Quick Start Guide

Beta Ensemble EX Development Kit

Table of Contents

Introduction

This user guide will take you through the steps of initial setup to use the Ensemble E7 Beta Development Kit for basic single-core design and will show you the steps to set up your Ensemble E7 Beta Development Kit for basic bare metal or RTOS design as well as provide detailed information on the major components, connectors, test points and configuration jumpers for both the Baseboard and the E7 CPU Board.

The Alif Semiconductor Ensemble family of Fusion Processors incorporate multiple CPU and NPU cores, large on-chip RAM and non-volatile memory, analog and digital peripherals supporting wired connectivity. The range of computational performance, power efficiency and their rich peripheral set makes them suitable for a wide range of embedded IoT applications.

The E7 Fusion Processor architecture is comprised of multiple subsystems optimized for specific roles. Each subsystem has one or two CPUs, optional NPU, memory and peripherals. Each subsystem can execute code independently of others but may share a power domain and memory and peripherals can be shared between subsystems.

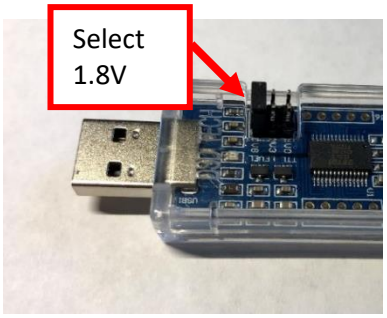
The E7 Fusion Processors have **Application Processor cores** consisting of dual-core Arm Cortex-A32 processors and **Real Time Processor Cores** consisting of Cortex-M55 processors. The Real-Time Processor cores implement the Arm v8.1 instruction set, including Helium M-Profile Vector Extension (MVE).
The Real Time Processor cores are:

- High-Performance Arm Cortex-M55 (M55-HP) operating at up to 400 MHz.
- High-Efficiency Arm Cortex-M55 (M55-HE) operating at up to 160 MHz.

Details of the E7 Fusion Processor architecture are discussed in the white paper "Fusion Processors System Architecture Introduction".

The "Kits & Software" technical documentation page will have links to the schematics and Bill Of Materials (BOM) for the Baseboard and CPU Board.

IMPORTANT – DO THIS BEFORE APPLYING POWER TO THE BOARD OR PLUGGING THE FTDI MODULE INTO A USB PORT: Set the jumper on the FTDI module for 1.8V TTL levels as shown in the image on the right. This must be done before plugging the FTDI module in a USB port or damage may result to the CPU board. The jumper must be on the pair of pins closest to the USB connector.



