AN 043: Using the 10/100/1000 Mbps TSE MAC with Sapphire SoC on lwIP

AN043-v1.3
March 2024
www.efinixinc.com

Contents

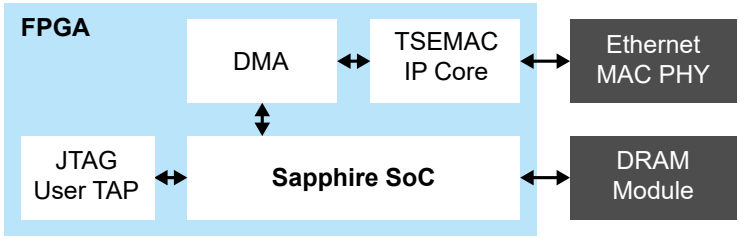
Introduction

The 10/100/1000 Mbps TSE MAC with Sapphire SoC on lwIP example design illustrates how to use a RISC-V processor to control the Triple Speed Ethernet MAC (TSEMAC) IP core and transfer data through direct memory access (DMA) in Titanium FPGAs. The design supports 10/100/1000 Mbps Ethernet speed and targets the:

- Titanium Ti60 F225 Development Board
- Titanium Ti180 M484 Development Board
- Titanium Ti180 J484 Development Board
- Trion T120 BGA324 Development Board
- Trion T120 BGA576 Development Board

The design also includes lwIP application that lets you establish an iPerf server to test the overall bandwidth. lwIP is a small independent implementation of the TCP/IP protocol suite that provides a lightweight TCP/IP stack which is an ideal solution for microcontrollers.

Figure 1: 10/100/1000 Mbps TSE MAC with Sapphire SoC on lwIP Example Design Block Diagram



FPGA Support

The example design supports Titanium Ti60 and Ti180 FPGAs.

Resource Utilization and Performance

FPGA	Logic and Adders	Flip-flops	Memory Blocks	DSP Blocks	f_{MAX} (MHz)	Language	Efinity Version
Ti60 F225 C4	17,958	15,117	176	4	TSEMAC: 245 SoC: 289	Verilog HDL	2023.1
Ti180 M484 C4	24,224	28,669	284	4	TSEMAC: 245 SoC: 289	Verilog HDL	2023.1
Ti180 J484 C4	24,369	28,668	284	4	TSEMAC: 233 SoC: 241	Verilog HDL	2023.1
T120 F324	15,678	14,223	204	4	TSEMAC: 134 SoC: 63	Verilog HDL	2023.1
T120 F576	15,588	14,215	204	4	TSEMAC: 134 SoC: 65	Verilog HDL	2023.1

Required Hardware and Software

Hardware Requirements

Table 1: Required Hardware for Each Board Example Design

Example Design	Required Hardware	
	Included in the Development Kit	**Self-Provided**
Titanium Ti60 F225	• Titanium Ti60 F225 Development Board • USB cable • Board power adapter	• Ethernet cable • Computer with Ethernet port • Ethernet Connector Daughter Card
Titanium Ti180 M484	• Titanium Ti180 M484 Development Board • USB cable • Board power adapter • Ethernet Connector Daughter Card • FMC-to-QSE Adapter Card	• Ethernet cable • Computer with Ethernet port
Titanium Ti180 J484	• Titanium Ti180 J484 Development Board • USB cable • Board power adapter • Ethernet Connector Daughter Card • FMC-to-QSE Adapter Card	• Ethernet cable • Computer with Ethernet port
Trion T120 BGA324	• Trion T120 BGA324 Development Board • USB cable • Board power adapter	• Ethernet cable • Computer with Ethernet port • USB-to-UART module
Trion T120 BGA576	• Trion T120 BGA576 Development Board • USB cable • Board power adapter	• Ethernet cable • Computer with Ethernet port • USB-to-UART module

Software Requirements

The example design uses the following software:

- Efinity® software version 2023.1
- Efinity® RISC-V Embedded Software IDE
- iPerf2 network speed test tool (https://iperf.fr)

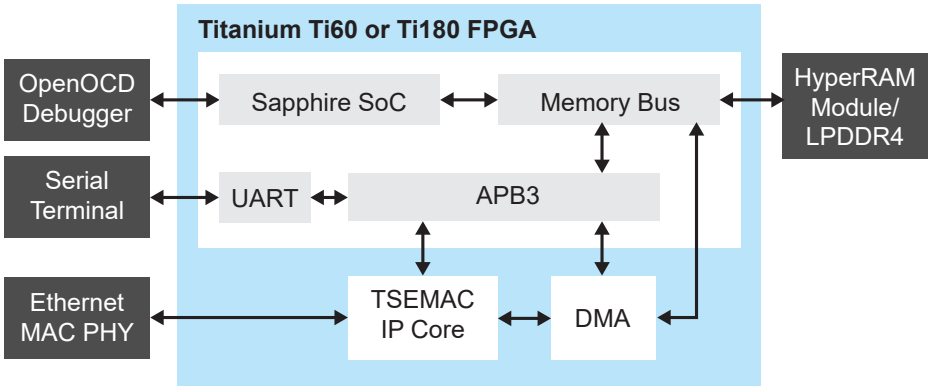
Note: Refer to the **Sapphire RISC-V SoC Hardware and Software User Guide** for instructions on installing the RISC-V IDE.

Functional Description

The 10/100/1000 Mbps TSE MAC with Sapphire SoC on lwIP example design combines the Sapphire SoC with a DMA block and TSEMAC IP core. The DMA block has a memory-mapped interface that connects to the SoC's memory bus. A streaming interface connects the DMA block to the TSEMAC IP core, which transfers bulk data from/to the TSEMAC without needing to access the RISC-V processor. This arrangement gives the processor more bandwidth to complete other operations. The TSEMAC IP core transmits or receives Ethernet packets to/ from the host and ensures the data transmission over Ethernet meets the media access rules specified in the 802.3-2008 IEEE standard. The RISC-V processor controls the DMA block and TSEMAC IP core through the APB3 bus.

