## Programming the Application

**Note**: For AT command FOTA execution, follow steps mentioned in section: *Using Script (Ubuntu only)* on Ubuntu platform.

### Using Download Tool (Windows/Ubuntu)

Program atcmds.elf using the Download tool. Launch the Download tool provided with InnoPhase Talaria TWO SDK/EVK package. In the GUI window, select the appropriate EVB from the drop-down and load the atcmds.elf. Prog RAM or Prog Flash as per requirement.

Console output:

|  |
| --- |
| Y-BOOT 208ef13 2019-07-22 12:26:54 -0500 790da1-b-7  ROM yoda-h0-rom-16-0-gd5a8e586  FLASH:PNWWWWWAE  Build $Id: git-831e563 $  Flash detected. flash.hw.uuid: 39483937-3207-00b0-0064-ffffffffffff  Application Information:  ------------------------  Name : atcmd  Version : 2.0  Build Date : Oct 2 2023  Build Time : 23:34:29  Heap Available: 268 KB (274456 Bytes)  $App:git-e608081  SDK Ver: FREERTOS\_SDK\_1.0  At Command App  addr e0:69:3a:00:16:d4  domain:1-11@20before: magic1=0x  starting thread-sock0, val=0x0, magic2=0x0  Crash detection logic initialized  after: magic1=0x11223344, val=0x0, magic2=0x55667788  Serial-to-Wireless: Ready |

The boot arguments are the configuration parameters that can specify along with the application image (atcmds.elf) during programming. Once the system boots up, it uses the boot arguments for various levels of parameter configurations.

This application allows to configure baud rate with boot argument. Value: standard UART baud rate values (9600,115200 etc.), default baudrate being 115200.

Argument for setting baudrate 115200:

|  |
| --- |
| baudrate=115200 |

### Using Script (Ubuntu only)

With FreeRTOS SDK directory as the current directory, execute the following command:

|  |
| --- |
| python3 ./script/program\_flash.py -i binaries/product/at -spt tools/partition\_files/ssbl\_part\_table.json |

where,

1. Mandatory arguments:

-i <elf\_path or elf folder>

(For example: *binaries/product/at*) in FreeRTOS SDK or complete ELF path (For example: *binaries/product/at/bin/t2\_atcmds.elf*)

1. Optional arguments:
   1. -spt <ssbl ptable>: provide the input path for ssbl\_partition\_table along with -spt in case the SSBL partition table is being considered.
   2. --no\_reset: provide the --no\_reset flag if there is no need to reset at the end. Please reset from the tool in case of this option

**Note**:

1. The mentioned script also takes care of generating the root.img in the binaries/product/at folder considering changes in at/fs contents
2. Edit the part.json file and fota\_config.json file present in *sdk\_x.y/binaries/product/at/fs* if any configuration needs to be changed before issuing this command. Each time the above command is issued, it creates a new root fs image (root.img).
3. Ensure only one EVB is connected to the PC.

After successful programming, open a miniterm at baud rate of 2457600 and reset the EVB:



Figure 1: Miniterm console output

Reset the board either by giving the following command or by pressing the reset button on the EVB:

|  |
| --- |
| ./script/boot.py --device /dev/ttyUSB2 --reset=evk42 |