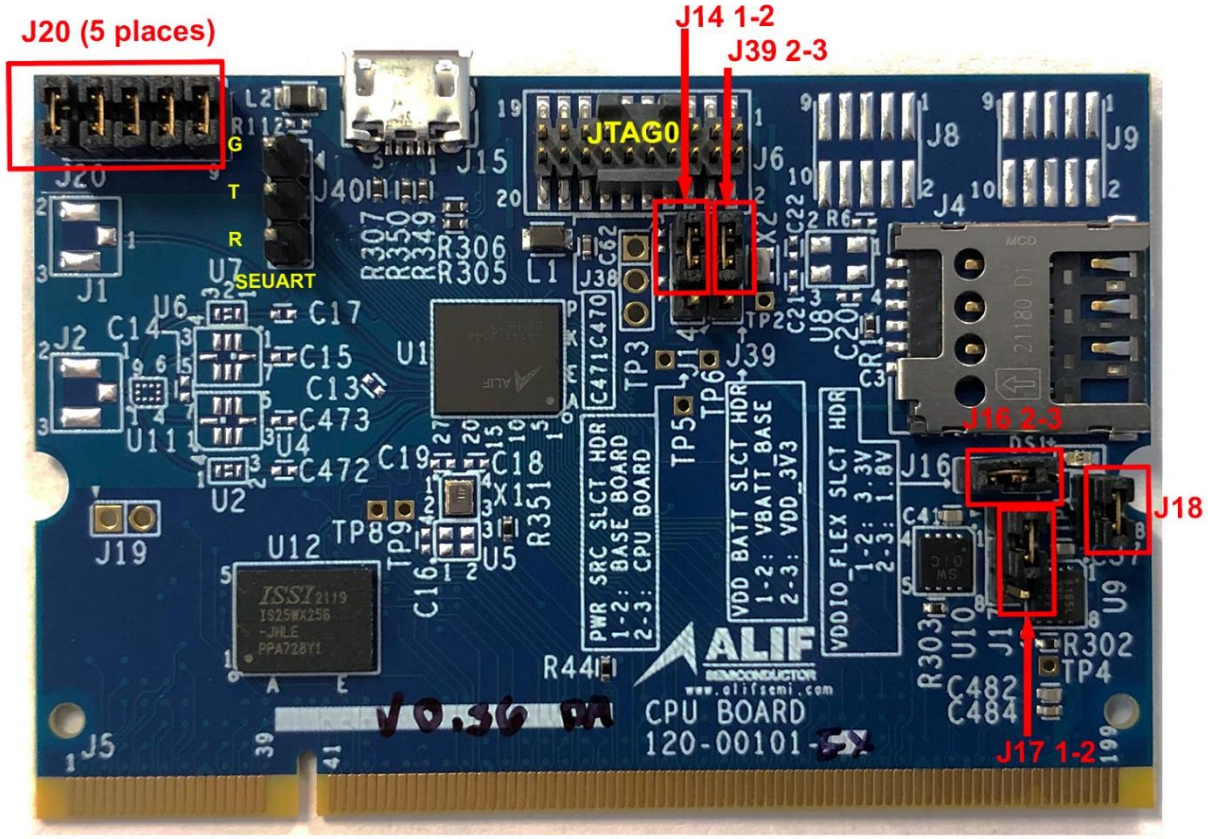
Target Setup Requirements

- DK-E7-BNDL-A1 Ensemble E7 Beta Development Kit (DevKit) with Baseboard + CPU Board
- FTDI UART Cable (included in the kit)
- Micro-USB Cable (included in the kit)
- Windows PC with terminal emulation software

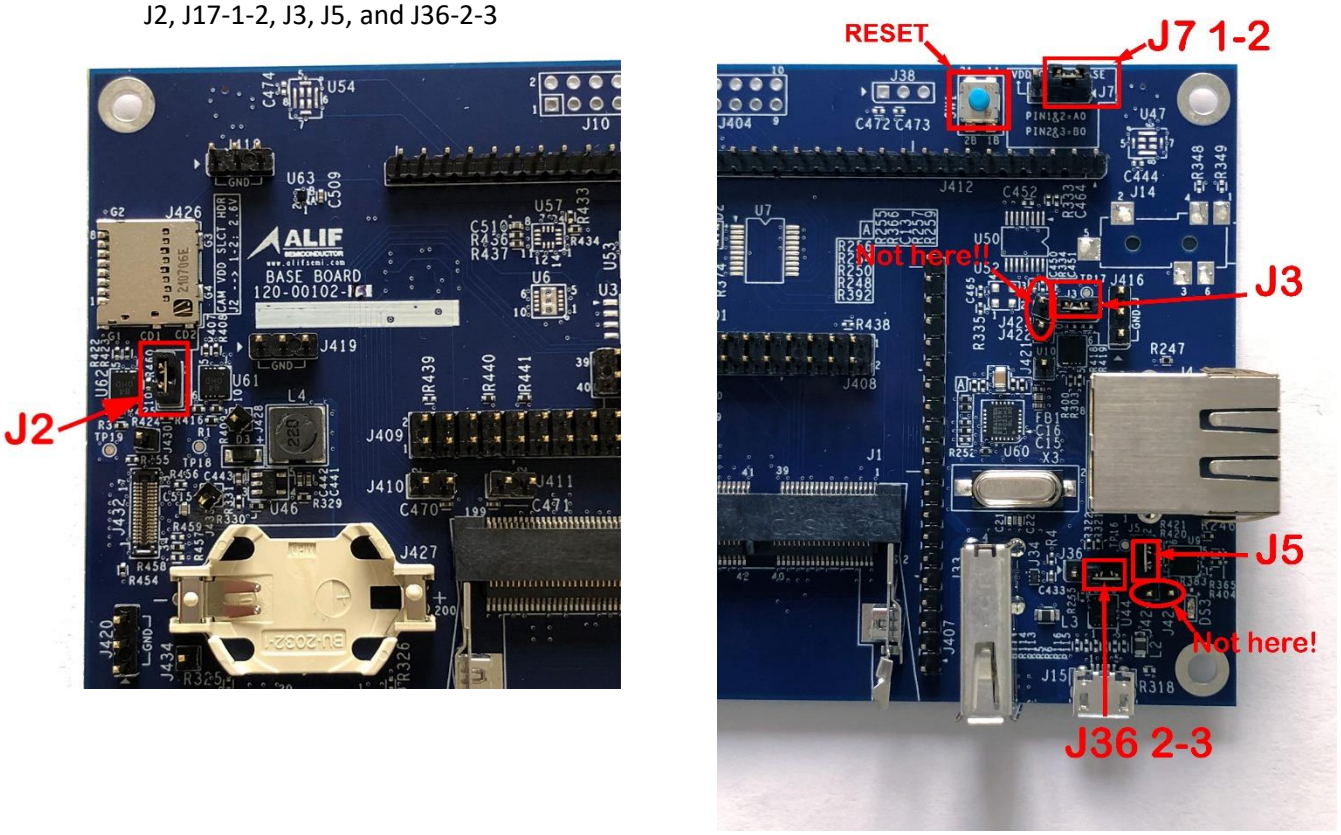
CPU Board and Baseboard Jumper Settings

CPU Board Jumper Settings:

Verify that jumpers are installed on the CPU Board at the following locations:
J14-1-2, J39-2-3, J16-2-3, J18, J17-1-2, and J20(5 places) as shown below