With the UART peripheral, you can use a terminal to display message output. Messages print when the TSEMAC successfully connects to a host or there is packet activity from the host to the FPGA that is requested by a software application, or vice-versa.

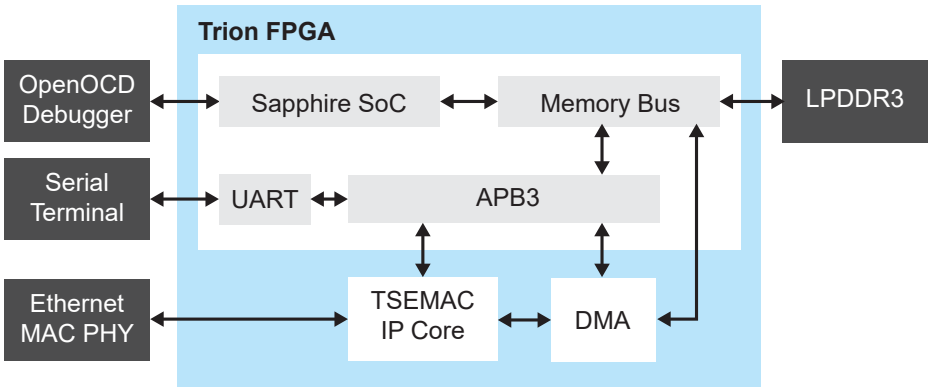
Figure 2: Titanium Sapphire SoC Peripherals



Note:

- Ti60 FPGA uses HyperRAM for the external memory.
- Ti180 FPGA uses LPDDR4 for the external memory.

Figure 3: Trion Sapphire SoC Peripherals



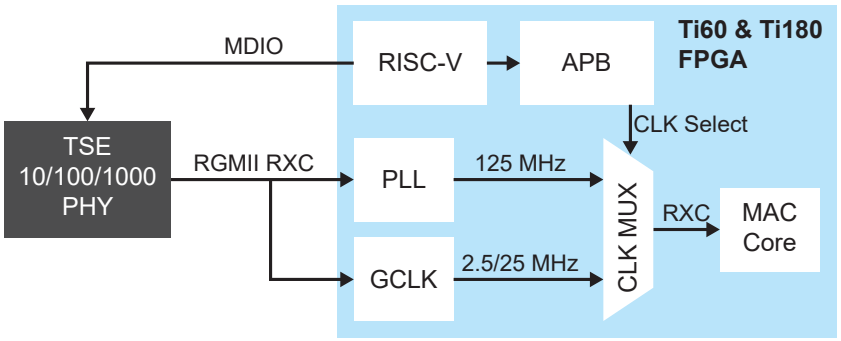
Note: The instructions in this example design assume that you are familiar with the Sapphire SoC and the RISC-V IDE.

RGMII Dynamic Clock Switch (Titanium only)

The 10/100/1000 Mbps TSE MAC with Sapphire SoC on lwIP design example includes an RGMII dynamic clock switch to ensure the interface uses the supported Ethernet speed. The RISC-V detects the PHY chip with auto-negotiation function. Then, the RISC-V sets the clock mux and MAC settings to use either 125 MHz (1000 bps) or 25 MHz (100 bps) / 2.5 MHz (10 bps).

Learn more: For more information about the clock mux connection selection, refer to the Clock and Control Network section of either Ti60 Data Sheet or Ti180 Data Sheet.

Figure 4: RGMII Dynamic Clock Switch Block Diagram

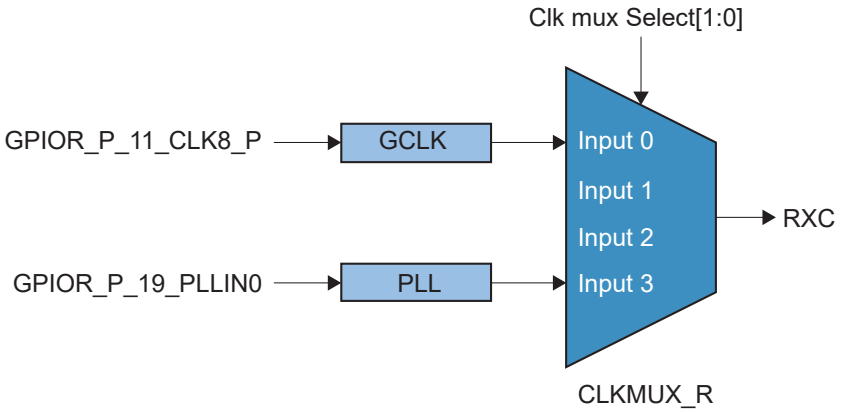


You need to assign a pin accordingly for the dynamic clock switch selection to work:

Before designing a new circuit, review the datasheet for the dedicated pins for corresponding dynamic clock mux. For the dummy clock mux requirement, input 0 must be assigned to the clock source for mux operation. For Ti60 design example, slow speed clock RXC (2.5/25 MHz) is assigned to input 0 of the dynamic clock mux. For Ti180 design example, input 0 is assigned an extra dummy clock signal for the mux operation.

