is OpenOCD
- Features of OpenOCD
- OpenOCD overview
- OpenOCD Hardware setup: xds110, cTI20, tag-connect
- Building OpenOCD
- OpenOCD integration with IDEs
- Debugging U-Boot
- Debugging Linux Kernel
- Self Hosted Debug

Texas Instruments

What is OpenOCD

OpenOCD (Open On-Chip Debugger) is an open-source software project that provides debugging and in-circuit programming support for various microcontrollers and microprocessors. It allows developers to interact with and control the hardware of embedded systems during both development and production phases. OpenOCD supports a wide range of target architectures and interfaces with various hardware debugging probes.

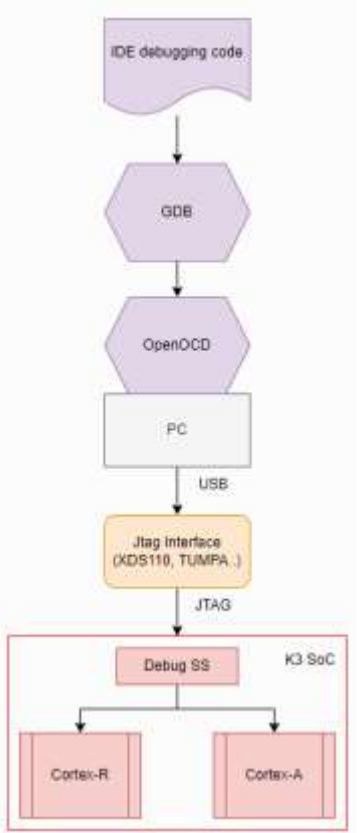
Texas Instruments

Features of OpenOCD

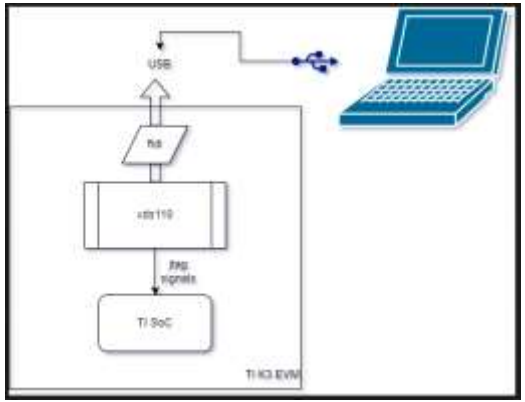
- **Debugging**: OpenOCD enables developers to perform source-level debugging, inspect memory, set breakpoints, and single-step through code on embedded systems.
- **Hardware Debug Probes**: OpenOCD interfaces with hardware debug probes like JTAG (Joint Test Action Group) and SWD (Serial Wire Debug) adapters to connect to the target hardware.
- **GDB Integration**: OpenOCD can be used as a target for the GNU Debugger (GDB), allowing developers to use familiar debugging tools with their embedded systems.
- **Scripting**: OpenOCD can be scripted to automate various debugging and programming tasks, making it useful for both manual and automated testing.
- **Flash Programming**: OpenOCD supports programming the flash memory of microcontrollers and microprocessors, allowing developers to load and update firmware.
- **Support for Various Targets**: OpenOCD provides support for a wide range of target architectures, including ARM, MIPS, RISC-V, and others, making it versatile for different embedded systems.
- **Cross-Platform**: OpenOCD is cross-platform and can be used on various operating systems, including Windows, Linux, and macOS.