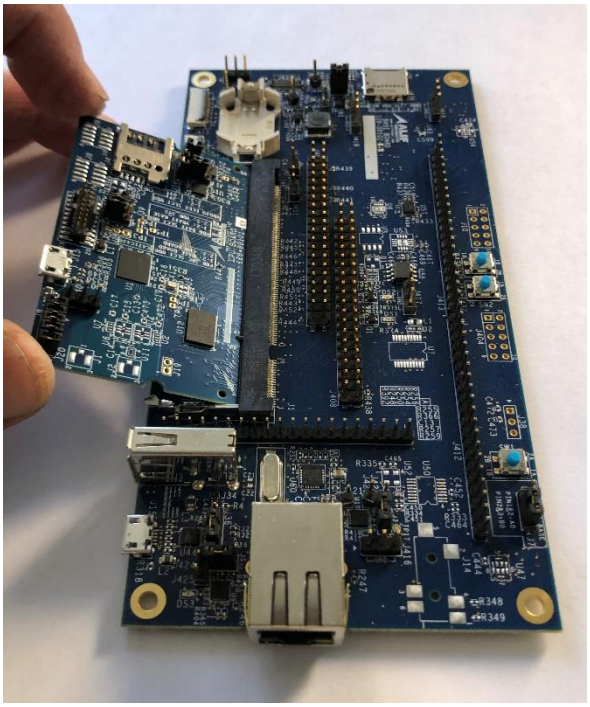
Baseboard Jumper Settings:
Verify that jumpers are installed on the Baseboard at the following locations:
J2, J17-1-2, J3, J5, and J36-2-3



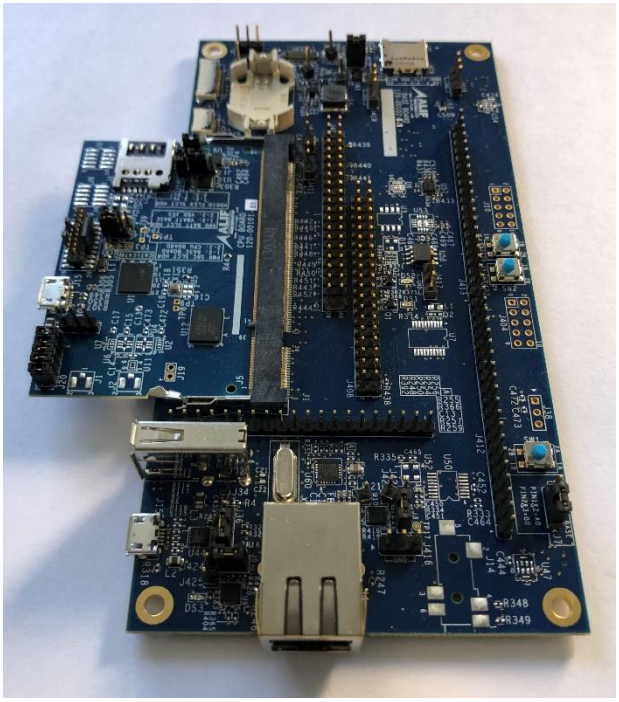
Assembling the CPU Board to the Baseboard

To assemble the CPU Board to the Baseboard:

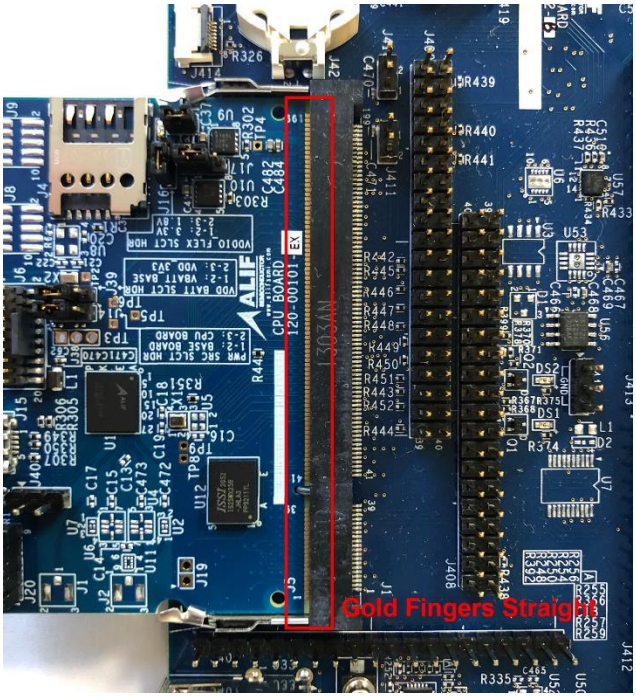
- **BE SURE THAT POWER IS OFF!!! (No MicroUSB power cord connected)**
- Insert DUT CPU Board into SODIMM Connector on Baseboard.
- Snap down mated boards flat.
- Gold fingers along the SODIMM connector should be straight.