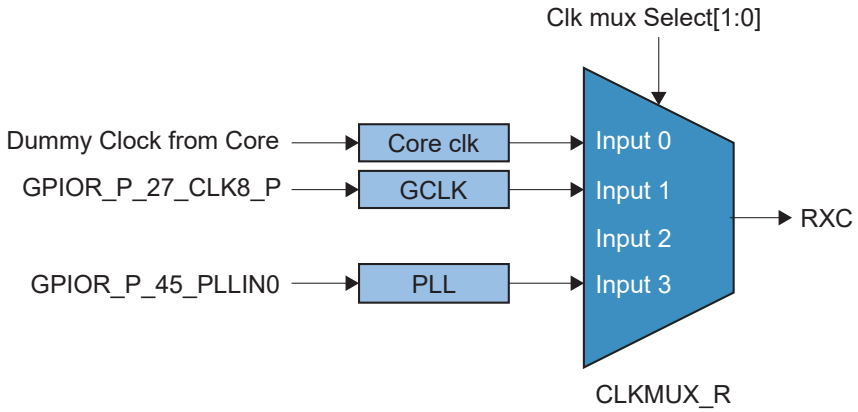
The Ti60 F225 design example uses `GPIOR_P_11_CLK8_P` for GCLK in and `GPIOR_P_19_PLLIN0` for PLL in. Both are accessible through the P1 header on the Titanium Ti60 F225 Development Board.

Figure 5: Clock Mux for Titanium Ti60 F225 Design Example



The Ti180 M484 design example uses `GPIOR_P_27_CLK8_P` for GCLK in and `GPIOR_P_45_PLLIN0` for PLL in. Both are accessible through the P1 header on the Titanium Ti180 M484 Development Board.

Figure 6: Clock Mux for Titanium Ti180 M484 Design Example

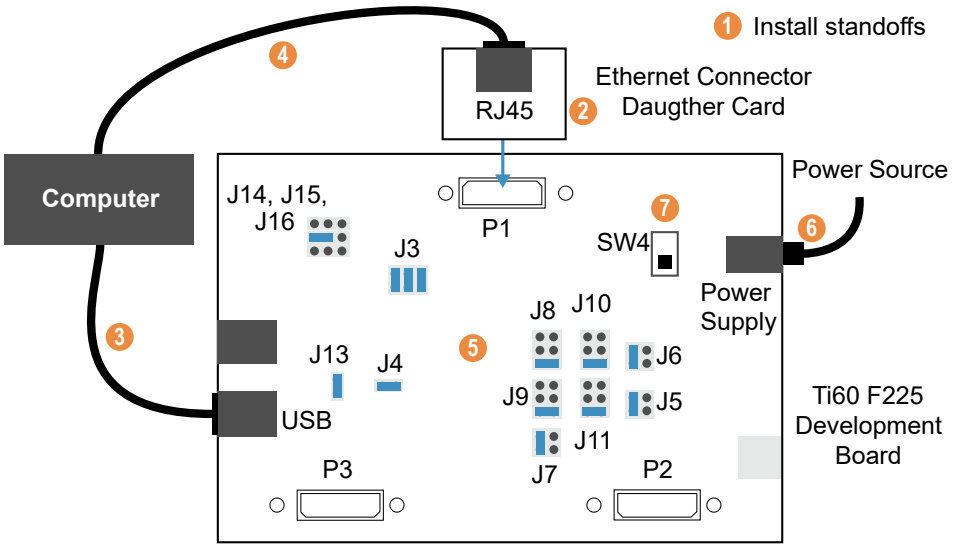


Set Up the Hardware

Titanium Ti60 F225 Development Board

Important: Make sure that the power supply and the board power switch are turned off before connecting any cables.

Figure 7: Hardware Setup for Titanium Ti60 F225 Development Board



1. If you have not already done so, attach standoffs to the board.
2. Connect the Ethernet Connector Daughter Card to header P1 of the Titanium Ti60 F225 Development Board.
3. Connect a USB cable to the board and to your computer.
4. Connect an Ethernet cable to the RJ45 jack on the board and to your computer.
5. Connect the following jumpers on the Titanium Ti60 F225 Development Board:

Header	Pins to Connect
J3	1 - 2 3 - 4 5 - 6
J4, J5, J6, J7, J8, J9, J10, J11, J12, J13, J15	1 - 2
J14, J16	Unconnected

6. Connect the 12 V power supply to the board connector and to a power source.
7. Turn on the board using the power switch.

Titanium Ti180 M484 and J484 Development Board

Important: Make sure that the power supply and the board power switch are turned off before connecting any cables.

