AN4964

MPLAB® Harmony v3 LAN865x Driver Example

1.0 INTRODUCTION

The LAN865x is a high-performance 10BASE-T1S MAC-PHY Ethernet Controller with SPI that is targeted for 10 Mbit/s half-duplex networking over a single pair of conductors.

This document guides you in creating a sample TCP/IP Client node (bare-metal or FreeRTOS™ based), using the LAN865x MAC-PHY. It describes how to configure the MAC-PHY in either Physical Layer Collision Avoidance (PLCA) or Carrier-Sense Multiple Access/Collision Detection (CSMA/CD) mode.

The description in this document is based on an ATSAME54P20A running on a SAM E54 Curiosity Ultra Development Board [4]. However, it can also be applied to other infrastructures; for example, to an ATSAME70Q21B running on a SAM E70 Xplained Ultra Evaluation Kit [5].

1.1 Audience

This document is written for developers who want to create a sample TCP/IP Client node, using the LAN865x MAC-PHY. Developers should be familiar with the infrastructure of MPLAB® X IDE and its plug-ins [1].

1.2 Prerequisites

The following software components are required for getting started and to configure the example described in this document:

- MPLAB X IDE
  https://www.microchip.com/en-us/tools-resources/develop/mplab-x-ide
- MPLAB XC32
  https://www.microchip.com/en-us/tools-resources/develop/mplab-xc-compilers
- MPLAB Code Configurator (MCC)
  https://www.microchip.com/en-us/tools-resources/configure/mplab-code-configurator

1.3 References

The following sources should be referenced when using this application note.

[1] MPLAB Code Configurator
https://www.microchip.com/en-us/tools-resources/configure/mplab-code-configurator

[2] MPLAB Code Configurator Getting Started
https://microchipdeveloper.com/mcc:start

[3] MPLAB XC Compiler Documentation
https://www.microchip.com/en-us/tools-resources/develop/mplab-xc-compilers/downloads-documentation#XC32

[4] SAM E54 Curiosity Ultra Development Board
https://www.microchip.com/Developmenttools/ProductDetails/DM320210

[5] SAM E70 Xplained Ultra Evaluation Kit
https://www.microchip.com/Developmenttools/ProductDetails/DM320113

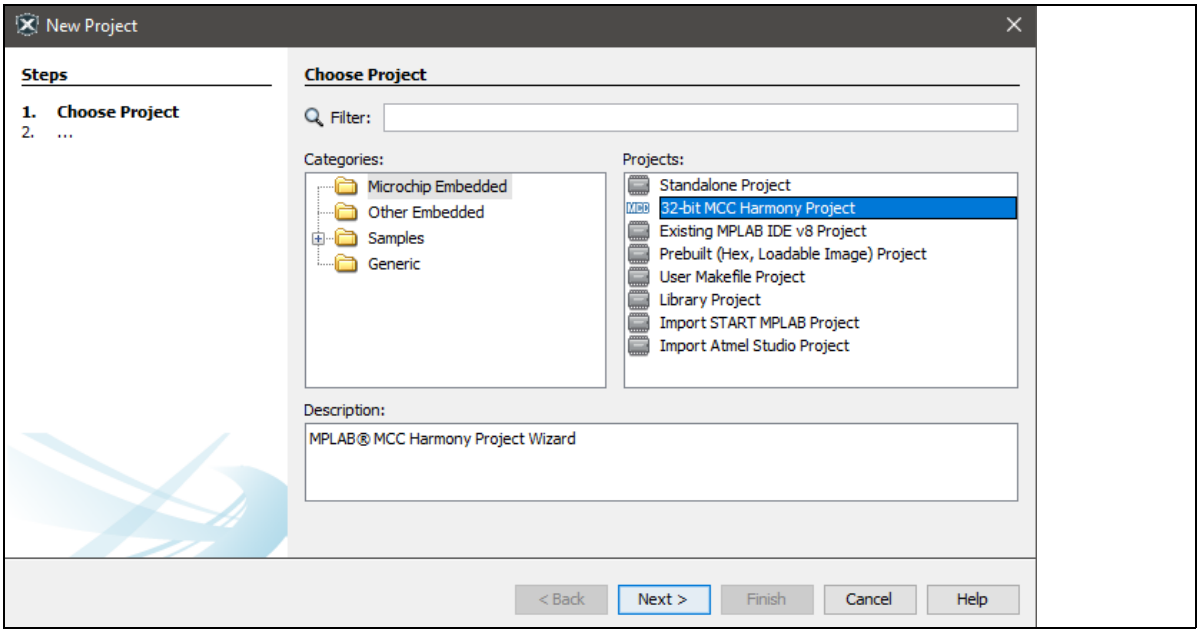
1.4 Abbreviations

Abbreviation	Definition
CSMA/CD	Carrier-Sense Multiple Access/Collision Detection
EDBG	Embedded Debugger
MCC	MPLAB Code Configurator
NVIC	Nested Vectored Interrupt Controller
PLCA	Physical Layer Collision Avoidance

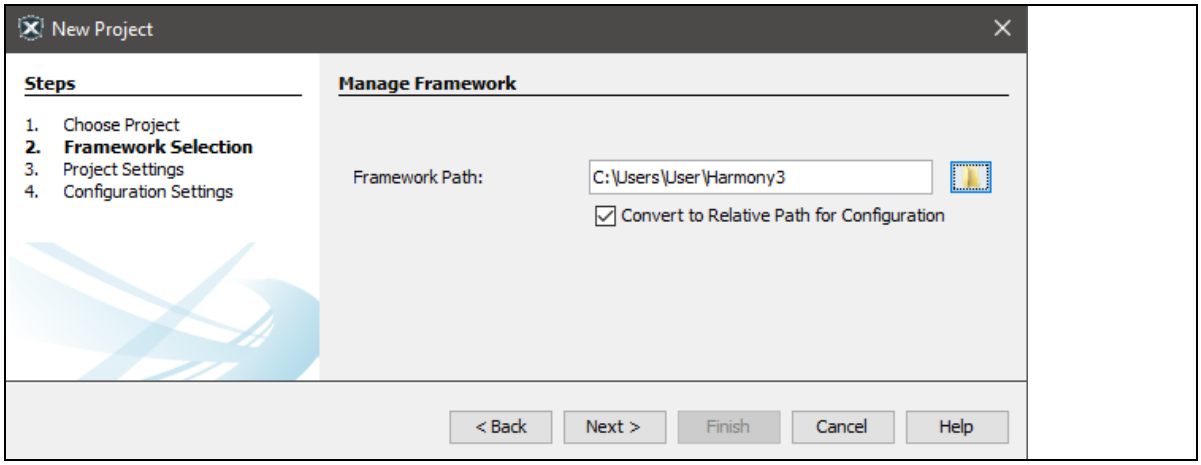
2.0 CREATE A NEW MCC HARMONY PROJECT

2.1 Create a New Project

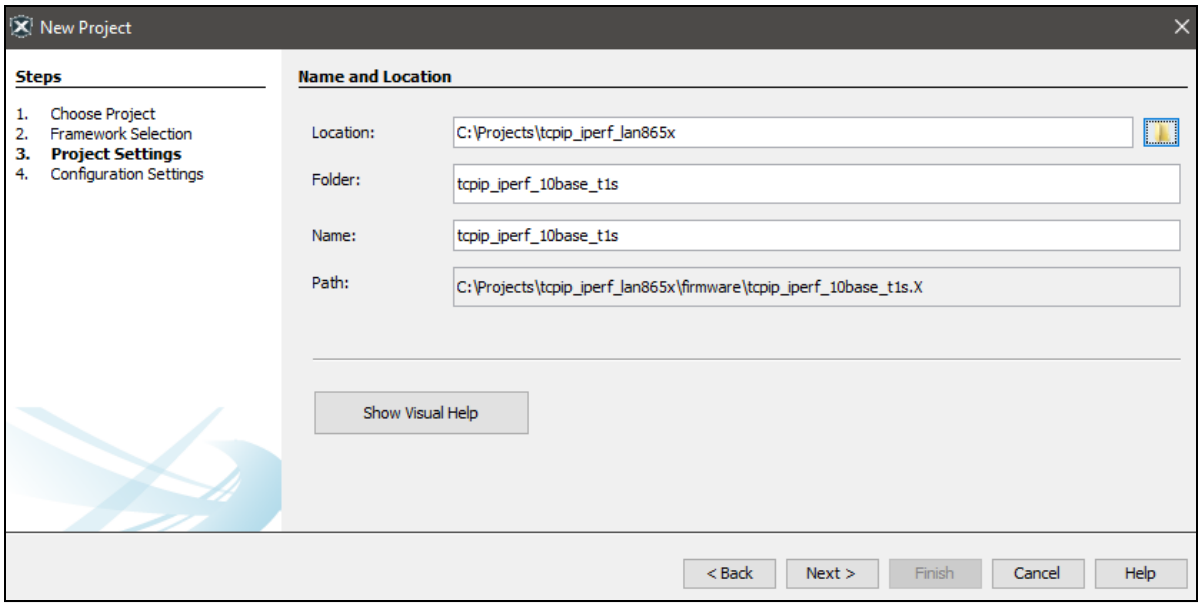
1. Open MPLAB X IDE.
2. Create a new project by either
   a) clicking the new project icon  or
   b) by selecting *File > New Project*.
3. In the “New Project” window:
   a) Select “32-bit MCC Harmony Project”.
   b) Click **Next**.



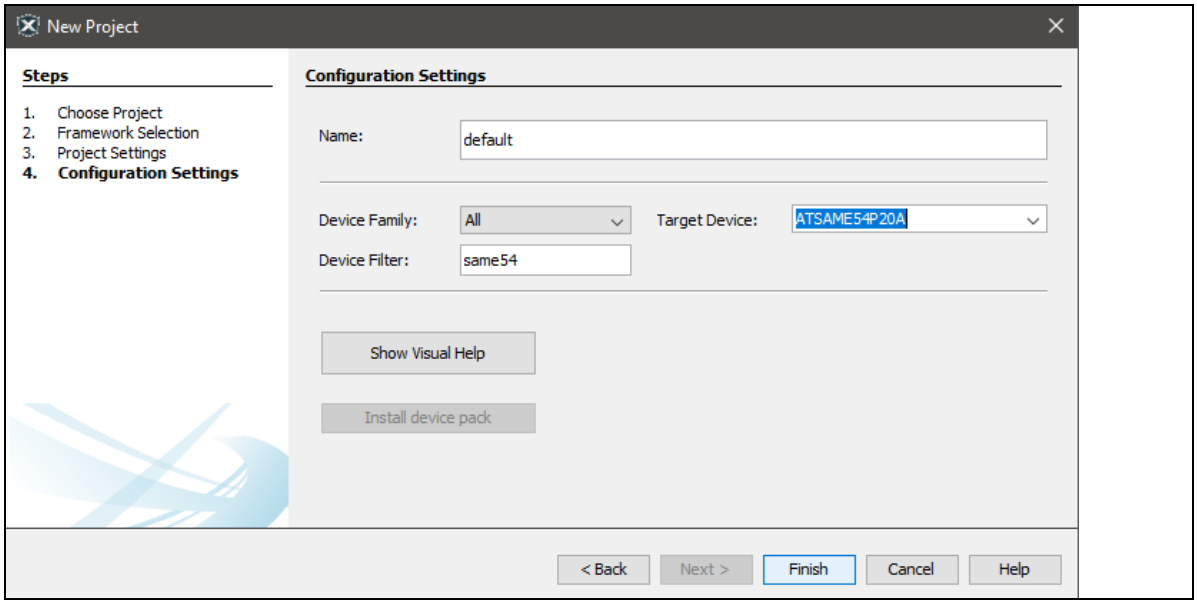
4. In the “Framework Path”:
   a) Enter the path of the folder to which the MCC Harmony packages are downloaded.
   b) Click **Next**.