Insert CPU Board into Baseboard at a 45° angle



CPU Board should snap down onto Baseboard as shown

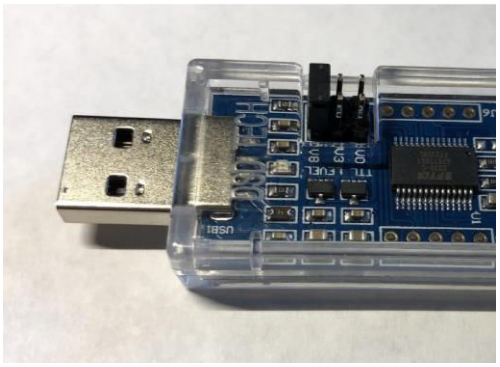


Gold fingers along the SODIMM connector should be straight

Connecting the FTDI Module

CAUTION

DO THIS BEFORE PLUGGING THE FTDI MODULE INTO A USB PORT: Set the jumper on the FTDI module for 1.8 V TTL levels, as shown in the image to the right. This must be done before plugging the FTDI module in a USB port or damage may result to the CPU Board. The jumper must be on the pair of pins closest to the USB connector.

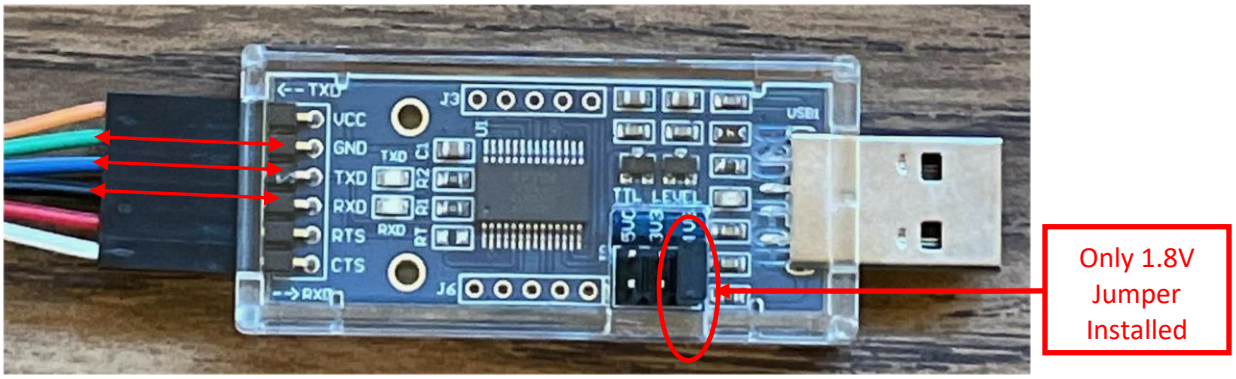


FTDI Module for 1.8 V TTL Levels

There may be a shorting jumper between the center two pins where the flat connector plugs into the module. This creates a loop-back connection in the absence of the cable. Remove this jumper.

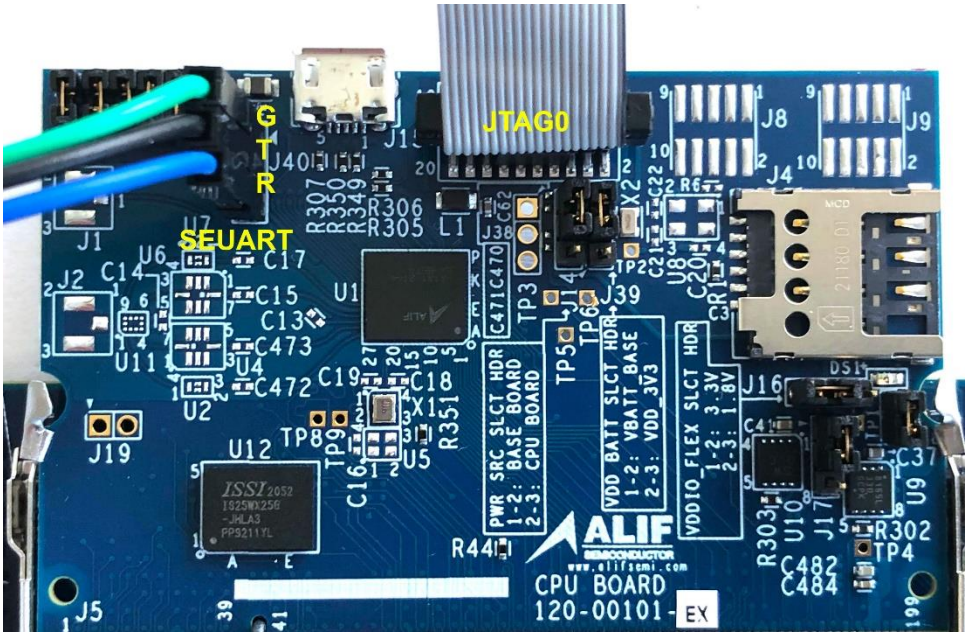
Connect the flat connector to the module with the green wire (if present on the connector) lined up with the GND designation on the circuit board as shown in the picture above.

- Note wire color for TXD, RXD and GND connections on the DTFI UART adapter.
  - **Blue** wire in this picture is **TXD**, color can be different depending on FTDI adapter used.
  - **Black** wire in this picture is **RXD**, color can be different depending on FTDI adapter used.
  - **Green** wire in this picture is **GND**, color can be different depending on FTDI adapter used.
- There are no connections needed for VCC, RTS and CTS wires
- Verify that only the 1.8V Jumper is installed.