Figure 8: Hardware Setup for Titanium Ti180 M484 Development Board

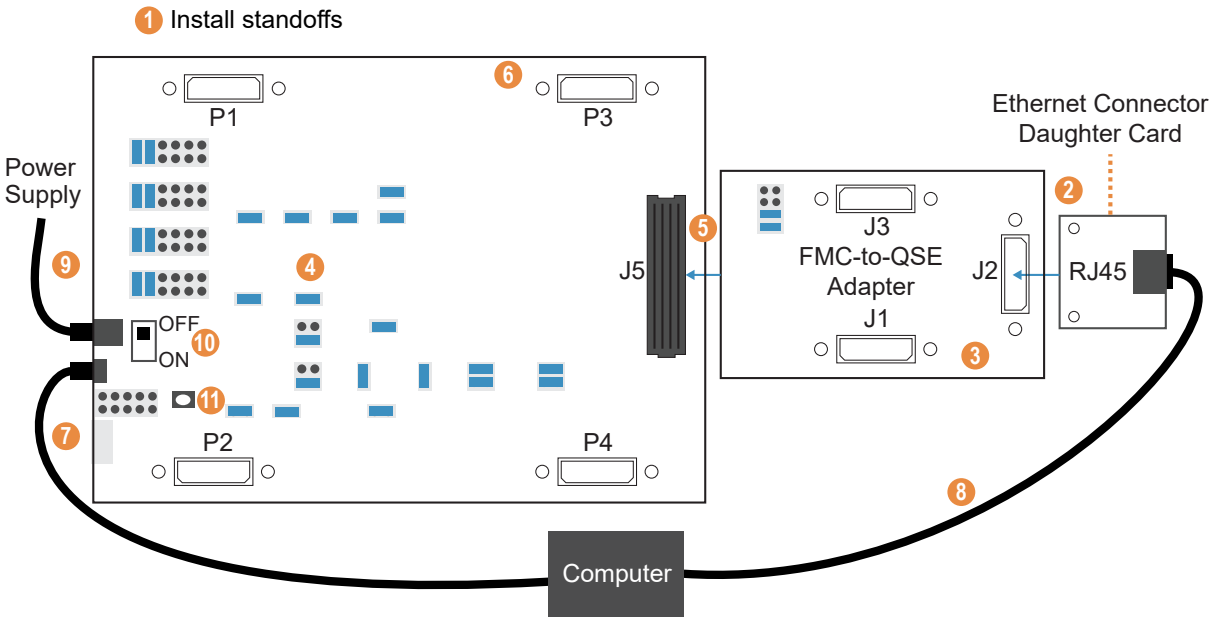
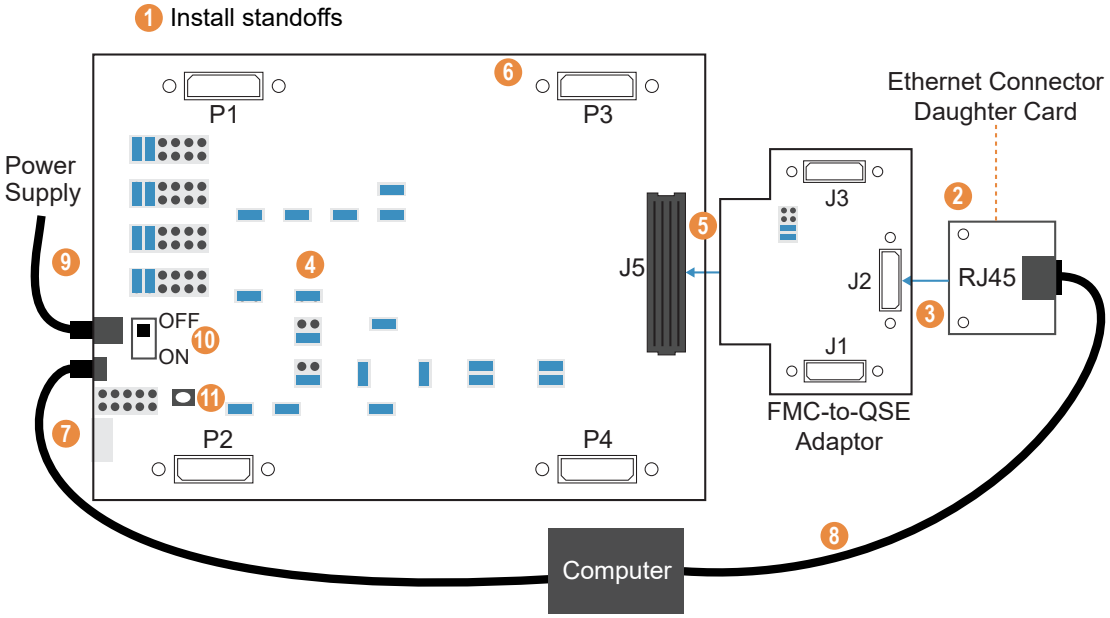


Figure 9: Hardware Setup for Titanium Ti180 J484 Development Board



1. If you have not already done so, attach standoffs to the board.
2. Connect the Ethernet Connector Daughter Card to the QSE connector (J2) on the FMC-to-QSE Adapter Card.