Texas Instruments

OpenOCD overview



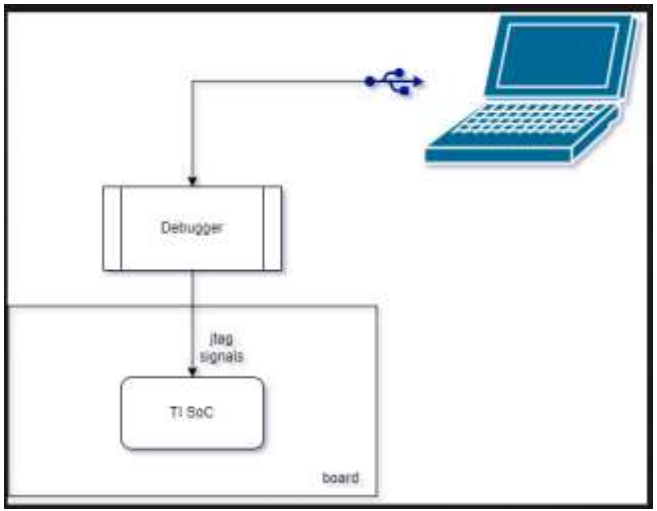
OpenOCD hardware setup – XDS110

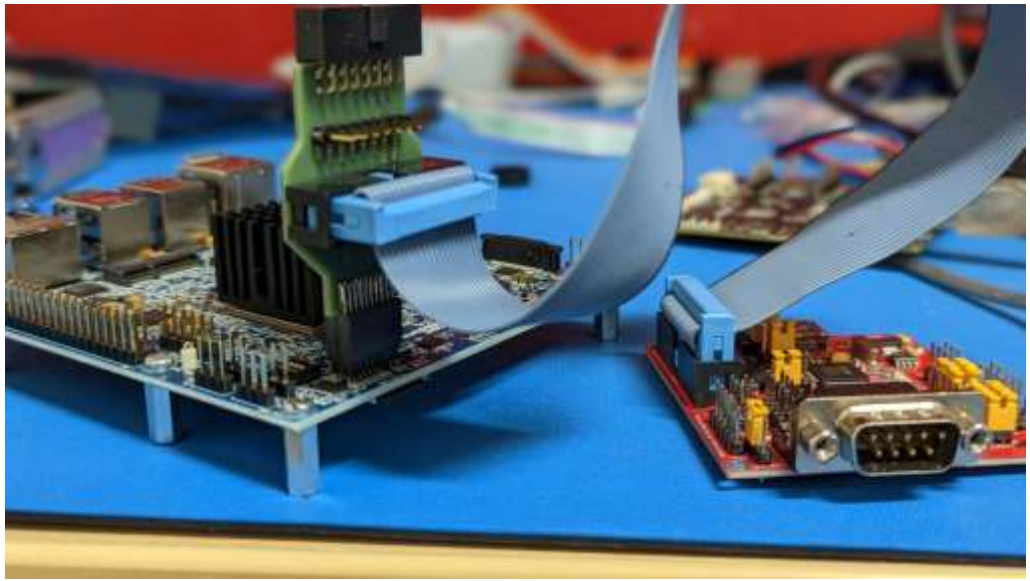


Connect a micro-USB cable to the board to the mentioned port.

[K3 Generation — Das U-Boot unknown version documentation]