Connect the FTDI UART cable between the host PC and the CPU Board SE-UART connection using the following SEUART pins:

- J40 pin 1 (GND): Connects to GND (green wire or as noted above) on the FTDI cable
- J40 pin 2 (SEUART_TX): Connects to RXDATA (black wire or as noted above) on the FTDI cable
- J40 pin 3 (SEUART_RX): Connects to TXDATA (blue wire or as noted above) on the FTDI cable



Verifying FTDI Driver

Plug the FTDI module into a free USB port on the host PC. Before trying to create a debug connection, verify that the proper driver for the FTDI module is installed.

To do this, go to the Windows 10 *Control Panel* and search for *Device Manager*. Click the link to *Device Manager* and scroll down to the *Ports (COM & LPT)* section. A device entry for a USB Serial Port should be seen. If there is a yellow explanation point symbol there, the FTDI driver likely did not load correctly.

In this case, go to the Alif Semiconductor Kits and Software technical documentation page where you will find the FTDI setup executable *.exe* file. Download this and follow the instructions in the installation guide file.

Verifying FTDI Connection

In order to verify the SE-UART connection, a terminal emulator program such as *Tera Term* or *Picocom* is needed. If such is not available, locate and install one before proceeding.

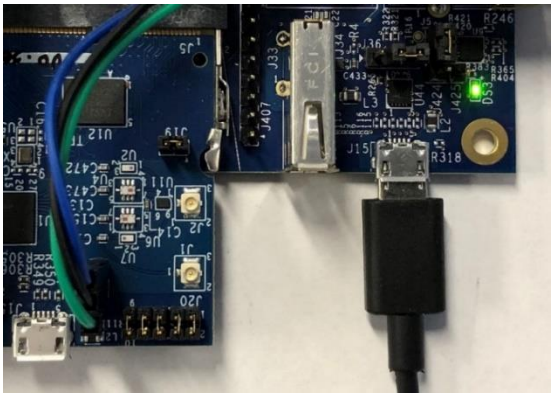
Alif recommends:

For Windows: Tera Term (or similar)
For Linux: Picocom (https://github.com/npat-efault/picocom) (or similar)

Start the terminal emulator and select the appropriate COM port for the FTDI interface. Then configure the terminal for the following settings. Since beta development kits being shipped were provisioned with either v38 or v42 firmware with very few exceptions, the setting below reflect those firmware revisions.

- 115200 bps baud rate NOTE: use 100000 bps baud rate for SETOOLS v53 or later
- 8 data bits
- No parity
- One stop bit
- No flow control
- New line receive AUTO
- New line transmit CR

To power the Baseboard and CPU board, connect a micro-USB to the Baseboard as shown below. The other end of this cable can be connected to a PC USB port or a 5 W USB power adapter such as a phone charging adapter.



The application software programmed into the Ensemble device at the factory should blink the LEDs on the Baseboard every second and you will see a printout on the SEUART terminal screen detailing the status of what is loaded and running in each CPU core as shown below.

```
SES A1 EVALUATION_BOARD RELEASE v0.38.0 Dec 15 2021 22:14:41
[SES] Device ID = A100
[SES] PLL code version 0.0.3
[SES] LCS=1
[SES] System TOC address 0xA0580000
[SES] Wounding Data: 0x00000000
[SES] System      TOC is processed (ret=0x00000000) BL_STATUS_OK
[SES] Application TOC is processed (ret=0x00000000) BL_STATUS_OK

FC:Rgn - 7:1 7:2 7:3 7:4 7:5 8:1 8:2 8:3 8:4 8:5 13:0 13:1 13:2
Protected areas:
        0x80580000 - 0x805FFFFF

+----------+--------+------------+------------+------------+------------+----------+-------+--------+-----------+
|   Name   |  CPU   | Store Addr |  Obj Addr  | Dest Addr  | Boot Addr  |   Size   |Version| Flags  | Time (ms) |
+----------+--------+------------+------------+------------+------------+----------+-------+--------+-----------+
|   SERAM0 |  CM0+  | ---------- | 0x00000120 | ---------- | ---------- |    56256 | 1.0.0 | u s    |      0.00 |
|   SERAM1 |  CM0+  | ---------- | 0x00020B20 | ---------- | ---------- |    56256 | 1.0.0 | u s    |      0.00 |
|   DEVICE |  CM0+  | 0x805C1F20 | 0x805C1520 | ---------- | ---------- |      728 | 0.5.5 | u V    |     10.45 |
| A32_DBG  | A32_0  | 0x805C2C00 | 0x805C2200 | 0x02000000 | 0x02000000 |      644 | 1.0.0 | uLVB   |      9.71 |
|  HP_DBG  | M55-HP | 0x805C3890 | 0x805C2E90 | 0x50000000 | 0x50000000 |     2256 | 1.0.0 | uLVB   |      9.97 |
|  HE_DBG  | M55-HE | ---------- | 0x805C4160 | ---------- | ---------- |     2256 | 9.9.9 | u s    |      0.00 |
| BLINK-HE | M55-HE | 0x8057EC90 | 0x8057E290 | 0x60000000 | 0x60000000 |     4912 | 1.0.0 | uLVB   |     10.43 |
+----------+--------+------------+------------+------------+------------+----------+-------+--------+-----------+
Legend: (u)(C)ompressed,(L)oaded,(V)erified,(s)kipped verification,(B)ooted,(E)ncrypted,(D)eferred

[SES] CM0+ frequency is 100 MHz
[SES] os Kernel V10.4.2
[SES] Main Task - looping forever...
```

NOTE: If the FTDI adapter is not connected to the SEUART connections on the CPU board, the board will not boot properly, and the LEDs will not blink.