3. Install the nuts on the FMC-to-QSE Adapter Card.
4. Connect the following jumpers:

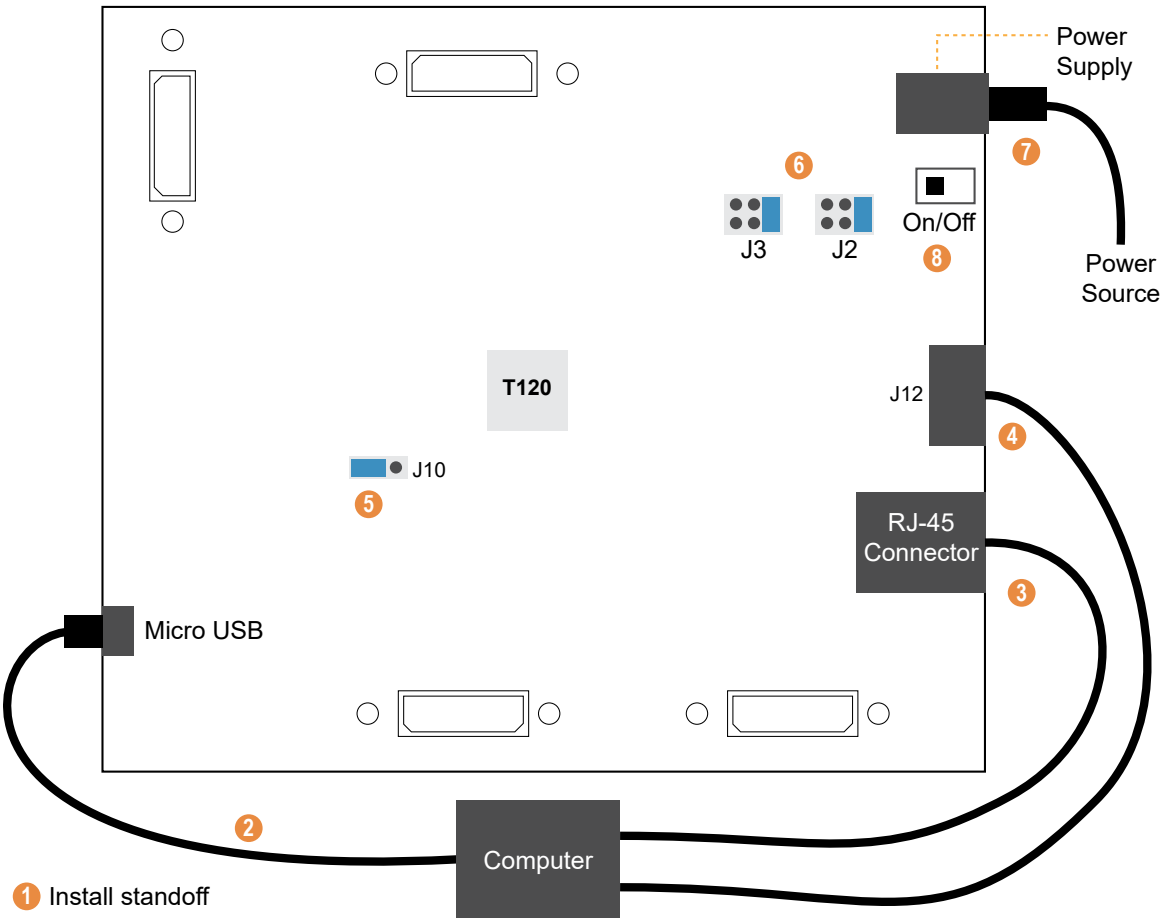
Board	Header	Pins to Connect
Development Doard	J9	Not connected
	J10, J11, J12, J13 PT1, and PT17	1 - 2 and 3 - 4
	PT2, PT3, PT4, PT5, PT6, PT7,PT8, PT9, PT10, PT11, PT12, PT13, PT14, PT15, and PT16	1 and 2
FMC-to-QSE Adapter Card	J5	5 - 6 and 7 - 8

5. Connect the FMC-to-QSE Adapter Card to the FMC connector (J5) on the development board
6. Install the nuts on the development board.
7. Connect a USB cable to the development board and your computer.
8. Connect an Ethernet cable to the RJ45 receptacle on the Ethernet Connector Daughter Card and to your computer.
9. Then, connect the 12 V power cable to the board connector and a power source.
10. Turn on the power supply and the board's power switch.
11. After turning on the board, press pushbutton SW1 (`CRESET_N`).

Trion T120 BGA324 Development Board

Important: Make sure that the power supply and the board power switch are turned off before connecting any cables.

Figure 10: Trion T120 BGA324 Development Board Hardware Setup

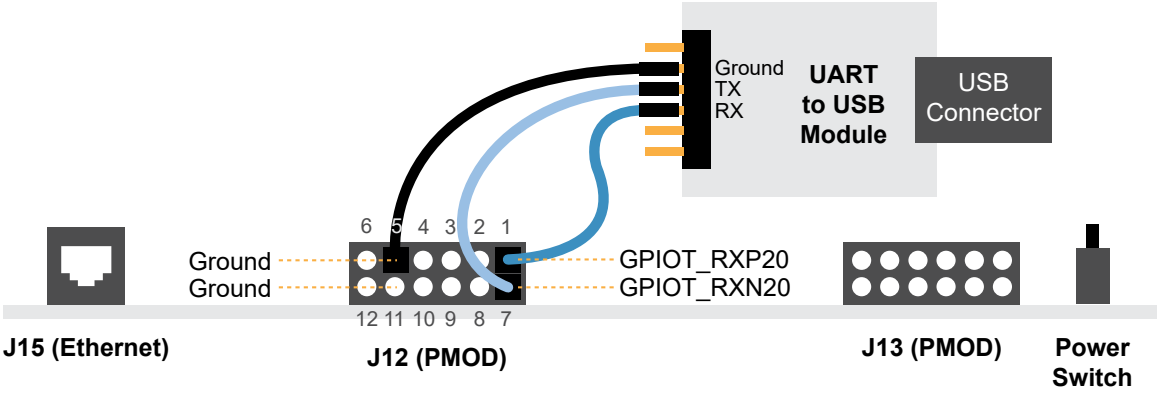


1. If you have not already done so, attach standoffs to the board.
2. Connect a USB cable to the board and to your computer.
3. Connect an Ethernet cable to the RJ-45 receptacle on the board and to your computer.
4. Connect a USB-to-UART module to the J12 header. See Set Up a USB-to-UART Module on page 12.
5. On header J10, connect pins 2 - 3 with a jumper (default) to use the on-board oscillator.
6. Leave the jumper on J2 at the defaults (connecting pins 1 - 2). On J3, connect pins 1 - 2.
7. Connect the 12 V power supply to the board connector and to a power source.
8. Turn on the board.

Set Up a USB-to-UART Module

The Trion T120 BGA324 Development Board does not have a USB-to-UART converter, therefore, you need to use a separate USB-to-UART converter module. A number of modules are available from various vendors; any USB-to-UART module should work.