5. In the “Project Setting” dialog window:
   a) Fill in or select the information needed as follows:
      - Location:
        First, create a *“tcpip_iperf_lan865x”* folder in the location of your choice (for example, create the application project inside the *C:\Projects folder*)
        Note: Use an appropriate project name.
        Then enter the path in this field.
      - Folder:
        Enter a project folder name (for example, *“tcpip_iperf_10base_t1s”*).
        (If you have a FreeRTOS project, you can enter as project folder name *“tcpip_iperf_10base_t1s_freertos”*.)
      - Name:
        The information entered in the above field auto populates the *Name*.
      - Path:
        The information entered in the above fields auto populates the *Path*.
   b) Click **Next**.

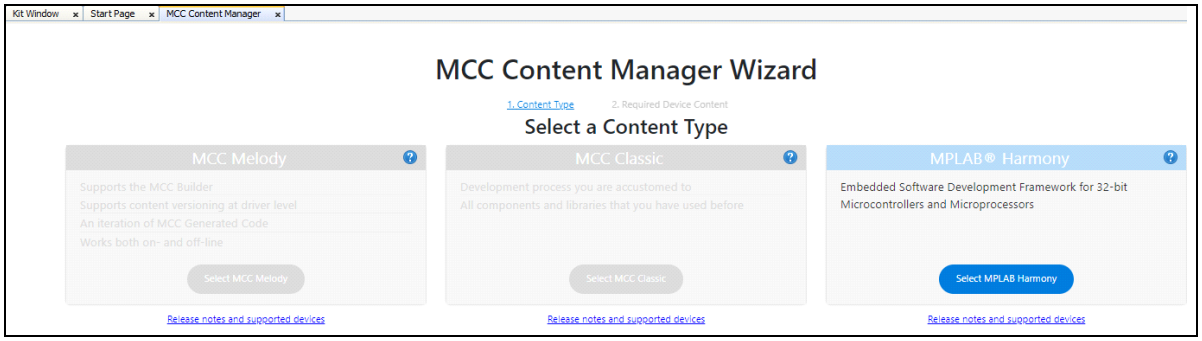


6. In the “Configuration Settings” dialog window:
   a) Fill in or select the information needed as follows:
      - Name:
        Enter the configuration name (for example, “*default*”).
      - Device Filter:
        Enter *“same54”* in device filter to find SAME54 board variants.
      - Target Device:
        Select “ATSAME54P20A” (the target device is running on a SAM E54 Curiosity Ultra Development Board [4]).
   b) Click **Finish**.



When you create an MCC project, the Content Manager page opens and you can select the MCC flavor, based on the project.

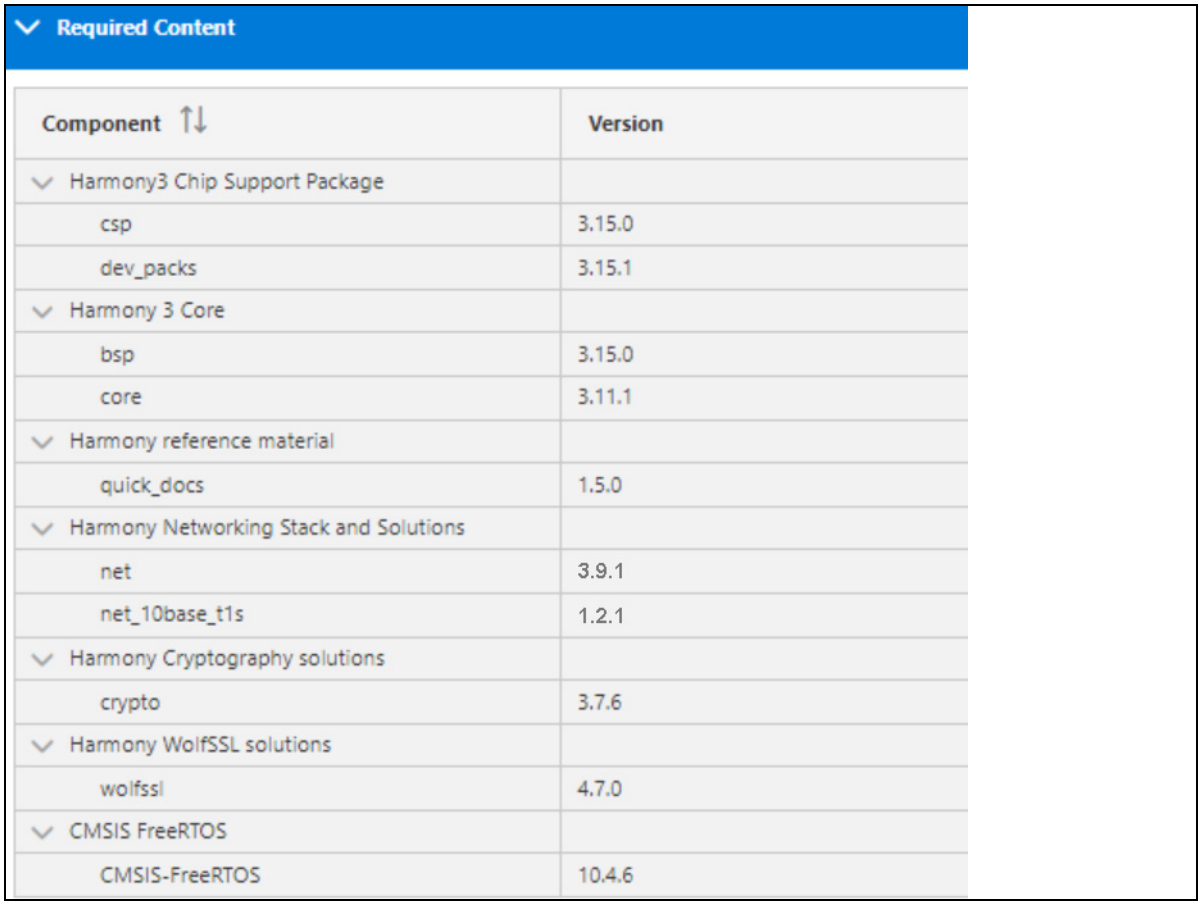
7. In the “MCC Content Manager Wizard”, click the **Select MPLAB Harmony** button.



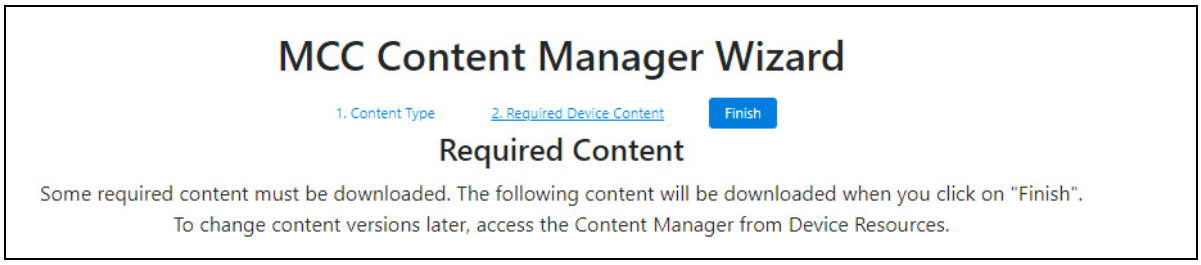
When you create an MCC project for the first time, you need to select and download all required device content libraries.
Selection of the libraries is done on the “Optional Content” list. Once selected, the libraries are reflected on the “Required Content” list.

Note: The “Required Content” list shown below gives an overview of the libraries that need to be downloaded. Version numbers may differ. Therefore, always use the latest versions.

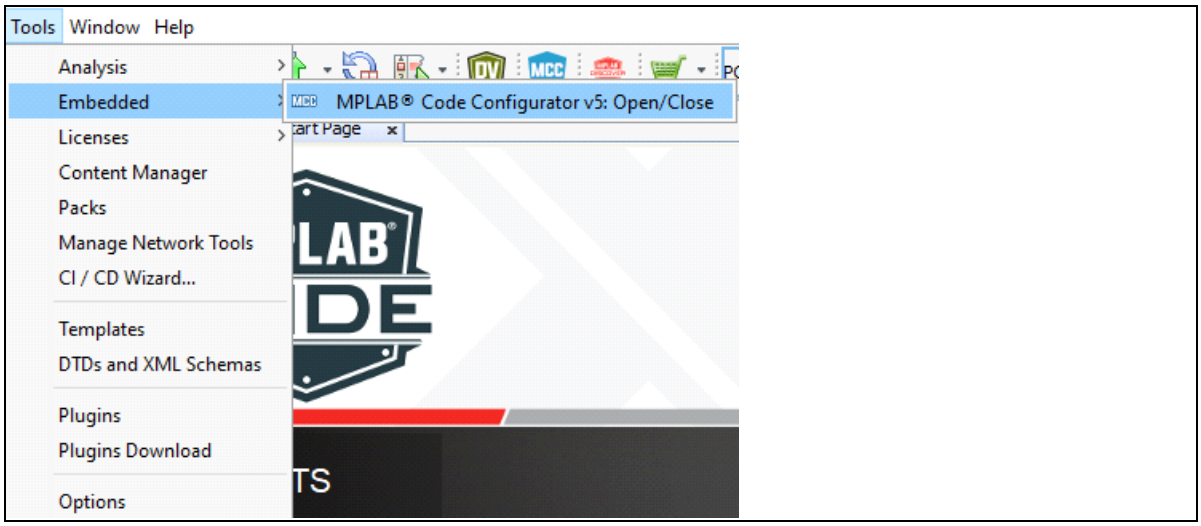
8. Select the libraries from the “Optional Content” list.



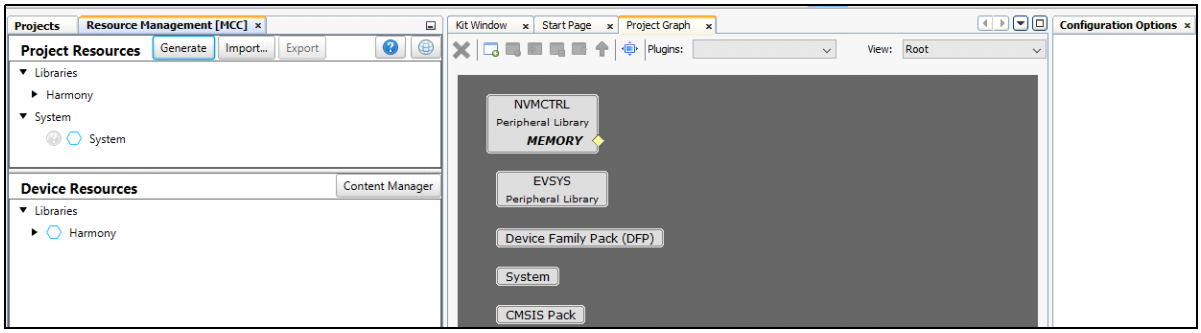
9. Click **Finish** to start the download.