If the output is not readable at 115200 baud, try 100000 baud. If that works, you have firmware v53 or later and should follow the rest of the instructions accordingly.

You must connect the FTDI adapter to the SE-UART, verify the connection and board are working as shown above, and then proceed to install the Alif Security Toolkit (SETOOLS) and update the system firmware image as described below.

Install Alif Security Toolkit

The next step is to install the latest release of the Alif Security Toolkit (SETOOLS). Refer to the Alif Security Toolkit Quick Start Guide [click here] for installation instructions.

Note that SETOOLS version 53 and later are now executable images, previously these python scripts were delivered as source code. These executables do not require python or any pre-requisite libraries to be installed.
Previously you were required to run

```
$ python3 <tool-name>.py
```

Now you just run

```
$ <tool-name>
$ ./<tool-name>
```

Update System Firmware Image

IMPORTANT: Before beginning design, you should update the internal system firmware on the CPU board as detailed in this section. Begin by loading the latest version of the Alif Security Toolkit obtained from the Alif website, then update the system image using the *updateSystemPackage* command as detailed below.

ISP Discovery

The first time you execute one of the SETOOLS scripts you will be prompted for the required serial port. When the ports are presented, just enter the port name and press [ENTER].

This port data is saved in a local configuration file (isp_config_data.cfg). The next time a command is invoked and this configuration file is present, it will use the parameters from this file.

To override this option simply use the -d option:

```
c:\app-release-exec>maintenance -d
COM ports detected = 1
-> COM5
Enter port name:
[INFO] COM5 open Serial port success
[INFO] baud rate 100000

Available options:
```

This will force a re-discovery of the Serial ports.

UART Errors

There is only one SE-UART on the device. If you are using ISP, please ensure you have no other Tera term or putty sessions using the same SE-UART. The following shows the output if the SE-UART is already being used by another program:

```
c:\app-release-exec>app-write-mram
Writing MRAM with parameters:
Device Part# E7 (AE722F80F55D5AE) - 5.5 MRAM / 13.5 SRAM - Rev: A1
- Available MRAM: 5767168 bytes
[INFO] Burning: ../build/AppTocPackage.bin 0x8057e290
[INFO] baud rate  100000
[INFO] dynamic baud rate change  Enabled
[ERROR] openSerial could not open port 'COM5': PermissionError(13, 'Access is denied.', None, 5)
[ERROR] isp openSerial failed for COM5

c:\app-release-exec>
```

This indicates that the UART is already being used (e.g., a Tera-Term session is still running).

updateSystemPackage

With the SEUART connected to your PC, open a Windows command prompt and navigate to the SETOOLS release directory.

If your board is running a previous version of the SETOOLS (version 52 or earlier), the command is:

updateSystemPackage -b 115200 -nr

NOTE: Most development kits being shipped have either version 38 or 42 SETOOLS and will require this command format. You need to specify this baud rate option as releases before Version 53 are using a different baud rate from the default 100000 baud rate of v56.

If you are updating a firmware image version 53 or later, the command is:

updateSystemPackage

Before running this utility, be sure the CPU board is powered on and the SEUART is connected to the PC and there is no TeraTerm session open. Open Windows Command prompt and change the directory to the release directory. Call the program as the example shows. In this example, we were updating from SETOOLS v38 so we specified the baud rate to be 115200 bps.

```
C:\app-release-exec>updateSystemPackage -b 115200 -nr
Burning: System Package in MRAM
Device Part# E7 (AE722F80F55D5AE) - 5.5 MRAM / 13.5 SRAM - Rev: A1
- MRAM Base Address: 0x80580000


[INFO] baud rate  115200
[INFO] dynamic baud rate change  Disabled
[INFO] COM3 open Serial port success
Maintenance Mode = Disabled
Authenticate Image:  True
Signature File:  alif\SystemPackage.bin.sign
Auth Token:  0x4156bb1
Verify Image
alif\SystemPackage.bin          [####################]100%: 275536/275536 bytes
Authenticate Image:  True
Signature File:  alif\offset.bin.sign
Auth Token:  0x4e96fe
Verify Image
alif\offset.bin                 [####################]100%: 16/16 bytes

C:\app-release-exec>cd ..
```

NOTE: When running updateSystemPackage, if you get a message “error= ISP_UNKNOWN_COMMAND”, it is likely that you did not specify the “-nr” option in the command.
If you get a message “[ERROR] Target did not respond”, try pressing the [RESET] button in the upper right corner of the board and retry the updateSystemPackage command.
If you get a message “ISP_BAD_DEST_ADDRESS”, you can ignore it. The update should complete without errors.