Figure 11: Connect the UART Module to PMOD Connector J12

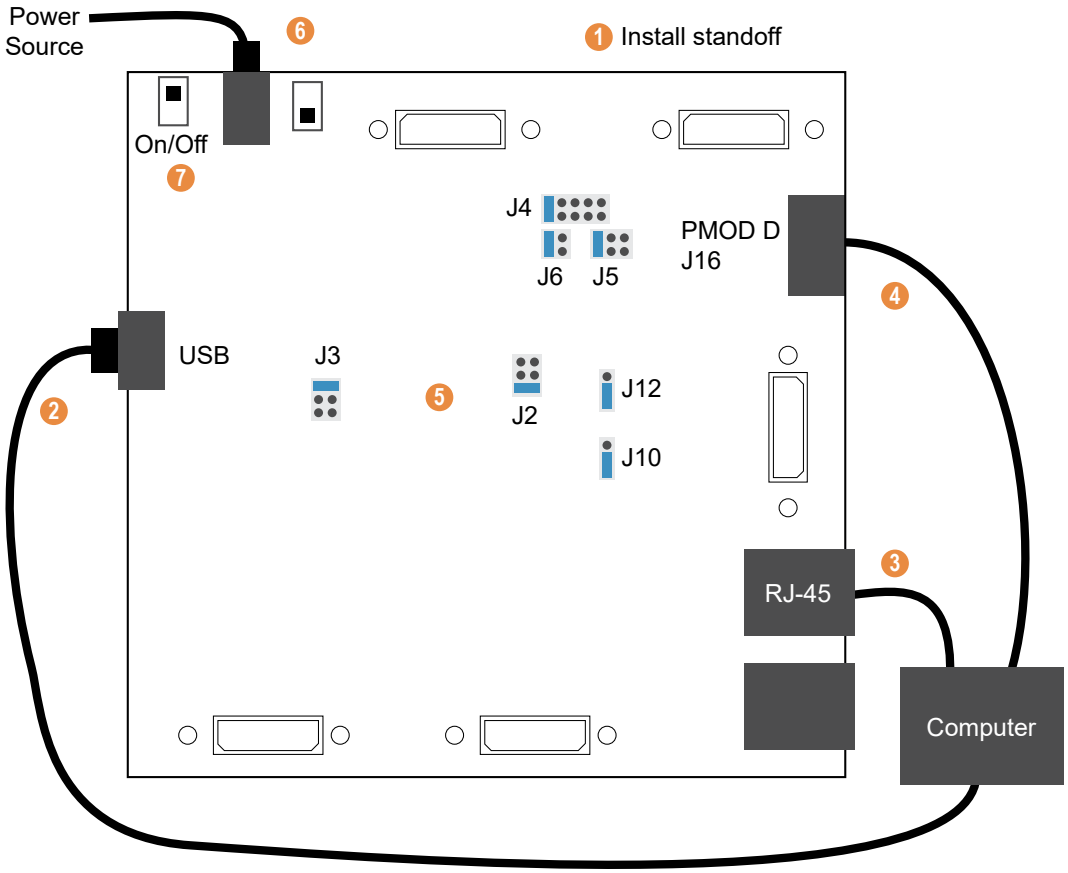


1. Connect the UART module to the PMOD port J12.
   a. *RX*—GPIOT_RXP20, which is pin 1 on PMOD J12
   b. *TX*—GPIOT_RXN20, which is pin 7 on PMOD J12
   c. *Ground*—Use ground pin 5 or 11 on PMOD J12.
2. Plug the UART module into a USB port on your computer. The driver should install automatically if needed.

Trion T120 BGA576 Development Board

Important: Make sure that the power supply and the board power switch are turned off before connecting any cables.

Figure 12: Trion T120 BGA576 Development Board Hardware Setup



1. If you have not already done so, attach standoffs to the board.
2. Connect a USB cable to the board and to your computer.
3. Connect an Ethernet cable to the RJ-45 receptacle on the board and to your computer.
4. Connect a USB-to-UART module to the J16 header. See Set Up a USB-to-UART Module on page 14.
5. Connect the following jumpers:

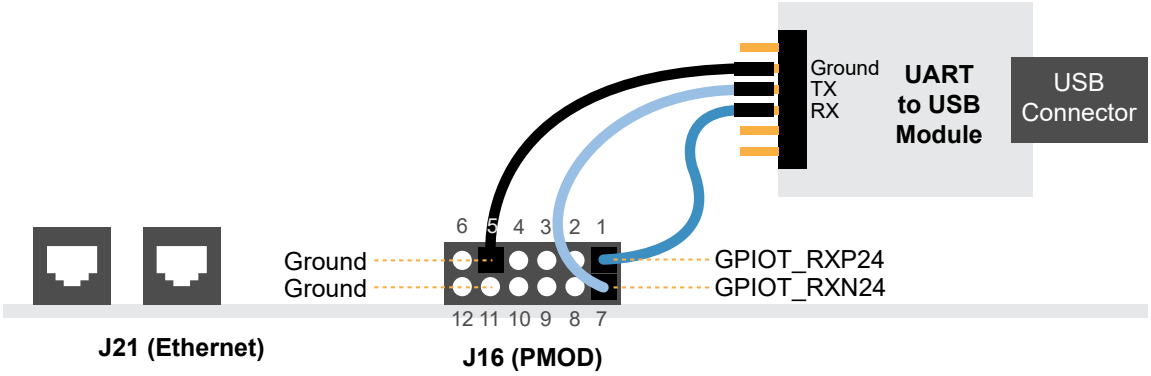
Header	Pins to Connect
J10, J12	2 - 3
J2, J3	1 - 2
J5	5 - 6
J6	3 - 4
J4	9 - 10

6. Connect the 12 V power supply to the board connector and to a power source.
7. Turn on the board.

Set Up a USB-to-UART Module

The Trion T120 BGA576 Development Board does not have a USB-to-UART converter, therefore, you need to use a separate USB-to-UART converter module. A number of modules are available from various vendors; any USB-to-UART module should work.

Figure 13: Connect the UART Module to PMOD Connector J12



1. Connect the UART module to the PMOD D port J16.
   a. *RX*—GPIOT_RXP24, which is pin 1 on PMOD D J16
   b. *TX*—GPIOT_RXN24, which is pin 7 on PMOD D J16
   c. *Ground*—Use ground pin 5 or 11 on PMOD D J16
2. Plug the UART module into a USB port on your computer. The driver should install automatically if needed.