If the “MCC Content Manager Wizard” window does not appear, the MCC can be launched under *Tools > Embedded*, by selecting “MPLAB Code Configurator v5”.



The default MCC looks as shown below.



The different parts of the MCC are:

- Resource Management
- Project Resources
- Device Resources
- Project Graph
- Configuration Options

Depending upon application needs, the necessary components can be added from the “Project Resources”.

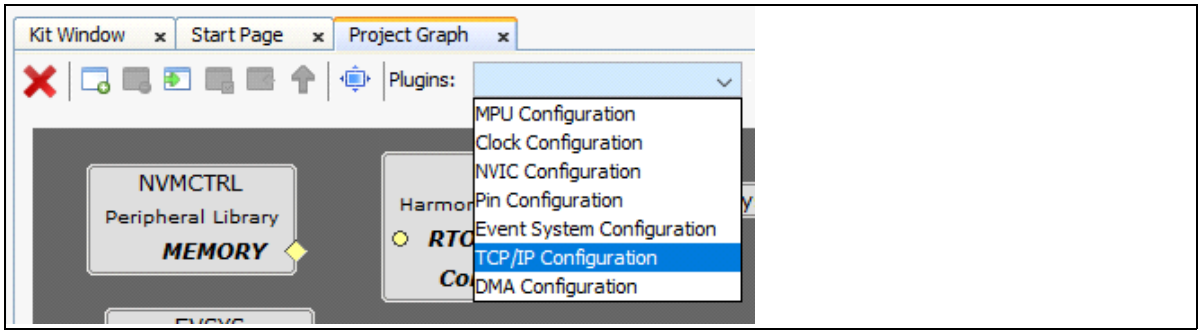
Basic components needed for creating a project are available in the “Project Graph” by default.

2.2 Add TCP/IP Components

In this section we configure the components based on application needs.

The TCP/IP components can be added from the “TCP/IP Configuration” plugin.

1. Go to *Project Graph* > *Plugins* > *TCP/IP Configuration*.
   This opens the “TCP/IP Configuration” window.



The TCP/IP components are categorized into different groups. For each group, a configurator is available.

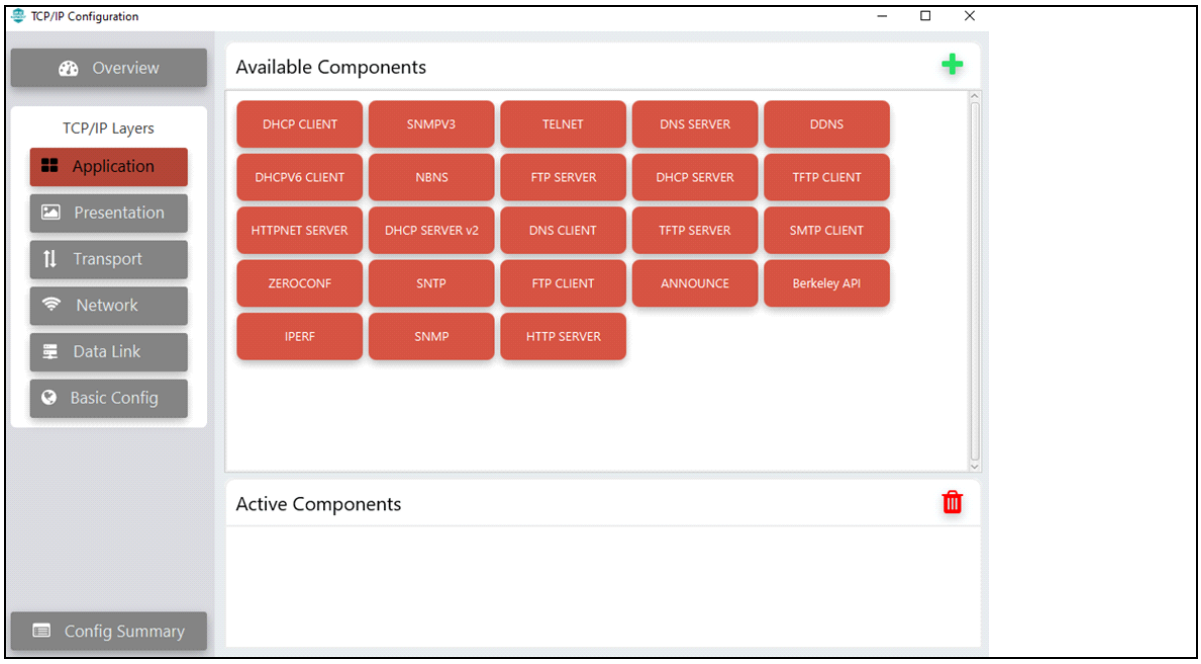
These configurators are:

- Application Layer Configurator
- Transport Layer Configurator
- Network Layer Configurator
- Data Link Layer Configurator
- Basic Configurator

2.3 Application Layer Configurator

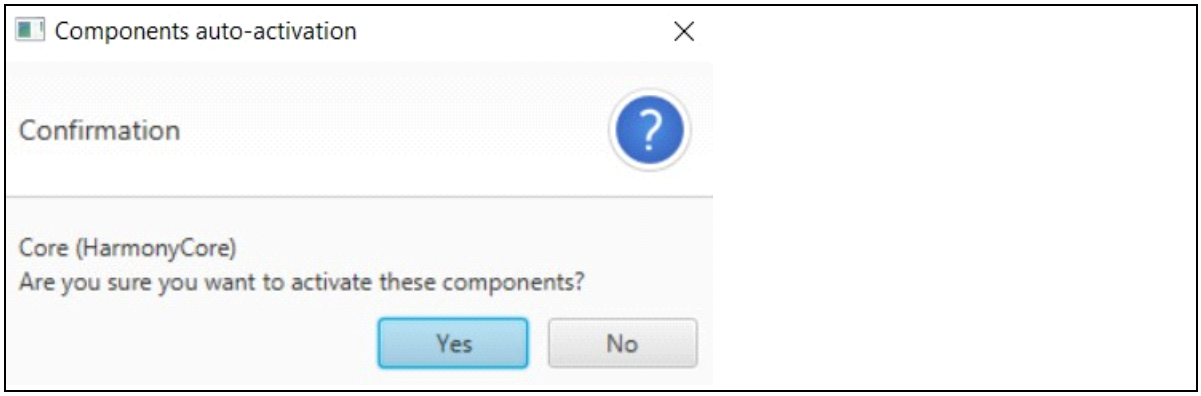
1. Click the **Application** button to select the application layer.
   This step also creates a hierarchy of groups as *Root >TCP/IP STACK >APPLICATION LAYER*.

   Different application protocols supported by the MPLAB Harmony TCP/IP stack are listed in the “Available Components” area.

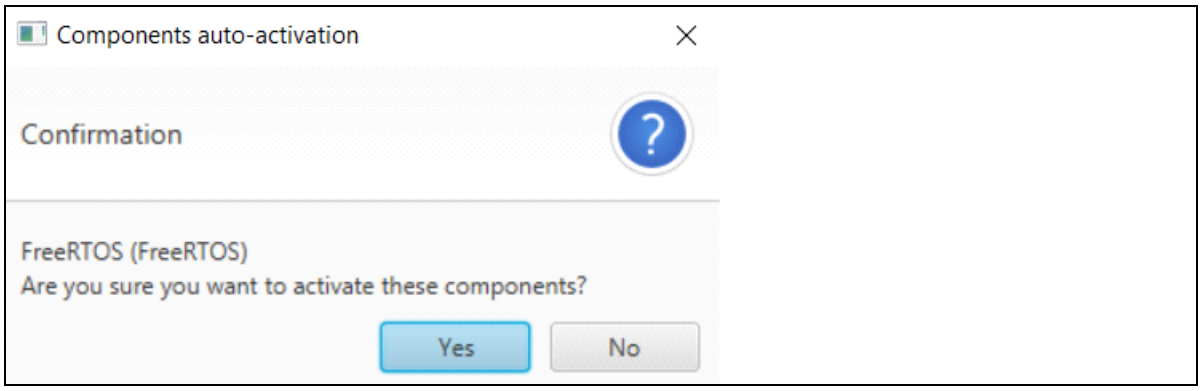


   Start selecting the required TCP/IP application layer protocols from the “Available Components” on the right-hand side.

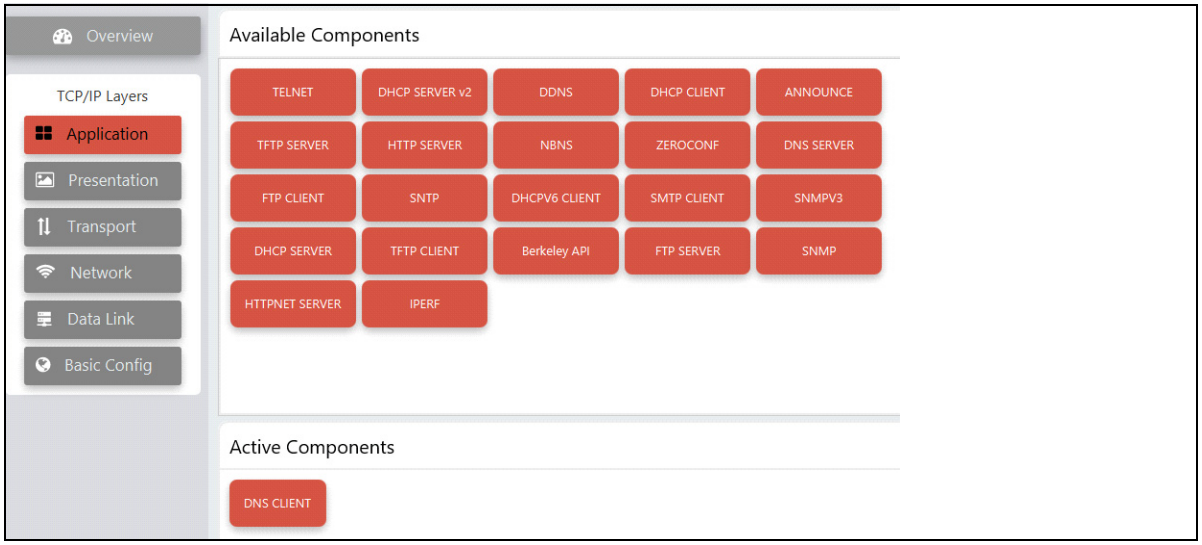
2. Drag and drop “DNS CLIENT” into the “Active Components” area.
   The following steps auto-activate all dependent components.
   a) Click **Yes** to add the “Harmony Core” component.