OpenOCD hardware setup – cTI20

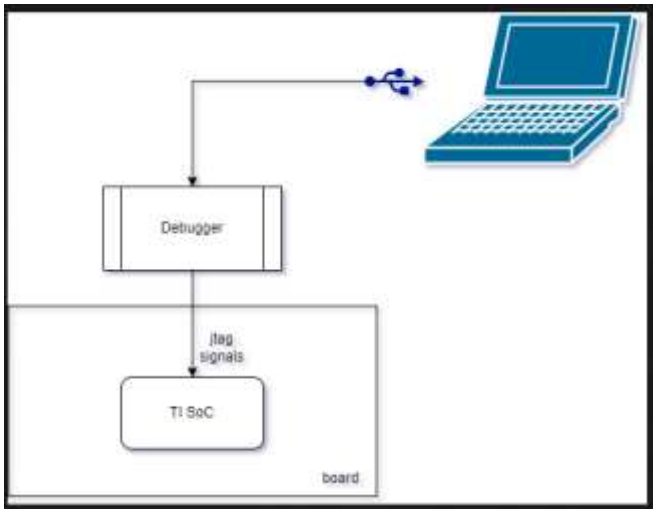


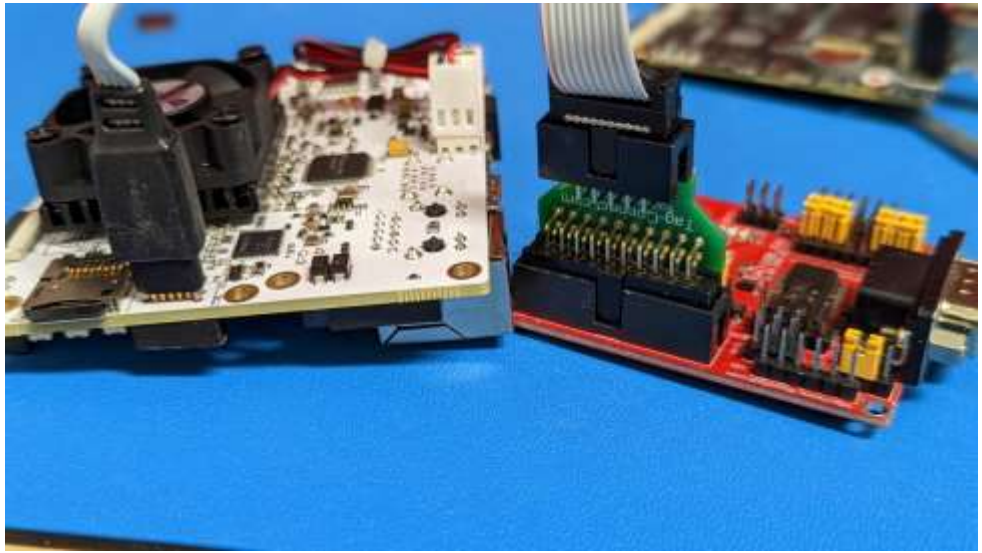


•TUMPA or equivalent dongles supported by **OpenOCD**.
•Cable such as Tag-connect ribbon cable
•Adapter to convert cTI20 to ARM20 such as those from Segger or Lauterbach LA-3780 Or optionally, if you have manufacturing capability then you could try BeagleBone JTAG Adapter

K3 Generation — Das U-Boot unknown version documentation

OpenOCD hardware setup – tagConnect





- TUMPA or equivalent dongles supported by **OpenOCD**.
- Tag-Connect cable appropriate to the board such as TC2050-IDC-NL
- In case of no-leg, version, a retaining clip
- Tag-Connect to ARM20 adapter

K3 Generation — Das U-Boot unknown version documentation

Building OpenOCD

```
# Check the packages to be installed: needs deb-src in sources.list
sudo apt build-dep openocd
# The following list is NOT complete - please check the latest
sudo apt-get install libtool pkg-config texinfo libusb-dev \
  libusb-1.0.0-dev libftdi-dev libhidapi-dev autoconf automake
git clone https://github.com/openocd-org/openocd.git openocd
cd openocd
git submodule init
git submodule update
./bootstrap
./configure --prefix=/usr/local/
make -j`nproc`
sudo make install
```

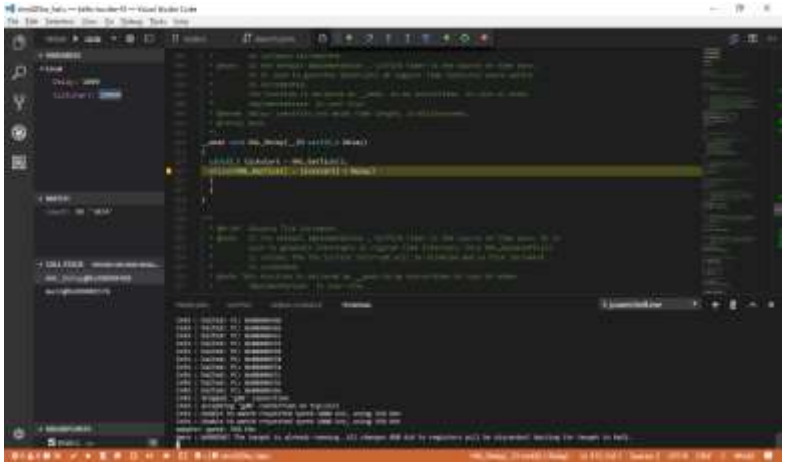
K3 Generation — Das U-Boot unknown version documentation

Texas Instruments

OpenOCD IDE integration with GDB

gdb-dashboard

GEF (GDB Enhanced Features)



Debugger Setup with GDB + OpenOCD in Visual Studio Code (justinmklam.com)

AND MANY MANY MANY MORE!