Finding the COM Port

Finding the COM Port in Windows

1. Open Windows Device Manager by typing **Device Manager** in the Windows search box.
2. Expand **Ports(COM & LPT)** to find out UART module assigned port number. The port number is listed as USB Serial Port (COM*n*) where *n*is the assigned port number. Note the COM port number.

Finding the COM Port in Linux

1. Open the terminal and enter the following command:

```
dmesg | grep ttyUSB
```

2. The terminal displays a series of messages about the attached devices.

```
usb 3-3: FTDI USB Serial Device converter now attached to ttyUSB0
usb 3-2: FTDI USB Serial Device converter now attached to ttyUSB1
usb 3-2: FTDI USB Serial Device converter now attached to ttyUSB2
```

Set Your Computer's IP Address

Make these settings to your computer's Ethernet network adapter:

1. Set the IP address for your computer to 192.168.31.222.
2. Set the default gateway to 192.168.31.1.
3. Set the Subnet mask to 255.255.255.0.
4. Set the DNS server to 8.8.8.8.

By default, the lwIP has an FPGA IP address of 192.168.31.55, a gateway of 192.168.31.1, and expects your computer address to be 192.168.31.222. You can change these defaults (as well as the default MAC address) in the **mac.h** file in the **src** directory in the **/riscv-tsemac-lwip/src** directory. If you change these defaults, you need to re-build the project in RISC-V IDE and download the new **.elf** file to the board.

Download the Files

The 10/100/1000 Mbps TSE MAC with Sapphire SoC on lwIP example design includes a hardware project and software example code.

1. Download the example design file, **riscv-tsemac-lwip-v<version>.zip** from the Support Center.
2. Unzip the file into a folder at the root level (Windows) or your project directory (Linux).
3. The following folders are included in the **riscv-tsemac-lwip-v<version>.zip\riscv-tsemac-lwip\bsp** directory:

Folder Name	Example Design
..\ti60f225_dk	Titanium Ti60 F225 Development Board
..\ti180m484_dk	Titanium Ti180 M484 Development Board
..\ti180j484_dk	Titanium Ti180 J484 Development Board
..\t120f324_dk	Trion T120 BGA324 Development Board
..\t120f576_dk	Trion T120 BGA576 Development Board

Note: For Windows, Efinix recommends that you unzip the files into a folder at the root of the filesystem. Otherwise, you may receive build failures because the filenames are too long for the file system to handle.

Example Design Files

The example design consists of the FPGA bitstream and the RISC-V SoC application binary. You need to program these files to use the example design. You can use the example design programming the SPI flash with:

- The FPGA bitstream and the application binary together with a combined bitstream
- The FPGA bitstream and the application binary separately with two bitstreams

Generally, you use the combined bitstream to use the default example design. Use the separate bitstream if you want to program either the FPGA bitstream or the application binary portion only.

Figure 14: Example Design SPI Flash Content

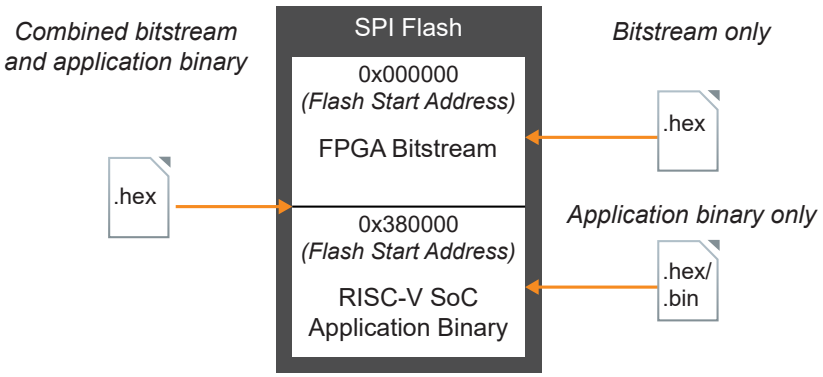


Table 2: Design Example Files and Directories

File or Directory	Description
rtl\	Folder containing common RTL files for all bsp projects.
bsp\<*board name*>\<*board name*>-lwip.xml	Example design project file.
bsp\<*board name*>\embedded_sw\soc0\	RISC-V SoC workspace.
bsp\<*board name*>\embedded_sw \soc0\software\standalone\lwip_iperf\	RISC-V SoC example design project files.
bsp\<*board name*>\embedded_sw \soc0\software\standalone\Bootloader	RISC-V SoC bootloader project files.
bsp\<*board name*>\Bitstream\FlashLoader \jtag_spi_flash_loader_dual.bit	Pre-compiled JTAG Bridge image file (TI180J/M484 Only)
bsp\<*board name*>\Bitstream\RestoreBitstream \Combine_<*board name*>_lwip.hex	Combined file consists of the FPGA bitstream and RISC-V SoC application binary. Program into the SPI flash using SPI active configuration mode.
bsp\<*board name*>\Bitstream\RestroeBitstream \<*board name*>-lwip.hex	FPGA bitstream only. Program this file into the SPI flash using SPI active configuration mode.
bsp\<*board name*>\Bitstream\RestroeBitstream \<*board name*>-lwip.bin	FPGA bitstream only. Use this file to configure the FPGA using JTAG mode configuration.
bsp\<*board name*>\Bitstream\SocFW\<*board name*>-lwip.bin	Pre-compiled example design application binary file to program into SPI flash using OpenOCD Debugger.
bsp\<*board name*>\Bitstream\SocFW\<*board name*>-lwip.elf	Pre-compiled example design application binary file to run using OpenOCD Debugger.