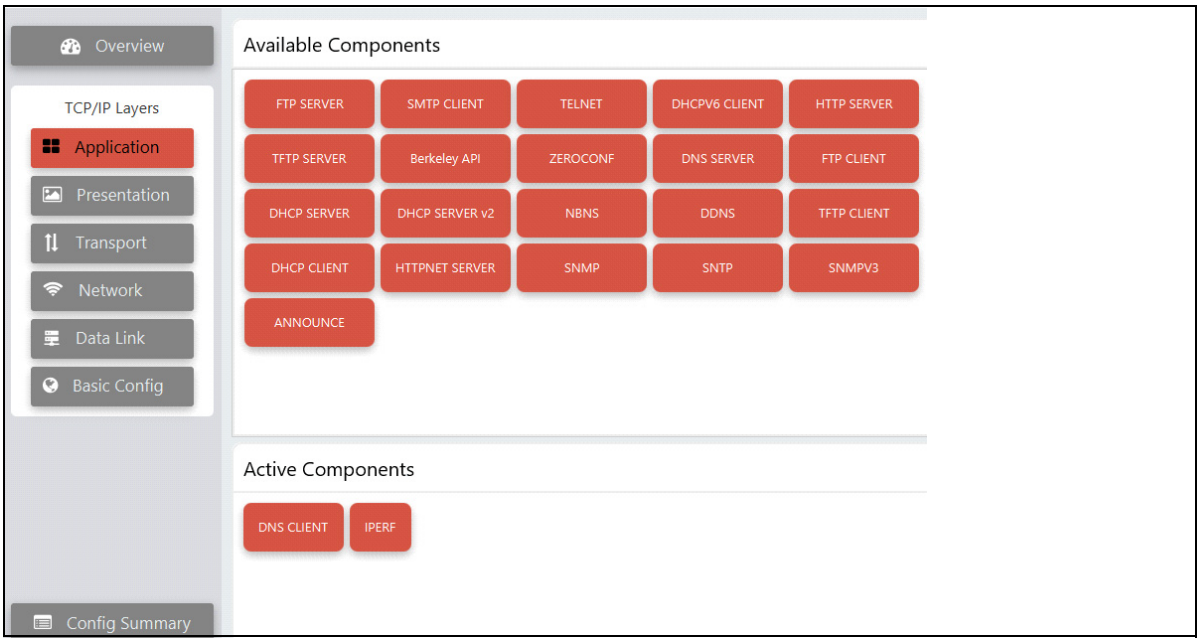
b) If you have a bare-metal project, click **No** for the “FreeRTOS” component.
c) If you have a FreeRTOS project, click **Yes** for the “FreeRTOS” component and add necessary FreeRTOS settings (see FreeRTOS Configuration).



After the auto-activation of dependent components, the “DNS CLIENT” component is added to the “Active Components” and “Project Graph”.



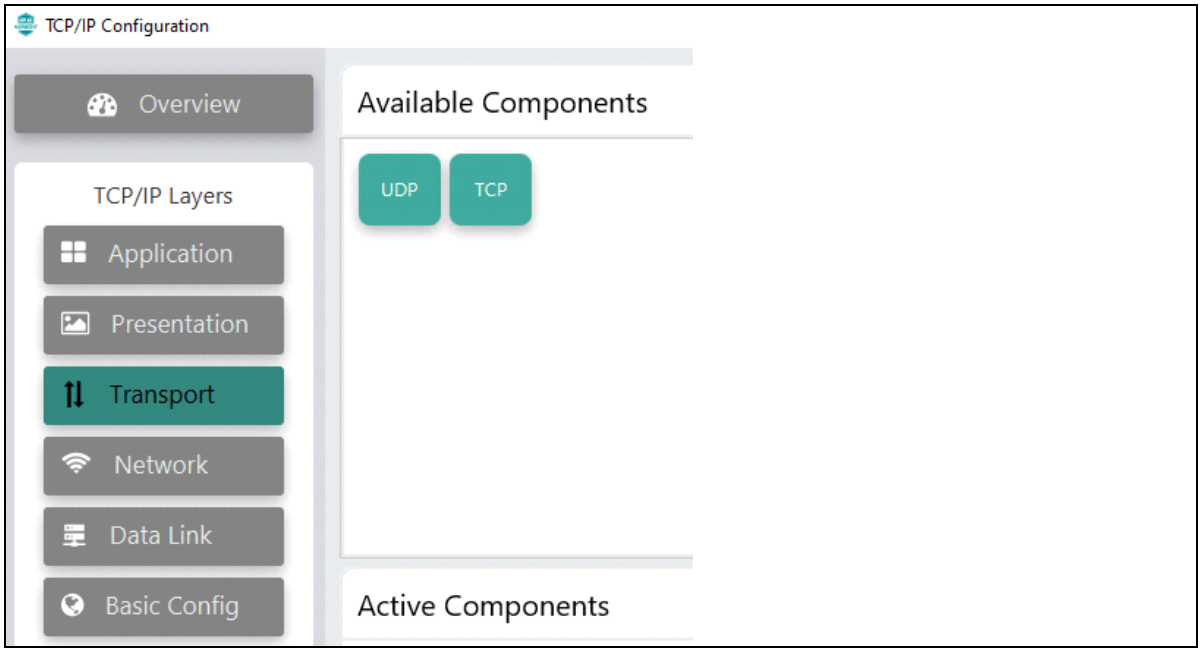
3. Drag and drop “IPERF” into the “Active Components” area.



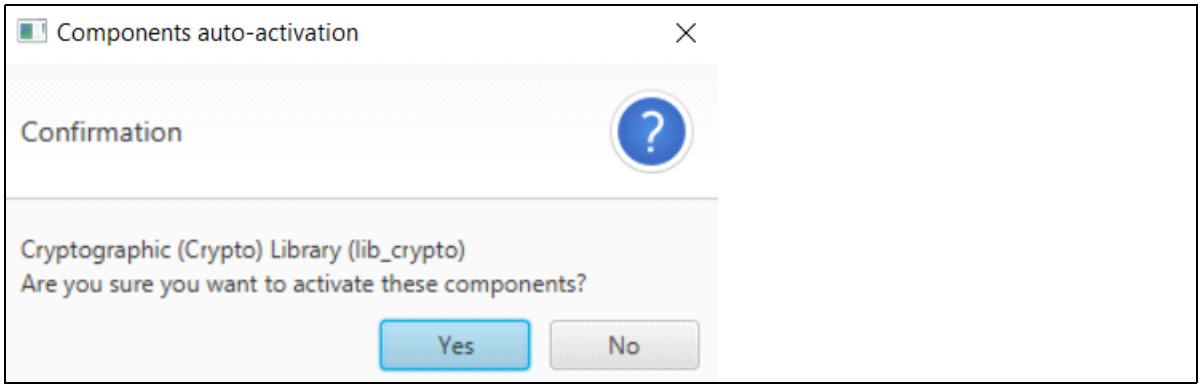
This adds the “IPERF” components to the “Active Components” and “Project Graph”.

2.4 Transport Layer Configurator

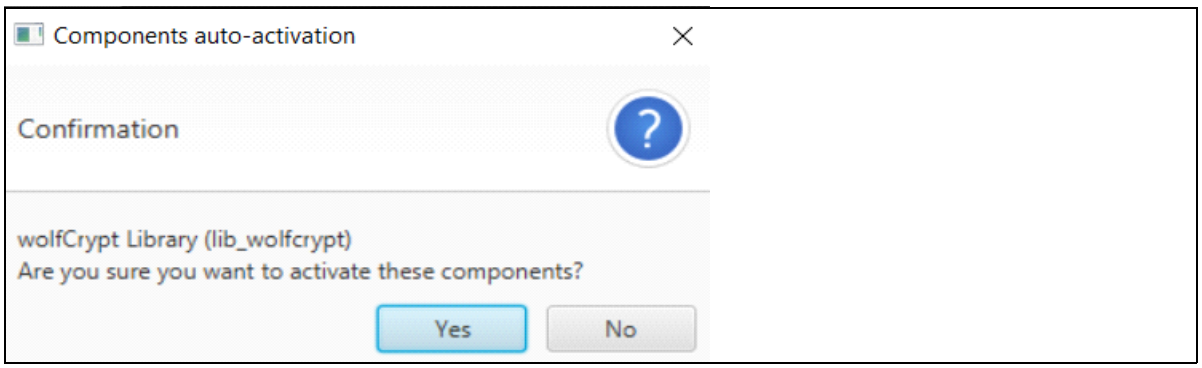
1. Click the **Transport** button to select the transport layer.



2. Drag and drop “TCP” into the “Active Components” area.
   The following steps auto-activate all dependent components.
   a) Click **Yes** to add the “lib_crypto” component.

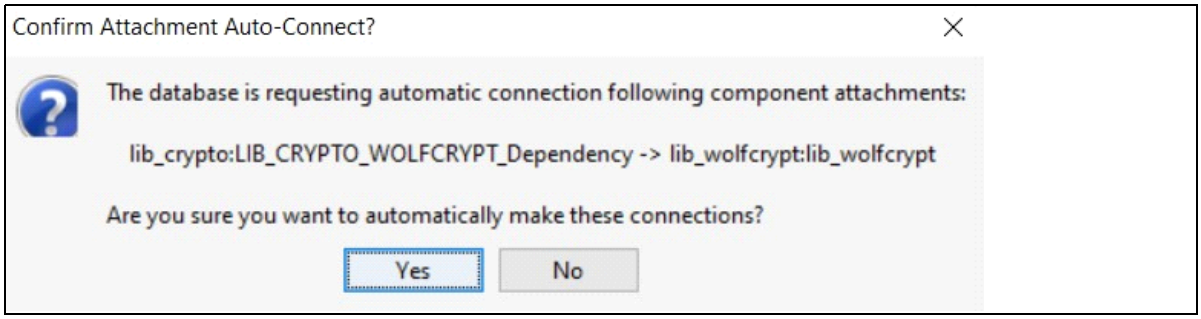


   b) Click **Yes** to add the “lib_wolfcrypt” component.



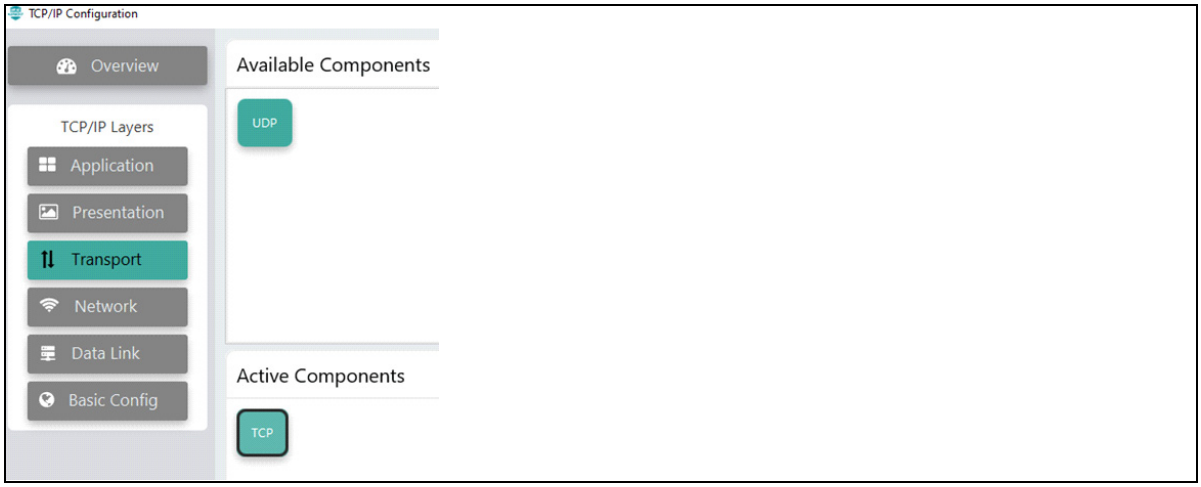
c) Click **Yes** to add the connection “lib_wolfcrypt”.