You have now updated your internal system image and can proceed with design and debug.

A complete step-by-step tutorial on doing bare metal design is available in the documents
“Getting Started With Bare Metal Design Using Arm DS User Guide” [click here] or
“Getting Started with GNU/VSCode/J-Link User Guide” [click here]
which are available on the Application Noes & User Guides Technical Documentation page.

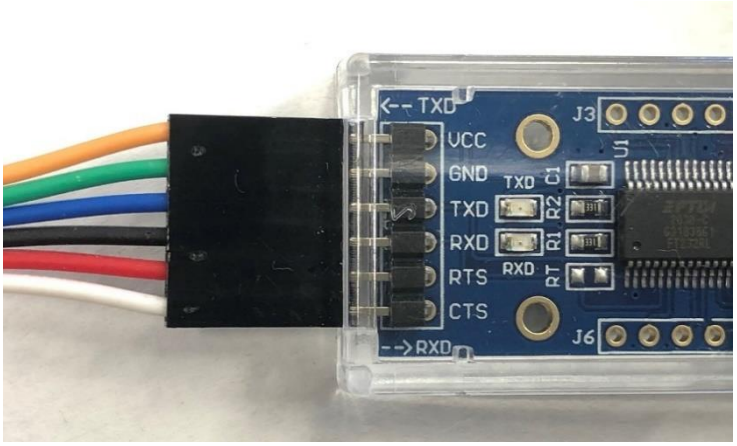
Optional UART Connections

If an example application you are going to use specifies the need for additional UART connections, the instructions below will show you how to connect additional FTDI adapters to UART2, UART4, or UART6.

You should have connected an FTDI adapter to the SEUART as part of the initial board setup.

IMPORTANT – DO THIS BEFORE APPLYING POWER TO THE BOARD OR PLUGGING THE FTDI MODULE INTO A USB PORT: Set the jumper on the FTDI module for 1.8V TTL levels as shown in the image on the right. This must be done before plugging the FTDI module in a USB port of damage may result to the CPU board. The jumper must be on the pair of pins closest to the USB connector.

NOTE: There can be variations in the colors of cables supplied with the adapter. The examples below use a cable connected to the adapter as shown in the photo to the right. Check the color of the wires connected to various FTDI signals as identified by the signal names silkscreened on the adapter PCB.

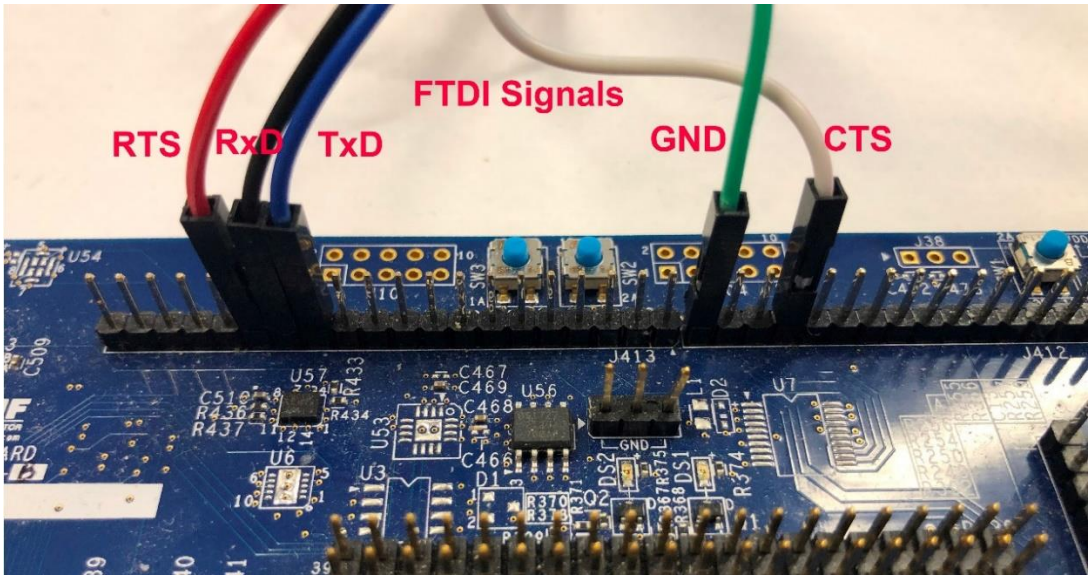


UART2 Connections

Connect the FTDI UART cable between the host PC and the CPU Board UART2 connection using the following UART2 pins:

J413 pin 13 (UART2_RX): Connects to TXDATA (blue wire or as noted above) on the FTDI cable
J413 pin 14 (UART2_TX): Connects to RXDATA (black wire or as noted above) on the FTDI cable
J413 pin 15 (RTS): Connects to RTS (red wire or as noted above
J412 pin 20 (GND): Connects to GND (green wire or as noted above) on the FTDI cable
J412 pin 17 (CTS): Connects to CTS (white wire or as noted above) on the FTDI cable

