Programming Talaria TWO over JTAG

The INP3000 Programmer Board version 3.0 has the JTAG interface to program Talaria TWO modules.

Install the appropriate dependencies for programming over JTAG.

## In Windows

1. Open command prompt and reset the device in boot loader mode by executing the following command from the SDK directory:

|  |
| --- |
| .\script\reset.py evk42\_bl |

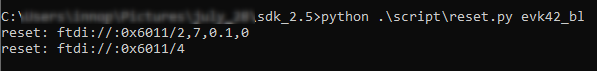


Figure 36: JTAG - Resetting the device in boot loader mode- console output (Windows)

1. Open the command prompt and start OpenOCD by executing the following command from the SDK directory:

|  |
| --- |
| openocd -s .\conf -f ftdi.cfg -f t2.cfg |

The following output is seen on command prompt console:

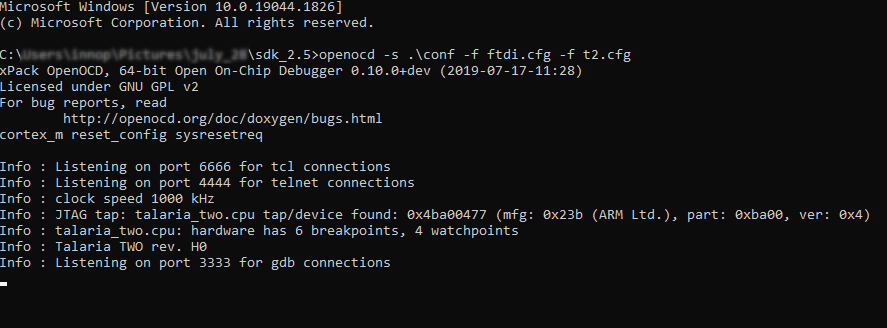


Figure 37: JTAG - Starting OpenOCD – console output

1. Flash the default partition table by executing the following command:

|  |
| --- |
| .\pc\_tools\T2\_Flasher\bin\T2\_Flasher\_Windows.exe --operation=write\_ptable .\<path to the standard\_part\_table.json file available in sdk\_x.y/tools/partition\_files directory> |

Following output is observed after flashing the default partition table:

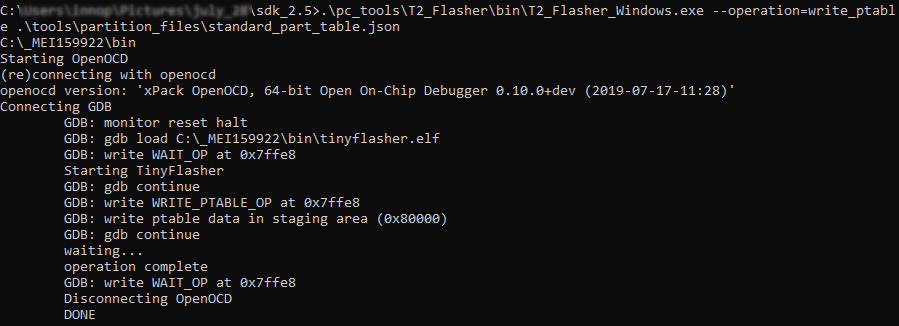


Figure 38: JTAG - Flashing the default partition table – console output

1. In the same terminal, execute the following command from the SDK directory to flash the data image (data.img).

**Note**: This step is needed only if the application uses any certificates.

|  |
| --- |
| .\pc\_tools\T2\_Flasher\bin\T2\_Flasher\_Windows.exe --operation=write\_part --partition=DATA .\<path to the generated data image>\data.img |

The following output is observed after flashing the data image:

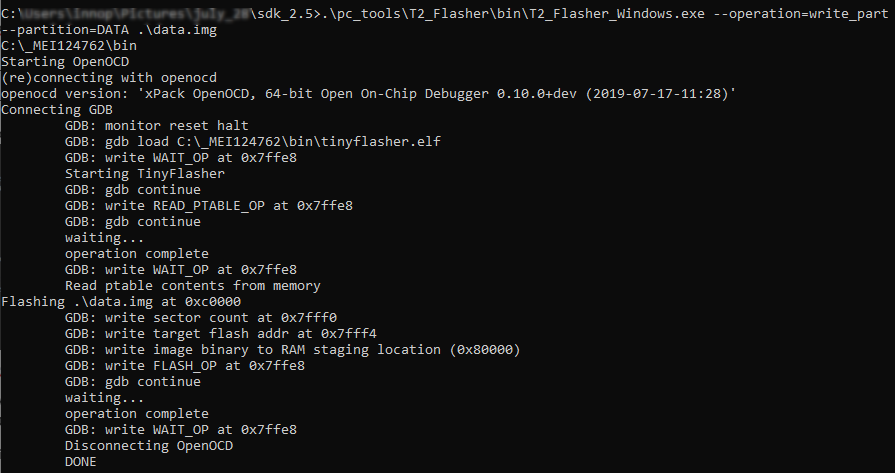


Figure 39: JTAG - Flashing the data image - console output

1. In a separate command prompt window, execute the following command from the SDK directory to flash the application image:

|  |
| --- |
| .\pc\_tools\T2\_Flasher\bin\T2\_Flasher\_Windows.exe --operation=write\_part --partition=BOOT .\<path to the generated application image>\app.img |

The following output is observed after flashing the application image:



Figure 40: JTAG - Flashing the application image - console output

1. In the same terminal, flash the VM image of the application by executing the following command:

|  |
| --- |
| .\pc\_tools\T2\_Flasher\bin\T2\_Flasher\_Windows.exe --operation=write\_part --partition=VIRT .\<path to the generated application image.vm> |

Following output is observed after flashing the application’s VM image:

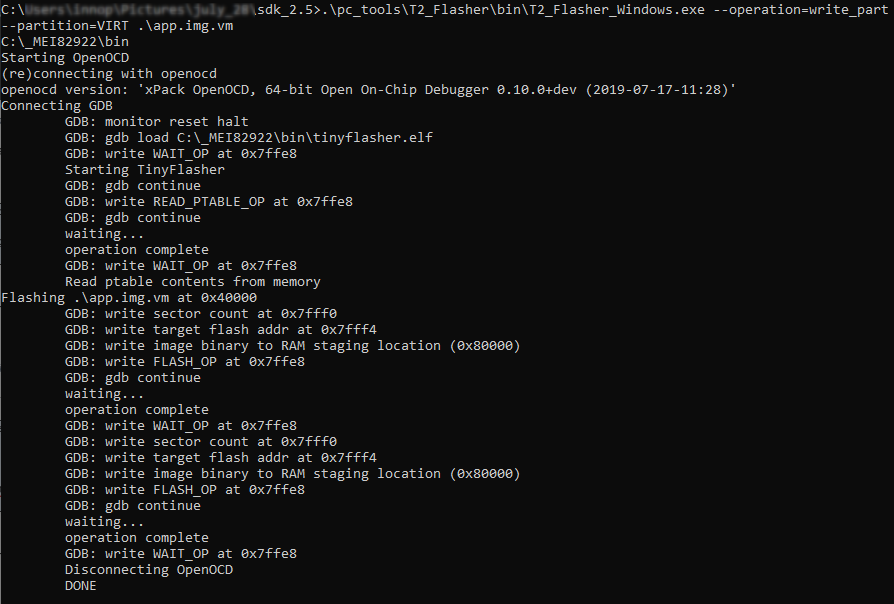


Figure 41: JTAG - Flashing application's VM image – console output

The application is successfully flashed over JTAG. Now, OpenOCD needs to be terminated before debugging using Eclipse. Close all the command prompt windows to terminate OpenOCD.