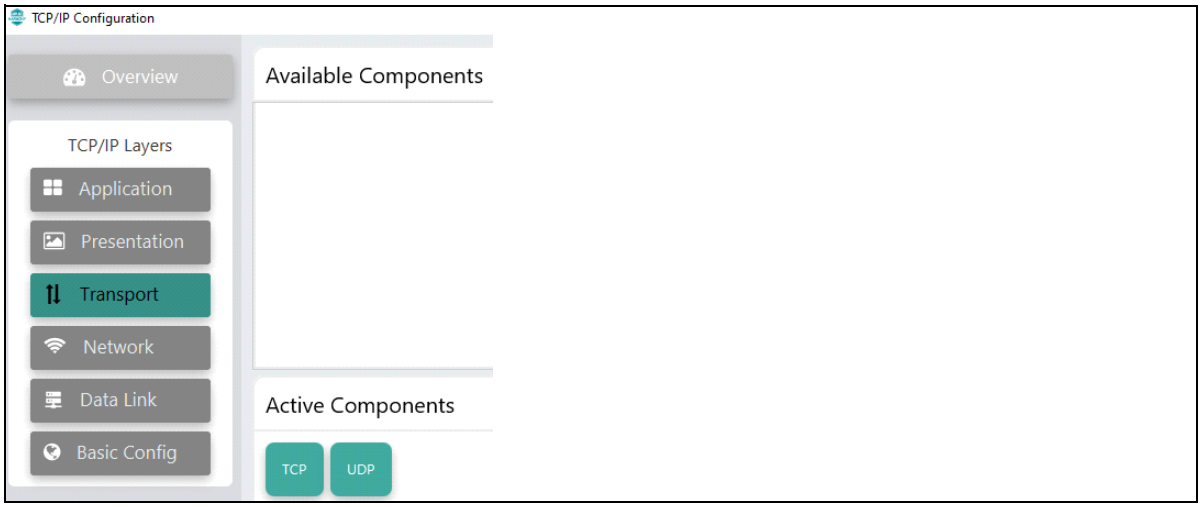
Note: The “Confirm Attachment Auto-Correct?” popup appears behind the TCP/IP Configuration window. Therefore, minimize the TCP/IP Configuration window to view this popup.



This adds the “TCP” component to the “Active Components” and “Project Graph”.

3. Go to *Project Graph* > *Plugins* > *TCP/IP Configuration*.
4. Drag and drop “UDP” into the “Active Components” area.

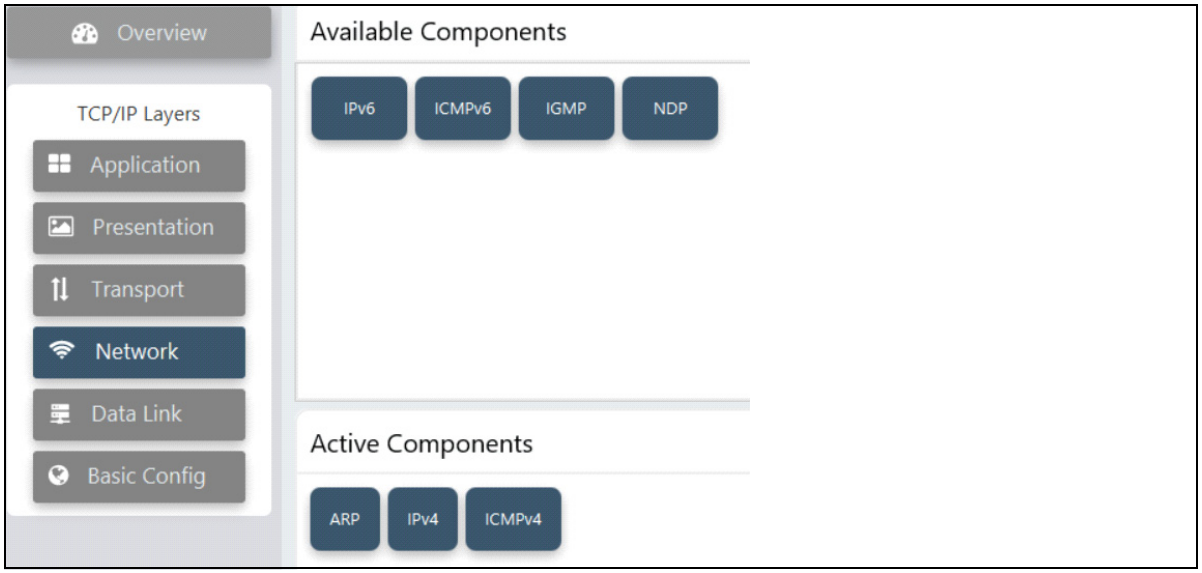




This adds the “UDP” component to the “Active Components” and “Project Graph”.

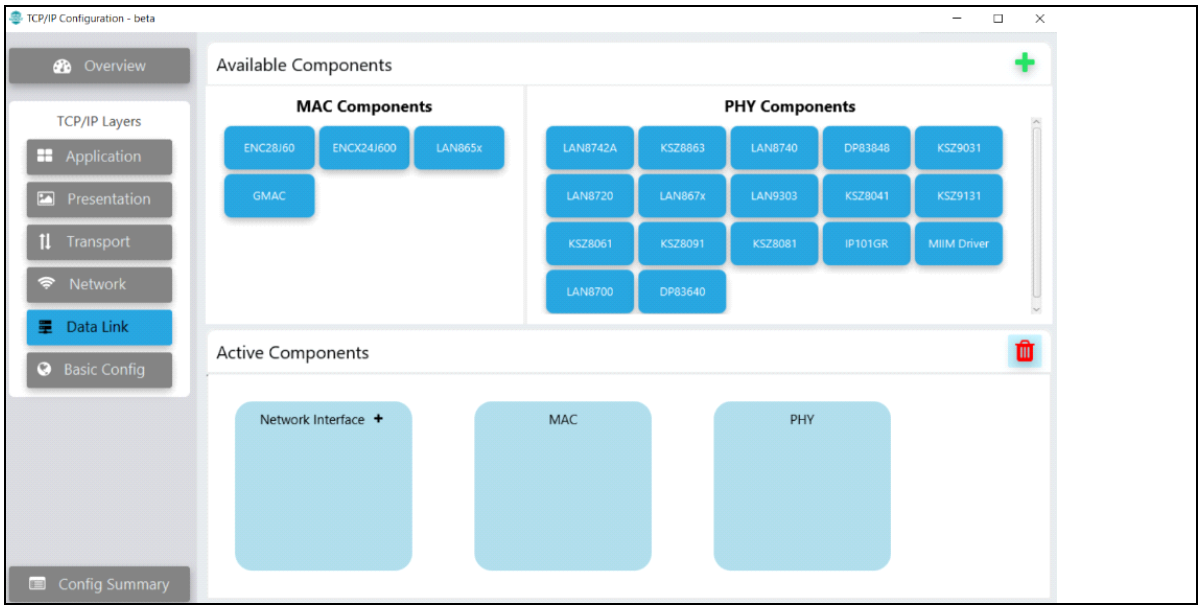
2.5 Network Layer Configurator

1. Click the **Network** button to select the network layer.
2. Drag and drop “ARP”, “IPv4” and “ICMPv4” from the “Available Components” area into the “Active Components” area.

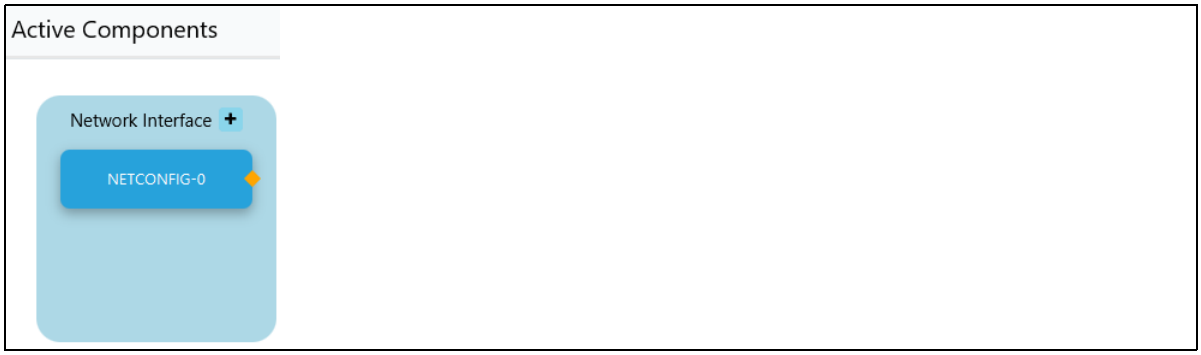


2.6 Data Link Layer Configurator

1. Click the **Data Link** button to select the data link layer.
   Different MAC and PHY components supported in the MPLAB Harmony TCP/IP stack are listed.

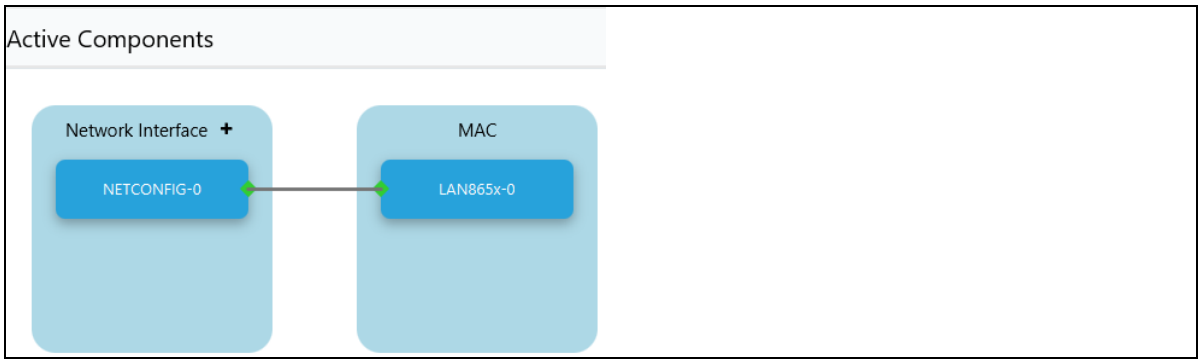


2. Add a Network Interface.
   Click the “+” sign on the “Network Interface” block located in the “Active Components” area.