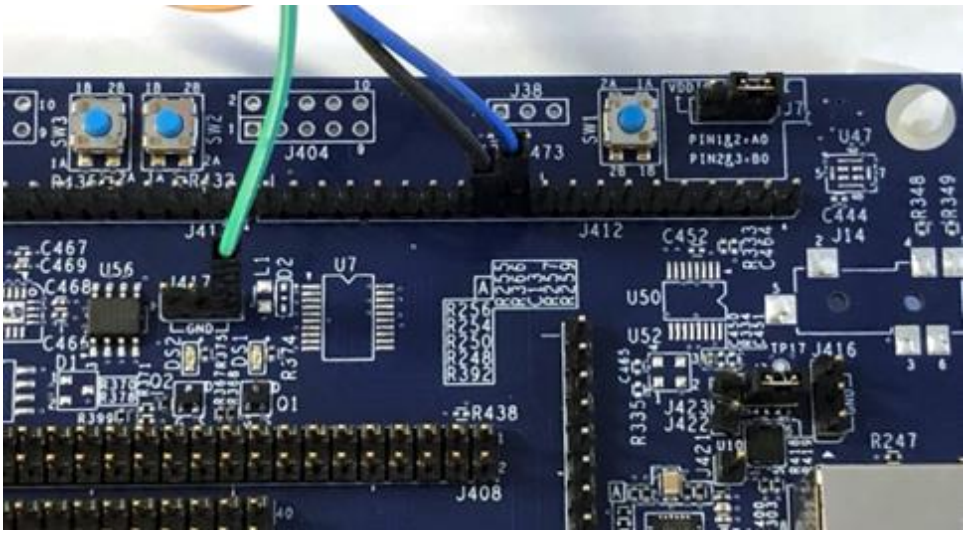
NOTE: RTS and CTS connections are required for the Linux example applications for programming the on-board OSPI flash memory. They can be omitted for applications that do not require them.



UART4 Connections

Connect the FTDI UART cable between the host PC and the CPU Board UART4 connection using the following UART4 pins:

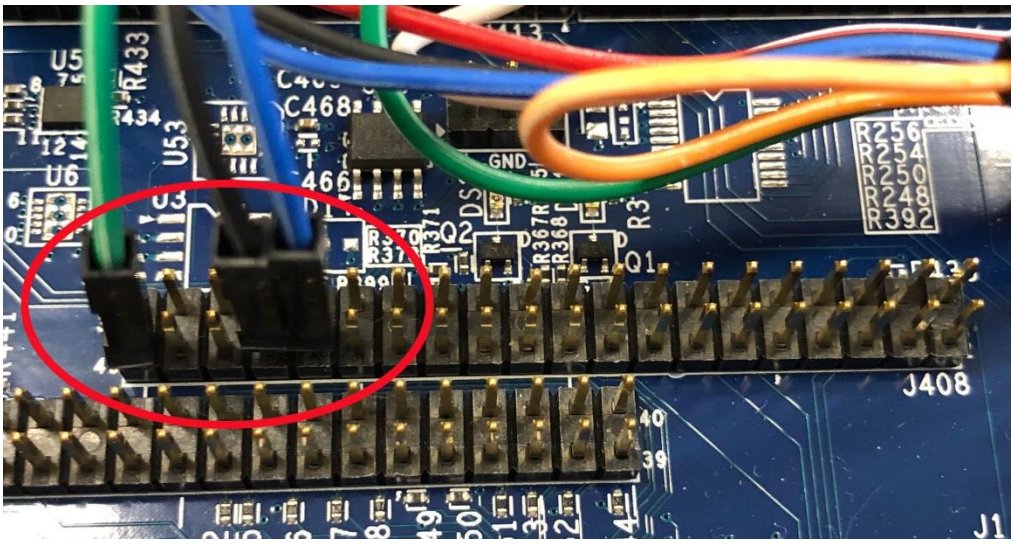
J412 pin 11 (UART4_RX): Connects to TXDATA (blue wire or as noted above) on the FTDI cable
J412 pin 12 (UART4_TX): Connects to RXDATA (black wire or as noted above) on the FTDI cable
J417 (any pin) (GND): Connects to GND (green wire or as noted above) on the FTDI cable



UART6 Connections

Connect the FTDI UART cable between the host PC and the CPU Board UART6 connection using the following UART6 pins:

J408 pin 32 (UART4_RX): Connects to TXDATA (blue wire or as noted above) on the FTDI cable
J408 pin 34 (UART4_TX): Connects to RXDATA (black wire or as noted above) on the FTDI cable
J408 pin 40 (GND): Connects to GND (green wire or as noted above) on the FTDI cable



Document History

Version	Change Log
1.0	Initial release for Beta program
1.1	Added links to supporting documents
1.2	Added “-a” option to the updateSystemPackage.py command
1.3	Revised to reflect SETOOLS v54 based on executables
1.4	Added instructions for optional UART2, UART4, or UART6 connections
1.5	Revised command syntax for updateSystemPackage to address compatibility with older chip firmware revisions (v38 and v42).
1.6	Minor edits to introduction