## In Linux

1. Open command prompt and reset the device in boot loader mode by executing the following command from the SDK directory:

|  |
| --- |
| ./script/reset.py evk42\_bl |

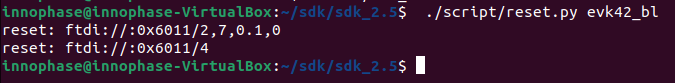


Figure 42: JTAG - Resetting the device in boot loader mode- console output (Windows)

1. Open command prompt and start OpenOCD by executing the following command from the SDK directory:

|  |
| --- |
| openocd -s ./conf -f ftdi.cfg -f t2.cfg |

The following output is seen on command prompt console:

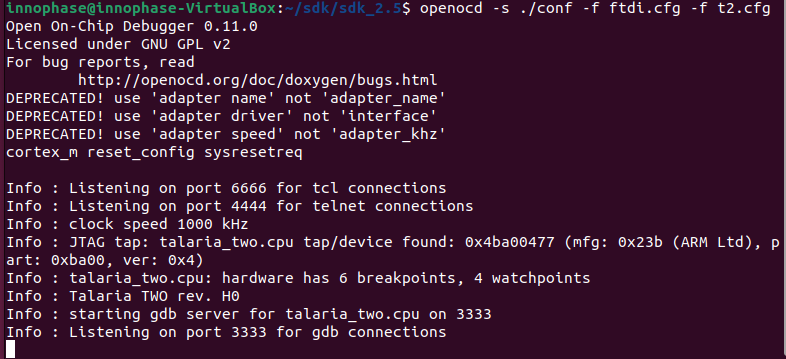


Figure 43: JTAG - Starting OpenOCD – console output

1. Flash the default partition table by executing the following command:

|  |
| --- |
| .\pc\_tools\T2\_Flasher\bin\T2\_Flasher\_Linux --operation=write\_ptable .\<path to the standard\_part\_table.json file available in sdk\_x.y/tools/partition\_files directory> |

Following output is observed after flashing the default partition table:



Figure 44: JTAG - Flashing the default partition table – console output

1. In the same terminal, execute the following command from the SDK directory to flash the data image (data.img).

**Note**: This step is needed only if the application uses any certificates.

|  |
| --- |
| .\pc\_tools\T2\_Flasher\bin\T2\_Flasher\_Linux --operation=write\_part --partition=DATA .\<path to the generated data image>\data.img |

The following output is observed after flashing the data image:

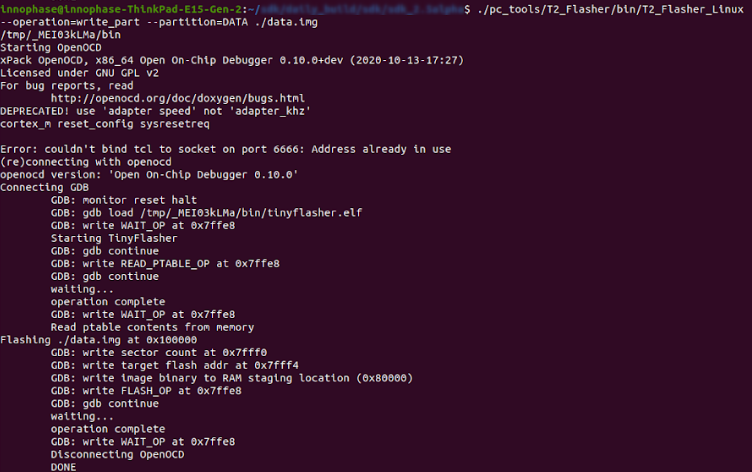


Figure 45: JTAG - Flashing the data image - console output

1. In a separate command prompt window, execute the following command from the SDK directory to flash the application image:

|  |
| --- |
| .\pc\_tools\T2\_Flasher\bin\T2\_Flasher\_Linux --operation=write\_part --partition=BOOT .\<path to the generated application image>\app.img |

The following output is observed after flashing the application image:

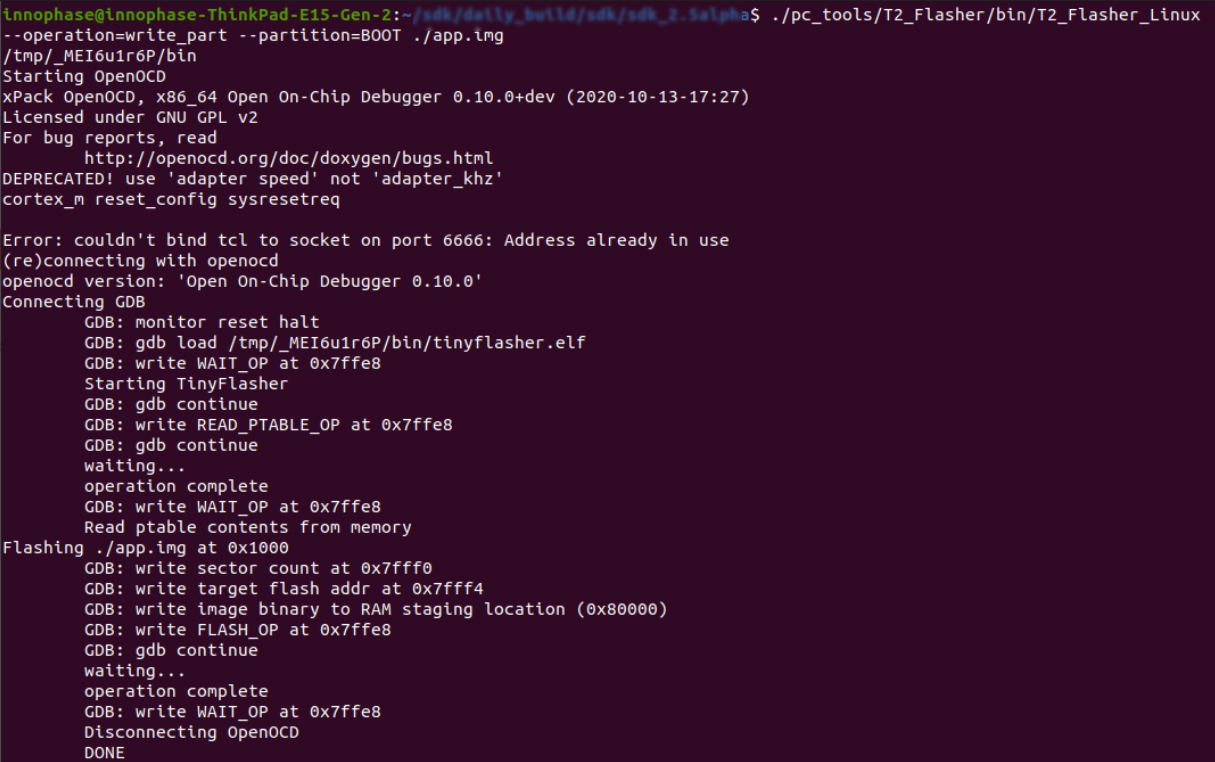


Figure 46: JTAG - Flashing the application image - console output

1. In the same terminal, flash the VM image of the application by executing the following command:

|  |
| --- |
| .\pc\_tools\T2\_Flasher\bin\T2\_Flasher\_Linux –operation=write\_part –partition=VIRT .\<path to the generated application image.vm>\app.img.vm |

Following output is observed after flashing the application’s VM image:

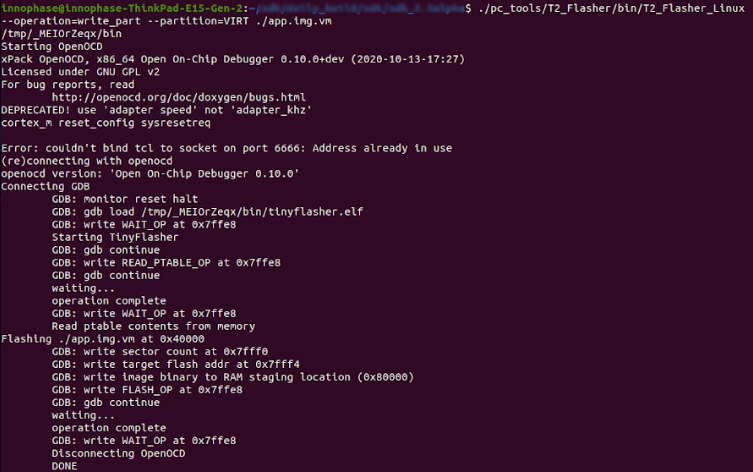


Figure 47: JTAG - Flashing application's VM image – console output

The application is successfully flashed over JTAG. Now, OpenOCD needs to be terminated before debugging using Eclipse. Close all the command prompt windows to terminate OpenOCD.