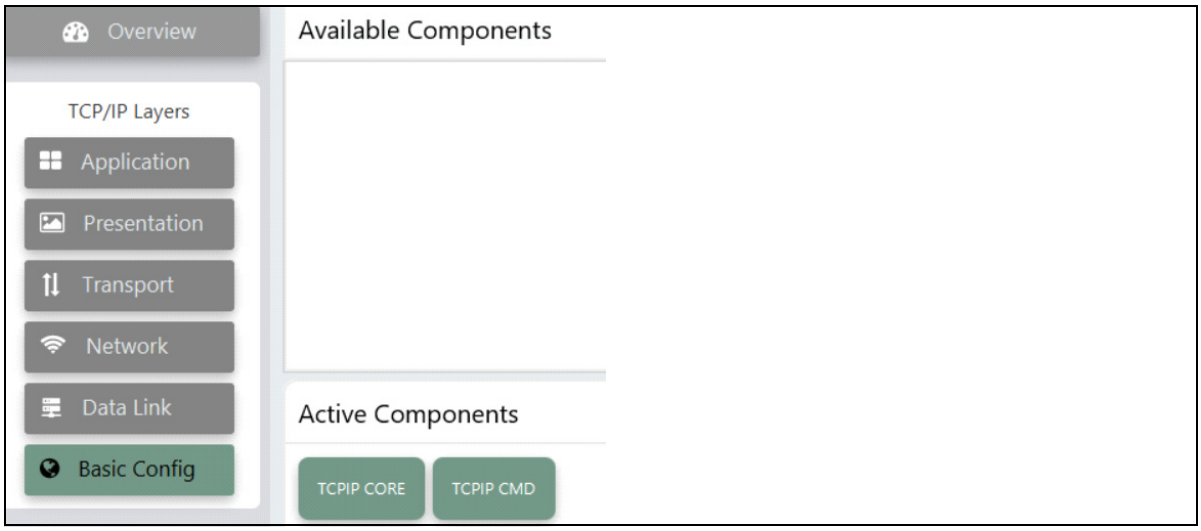
This adds a NETCONFIG Instance.

3. Add a LAN865x MAC to this demo.
   a) Drag and drop “LAN865x” from “Available Components” (“MAC Components”) into the MAC block in the “Active Components” area.
   b) Connect “NETCONFIG-0” with the “LAN865x-0” by dragging the orange diamond from NETCONFIG-0 to LAN865x-0.
   The two diamonds then become green.



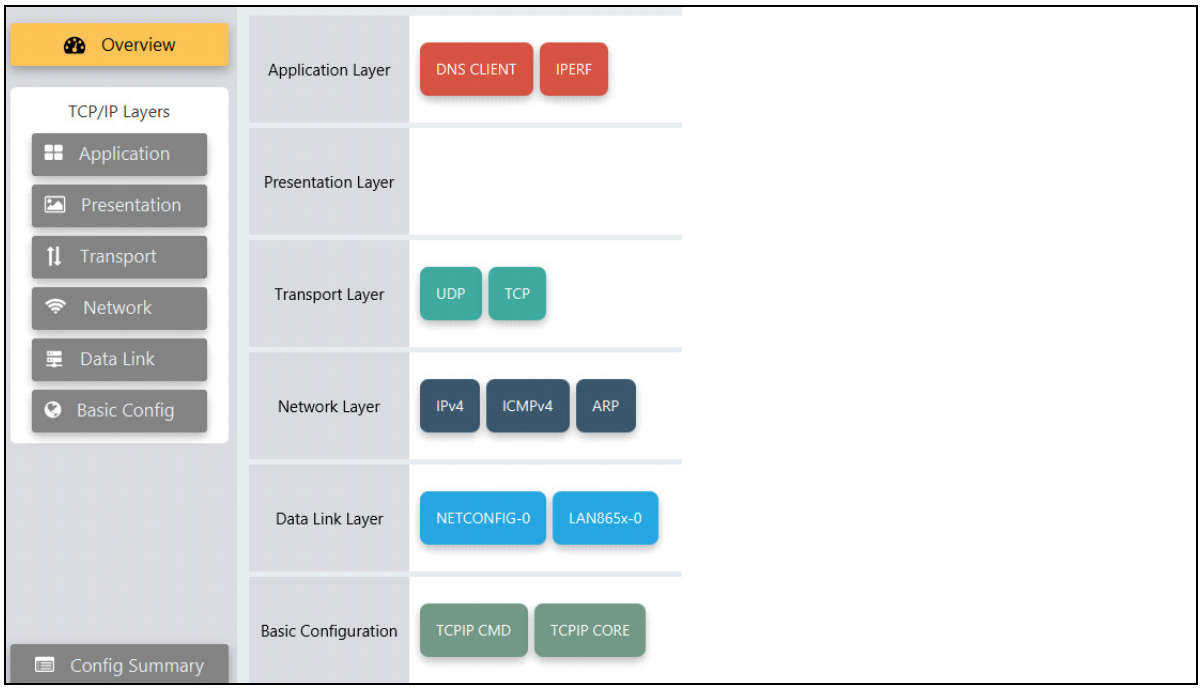
2.7 Basic Configurator

1. Click the **Basic Config** button to select the basic config layer.
2. Drag and drop “TCPIP CMD” from the “Available Components” area into the “Active Components” area.

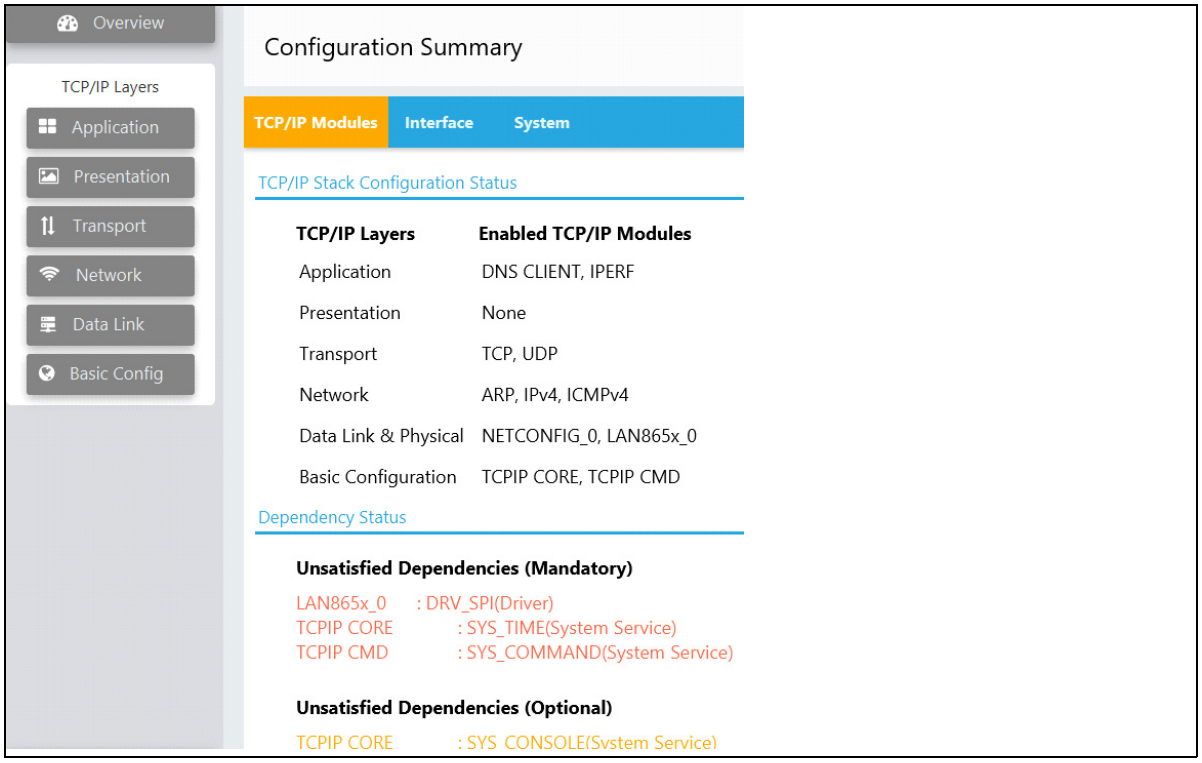


2.8 Overview and Configuration Summary

1. In the TCP/IP Configuration window:
   a) Click the **Overview** button to see all added components.

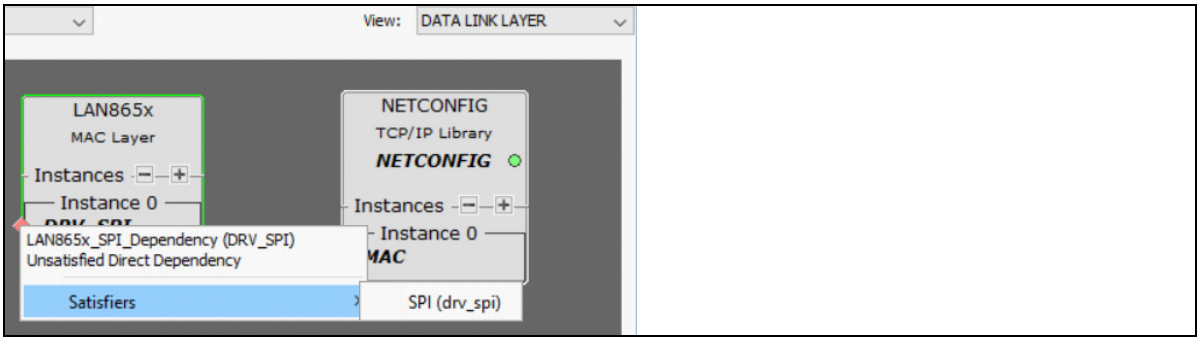


   b) Click the **Config Summary** button (at lower left) to view the dependency status.

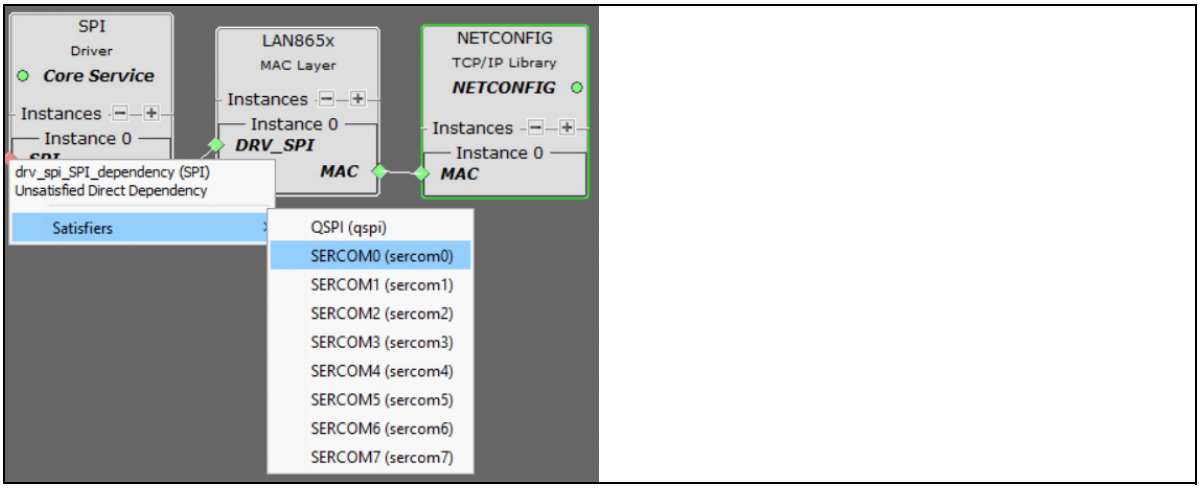


2.9 Add Dependency Components

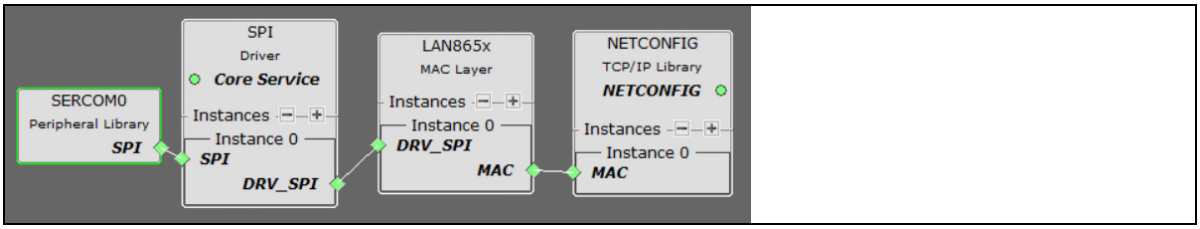
1. Minimize the “TCP/IP Configuration” window.
2. Click the “Project Graph” tab.
3. To Add “DRV_SPI” to satisfy the “LAN865x_0” dependency,
   a) select “DATA LINK LAYER” from the “View” drop-down menu,
   b) right click the red diamond next to “LAN865x_0”, to find the satisfier and
   c) select “SPI (drv_spi)”.