K3 Generation — Das U-Boot unknown version documentation

Debug U-boot

```
diff --git a/arch/arm/cpu/armv7/start.S b/arch/arm/cpu/armv7/start.S
index 69e281b086..744929e825 100644
--- a/arch/arm/cpu/armv7/start.S
+++ b/arch/arm/cpu/armv7/start.S
@@ -37,6 +37,8 @@
 #endif

 reset:
+dead_loop:
+    b dead_loop
     /* Allow the board to save important registers */
     b    save_boot_params
 save_boot_params_ret:
```

Option 1: reset

```
void board_init_f(ulong dummy)
{
    .
    .
    /* Code to run on the R5F (Wakeup/Boot Domain) */
    if (IS_ENABLED(CONFIG_CPU_V7R)) {
        volatile int x = 1;
        while(x) {};
    }
    ...
    /* Code to run on the ARMV8 (Main Domain) */
    if (IS_ENABLED(CONFIG_ARM64)) {
        volatile int x = 1;
        while(x) {};
    }
    .
    .
}
```

Option 2: **board_init_f**

[K3 Generation — Das U-Boot unknown version documentation]

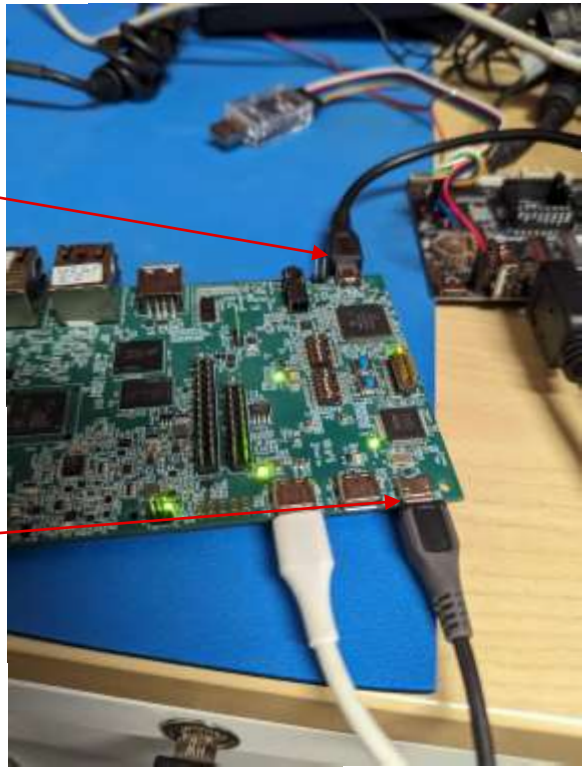
Texas Instruments

Demo – Debug U-Boot

xds110

AM625-SK

Console



K3 Generation — Das U-Boot unknown version documentation

Texas Instruments

Demo – Debug U-Boot – VsCode integration

https://gist.github.com/nmenon/c2984e28f1af73292c0bd472c7522460

Debug Kernel

- Step 1: Disable kernel options:
  - CONFIG_ARM_CORESIGHT_PMU_ARCH_SYSTEM_PMU=n
  - CONFIG_CORESIGHT=n
- Step 2: kernel command line options: Add:
  - rodata=off cpuidle.off=1 nokaslr
  - nosmp – to disable multi-cpu debug

```
openocd -c 'set V8_SMP_DEBUG 1' -c "set RTOS(am625.cpu.a53.0) hwthread" -f board/ti_am625evm.cfg
```

Pending Patches (on track to merge in next few weeks)
https://review.openocd.org/c/openocd/+/7896
https://review.openocd.org/c/openocd/+/7897
https://review.openocd.org/c/openocd/+/7898

Demo – Debug Kernel

Texas Instruments

OpenOCD Self hosted debug

- Debug M4F or R5F from A53 itself.

```
# Check the packages to be installed: needs deb-src in sources.list
sudo apt build-dep openocd
# The following list is NOT complete - please check the latest
sudo apt-get install libtool pkg-config texinfo libusb-dev \
  libusb-1.0.0-dev libftdi-dev libhidapi-dev autoconf automake
git clone https://github.com/openocd-org/openocd.git openocd
cd openocd
git submodule init
git submodule update
./bootstrap
./configure --prefix=/usr/local/ --enable-dmem
make -j`nproc`
sudo make install
```

```
sudo openocd -c "set RTOS(am625.cpu.gp_mcu) Zephyr" -f board/ti_am625_swd_native.cfg
```

References

- https://medium.com/@aliaksandr.kavalchuk/diving-into-jtag-protocol-part-1-overview-fbdc428d3a16
- https://medium.com/@aliaksandr.kavalchuk/diving-into-jtag-protocol-part-2-debugging-56a566db3cf8
- https://u-boot.readthedocs.io/en/latest/board/ti/k3.html#common-debugging-environment-openocd
- https://review.openocd.org/

Contribute to upstream OpenOCD