File or Directory	Description
bsp\<*board name*>\Bitstream\SocFW\<*board name*>-lwip.hex	RISC-V SoC application binary only. Program into the SPI flash using SPI active configuration mode.

Program the Board

The development boards ship pre-loaded with an example design. To use the 10/100/1000 Mbps TSE MAC with Sapphire SoC on lwIP example design, you must program the design into the board. To program the board with the combined bitstream file (**/Bitstream/RestoreBitstream/Combine_<*board_name*>.hex)**:

1. Open the project file (**.xml**) according to the board you are using in the Efinity® software and review it.
2. In the Efinity® Programmer, select the following programming mode depending on the development board you use:

Development Board	Mode
Ti180 M484 and Ti1810 J484	SPI Active using JTAG Bridge Programming Mode
All others	SPI Active Programming Mode

3. Under **Image**, click the **Select Image File** button and select **Combine_<*board_name*>_lwip.hex**.
4. **Ti180 M484 and Ti1810 J484 only:** Under **Auto configure JTAG Bridge Image**, click the **Select Image File** button and select **jtag_spi_flash_loader_ti180j484.bit** file.
5. Ensure that the **Starting Flash Address** is set to `0x000000`.
6. Click the **Start Program** button.
7. Press the CRESET button on the development board to reset the FPGA. This reset ensures that the DDR memory initialization happens before the user application runs.

The board LEDs outputs the following to indicate successful configuration.

Development Board	LED	Description
Titanium Ti60 F225 Development Board	D16 blinking green	Ethernet RX clock received
	D17 turned on blue	HyperRAM calibration passed
	D18 blinking green	PLL locks with RGMII RXC clock
	D22 turned on green	Power good
	D15 turned on green	FPGA configuration done
	D20 turned on red	FPGA configuration error
Titanium Ti180 M484 and J484 Development Boards	LED11, LED12, LED13 turned on	Power good
	LED1 turned on	FPGA configuration done
	LED7 turned on	DDR configuration done
	LED6 turned on	DDR memory test done
	LED2 blinking	Data received
	LED3 blinking	Data transmitted
	LED4 blinking	Ethernet RX clock received
	LED5 blinking	Ethernet TX clock received

Development Board	LED	Description
Trion T120 BGA324 Development Board	D7 (User_LED0) blinking	Ethernet RX clock received
	D1, D2, D3, D4 turned on	Power Good
	D6 turned on	FPGA configuration done
Trion T120 BGA576 Development Board	D7 (User_LED0) blinking	Ethernet RX clock received
	D1, D2, D3, D4 turned on	Power good
	D6 turned on	FPGA configuration done

Using the Software Examples

The example comes with all of the standard Sapphire software in the lwip_iperf directory. You can build the software yourself or you can use the pre-compiled **.elf** files.

Using this Example with the RISC-V IDE

Before working with the software included with this example design, you should already be familiar with using the Sapphire SoC and Efinity RISC-V Embedded Software IDE. Refer to the following **Sapphire RISC-V SoC Hardware and Software User Guide** sections for detailed instructions on how to perform the required tasks:

- Install the Efinity RISC-V Embedded Software IDE
- Launching the Efinity RISC-V Embedded Software IDE
- Import Sample Projects
- Build
- Launch the Debug Script
- Open a Terminal

iPerf2 Server with lwIP and TCP Stack

This application shows how lwIP and a TCP stack TCP function with the TSEMAC IP core. The iPerf2 tool lets you actively measure the maximum achievable bandwidth on IP networks. You can tune parameters such as timing, buffers, and protocols. With this application you use the iPerf2 client to test the speed with the FPGA's iPerf2 server.

1. In Efinity RISC-V IDE, run the application.
2. Click the **Resume** button or press F8 to resume code operation.
3. Go to your serial terminal. You should see the following messages display:

```
PHY Init...
Waiting Link Up...
Linked Up...
iperf server up
=========================================
======lwIP Raw Mode Iperf TCP Server======
=========================================
======IP:              192.168.31.55
======Netmask:         255.255.255.0
======Gateway:         192.168.31.1
======Link Speed:      1000 Mbps
=========================================
```

Note: If the Ethernet does not link up, check that you have set up the IP address and other network settings for the correct network adapter.

4. Open a second terminal or command prompt.
5. Change to the directory has the iPerf2 executable.
6. Type the command:

```
.\iperf.exe -c 192.168.31.55 -i 1
```

You should see output similar to the following in the terminal:

```
------------------------------------------------------------
Client connecting to 192.168.31.55, TCP port 5001
TCP window size: 136 KByte (default)
------------------------------------------------------------
[  3] local 192.168.31.123 port 47110 connected with 192.168.31.55 port 5001
[ ID] Interval        Transfer        Bandwidth
[  3] 0.0- 1.0 sec    8.62 MBytes     72.4 Mbits/sec
[  3] 1.0- 2.0 sec    8.38 MBytes     70.3 Mbits/sec
[  3] 2.0- 3.0 sec    8.50 MBytes     71.3 Mbits/sec
[  3] 3.0- 4.0 sec    8.38 MBytes     70.3 Mbits/sec
[  3] 4.0- 5.0 sec    8.50 MBytes     71.3 Mbits/sec
...
```

Revision History

Table 3: Revision History

Date	Version	Description
March 2024	1.3	Updated figure and changed title from Sapphire SoC Peripherals to Titanium Sapphire SoC Peripherals. (DOC-1671) Added in new figure Trion Sapphire SoC Peripherals.
September 2023	1.2	Added support for Ti180 J484, T120 BGA324, and T120 BGA576 Development Boards. (DOC-1438)
March 2023	1.1	Added information for Ti180 M484.
February 2023	1.0	Initial release.