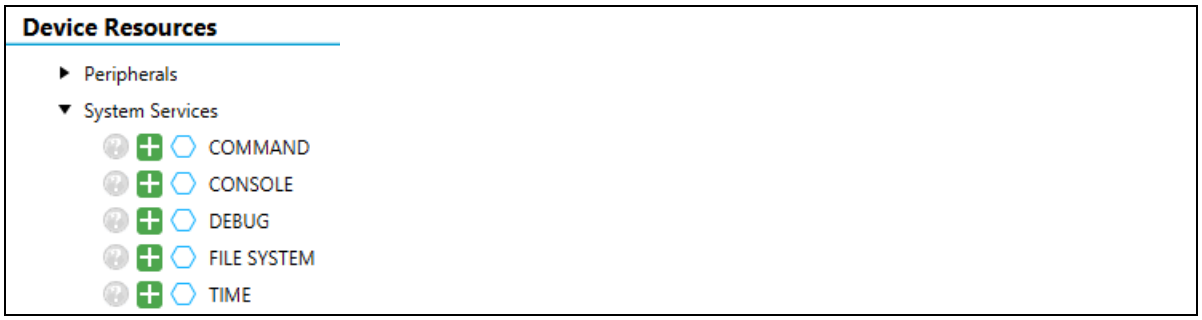
4. To Add “SERCOM0” to satisfy the “SPI” dependency,
   a) right click the red diamond next to “SPI”, to find the satisfier and
   b) select “SERCOM0”.
   (***Hint:*** *This is only valid for the SAM E54 Curiosity Ultra Board* [4]*. For other boards a different SERCOMx might be used.)*



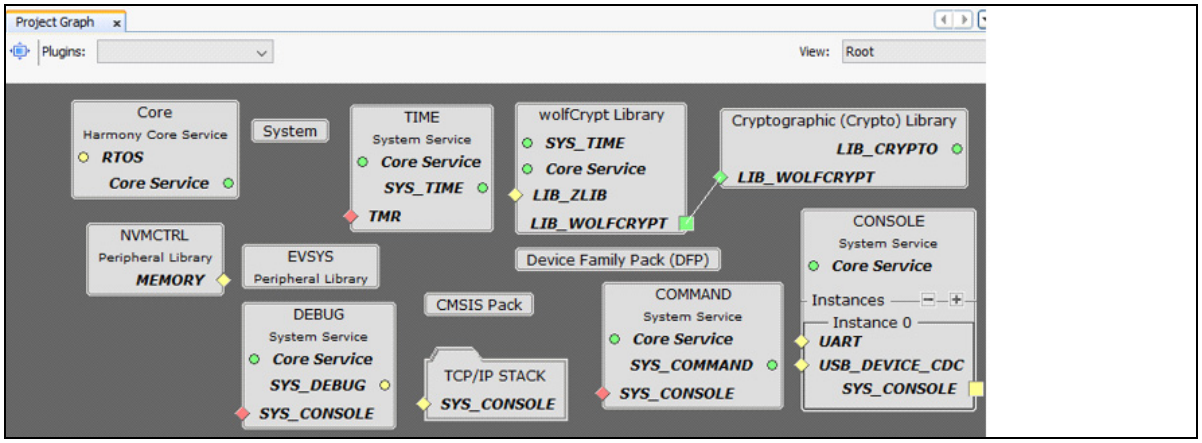
The “Project Graph” should look like shown below.



5. In the “Project Graph” window, select “Root” from the “View” drop-down menu.
6. Add “System Services” to satisfy the “TCPIP CORE” and the “TCPIP CMD” dependency.
   a) Go to the “Device Resource” tab.
   b) Double click “Harmony”.
   c) Double click “System Services”.
   d) Click on the respective “+” icons to add “COMMAND”, “CONSOLE”, “DEBUG” and “TIME” services to the demo.

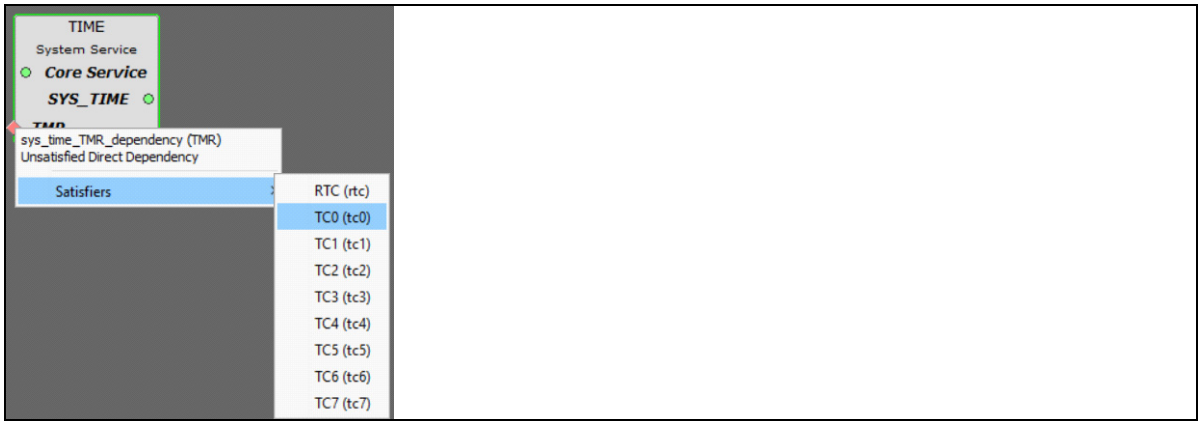


The “Root” layer should look like shown below.

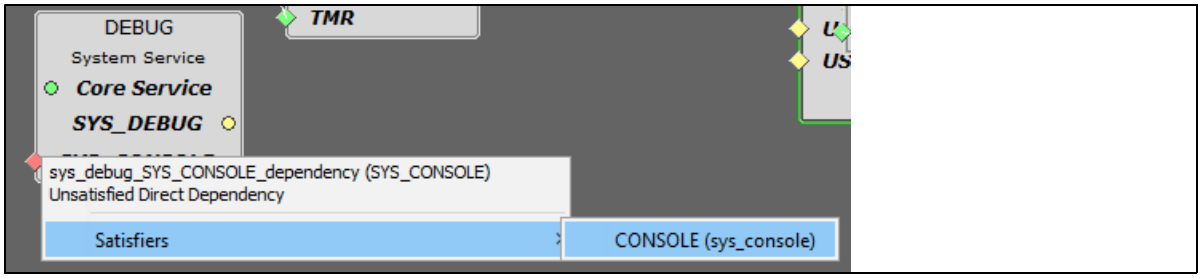


Note: Red and yellow diamond-shaped buttons indicate unsatisfied dependencies.

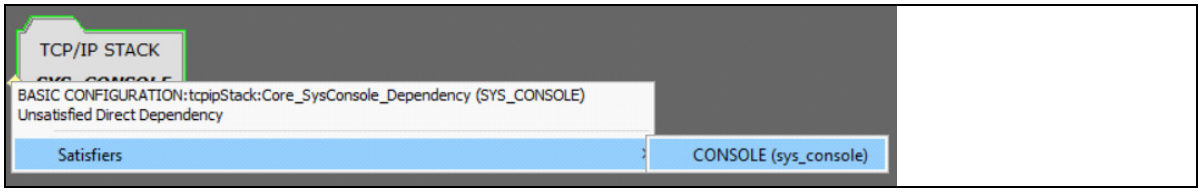
7. Add “TC0” to satisfy the “TMR” dependency.
   a) Right click the red diamond next to “TMR”, to find the satisfier.
   b) Select “TC0”.



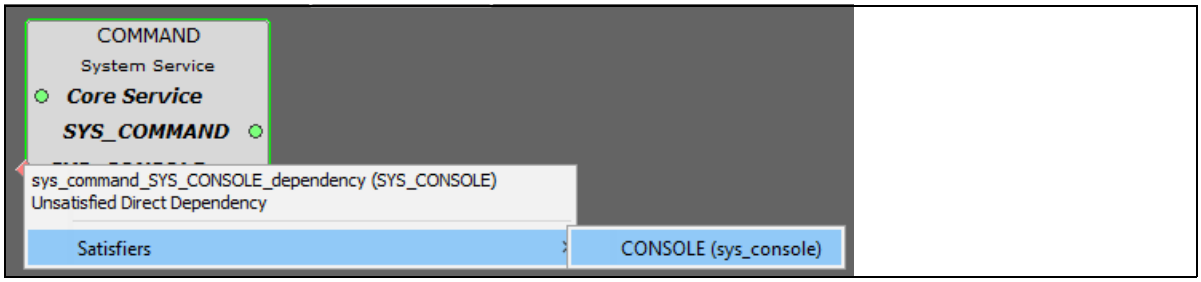
8. Add “CONSOLE” to satisfy the “DEBUG” dependency.
   a) Right click the red diamond next to “SYS_CONSOLE”, to find the satisfier.
   b) Select “CONSOLE”.



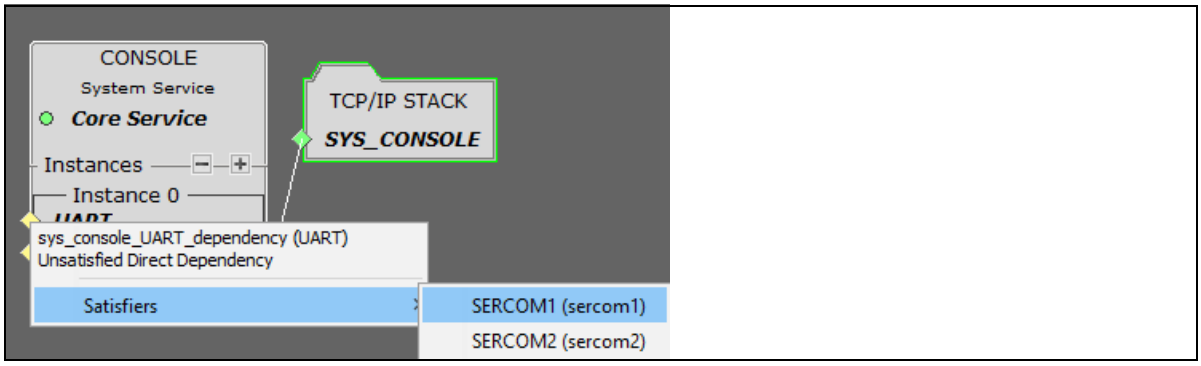
9. Add “CONSOLE” to satisfy the “TCP/IP STACK” dependency.
   a) Right click the yellow diamond next to “SYS_CONSOLE”, to find the satisfier.
   b) Select “CONSOLE”.



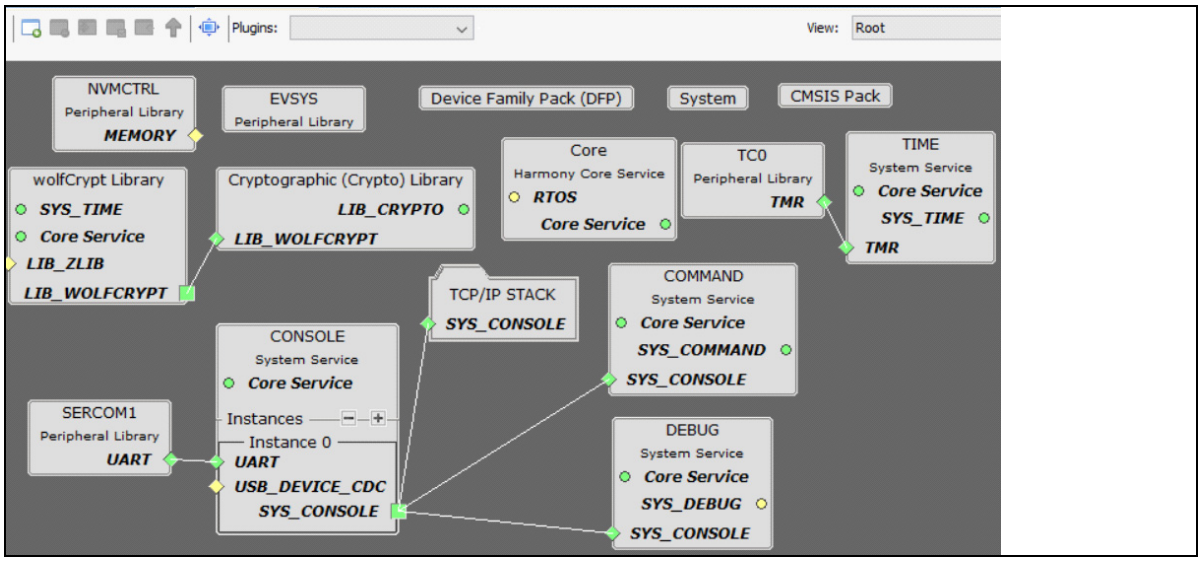
10. Add “CONSOLE” to satisfy the “COMMAND” dependency.
    a) Right click the red diamond next to “SYS_CONSOLE”, to find the satisfier.
    b) Select “CONSOLE”.



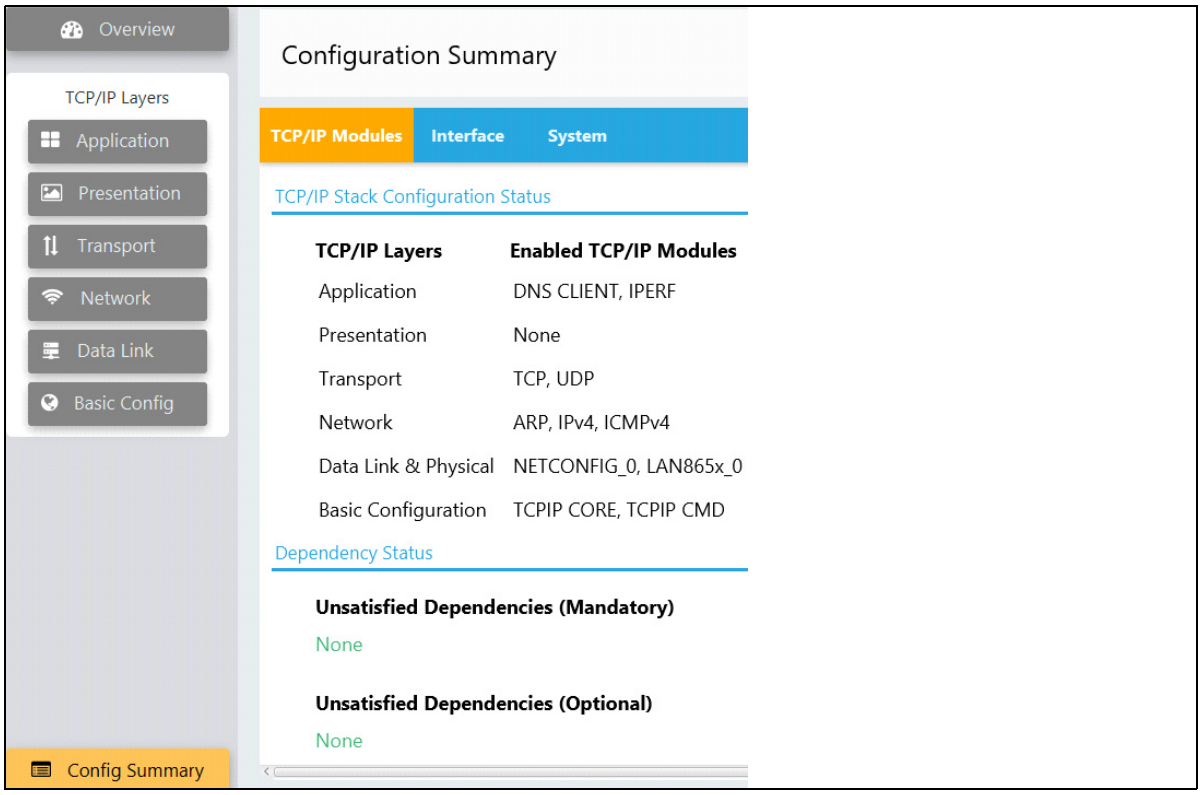
11. Add “SERCOM1” to satisfy the “UART” dependency.
    a) Right click the yellow diamond next to “UART”, to find the satisfier.
    b) Select “SERCOM1”.
    ***(Hint:** This is only valid for the SAM E54 Curiosity Ultra Board [4]. For other boards a different SERCOMx might be used.)*



Now, all dependency modules have been added. The “Project Graph” should look like shown below.



12. Open the “TCP/IP Configuration” window.
13. Check if all dependency are satisfied.

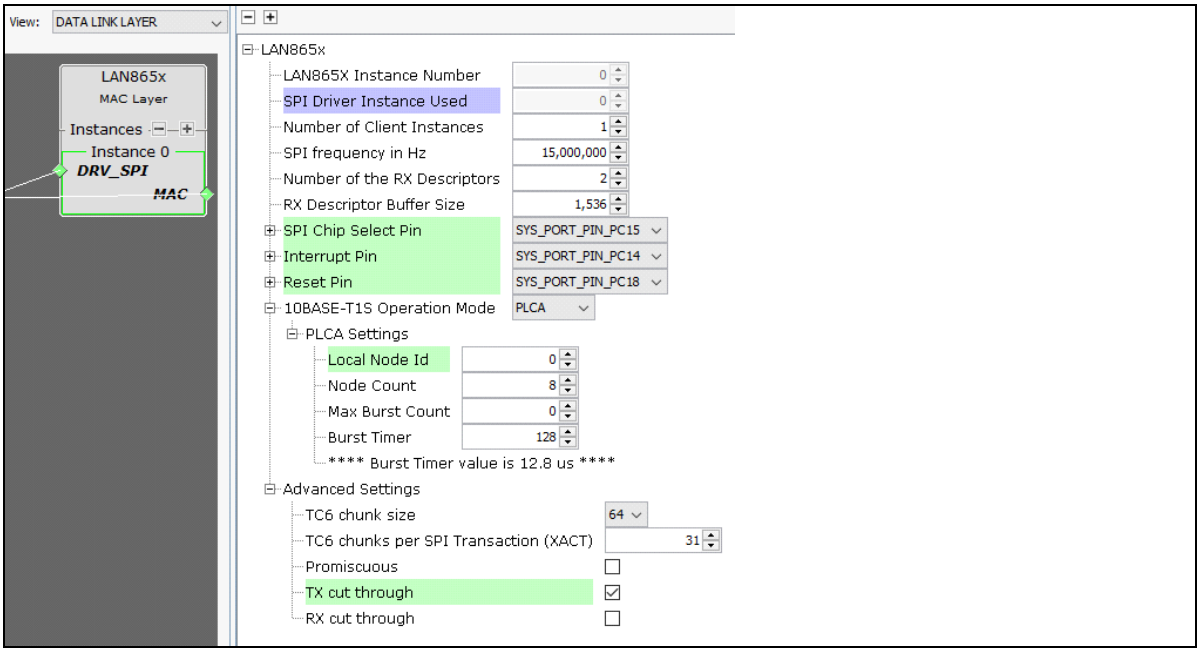


14. Close the TCP/IP Configuration window.

2.10 Add/Configure Additional Components

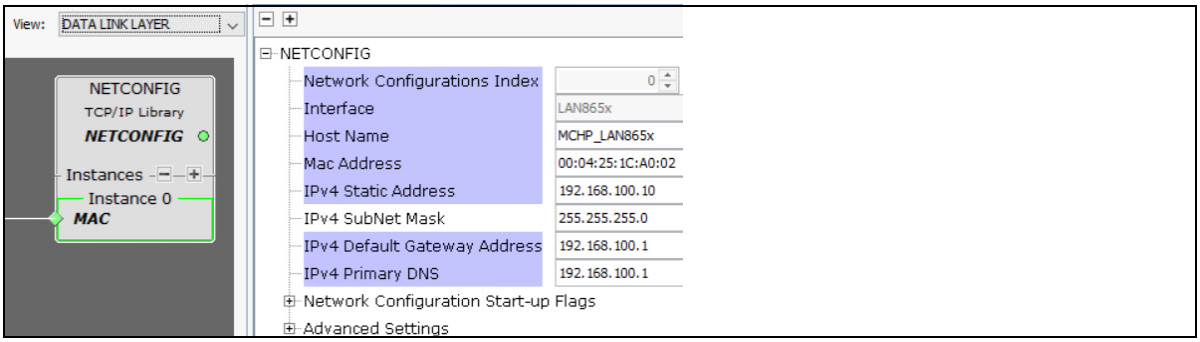
The demo implementation needs a few MPLAB Harmony components. In this section, we will add and configure the required components.

1. Configure the LAN865x to work in 10BASE-T1S mode.
   a) In the “Project Graph” window select “DATA LINK LAYER” from the “View” drop-down menu.
   b) Select “Instance 0” under “LAN865x”.
   Configuration Options appear on the right-hand side.
   c) Make the settings as shown below.
   Configure the PHY to operate in PLCA or CSMA/CD mode, as you wish.
   In case of PLCA mode, set “Local Node Id” and “Node Count” as shown below.
   The “Advanced Settings” section provides options to Configure TC6-specific configurations.
   Note: The “Local Node Id” must be unique for each node. In the example below the “Node Count” is 8; this means the value of the “Local Node Id” must be in the range of 0...7.



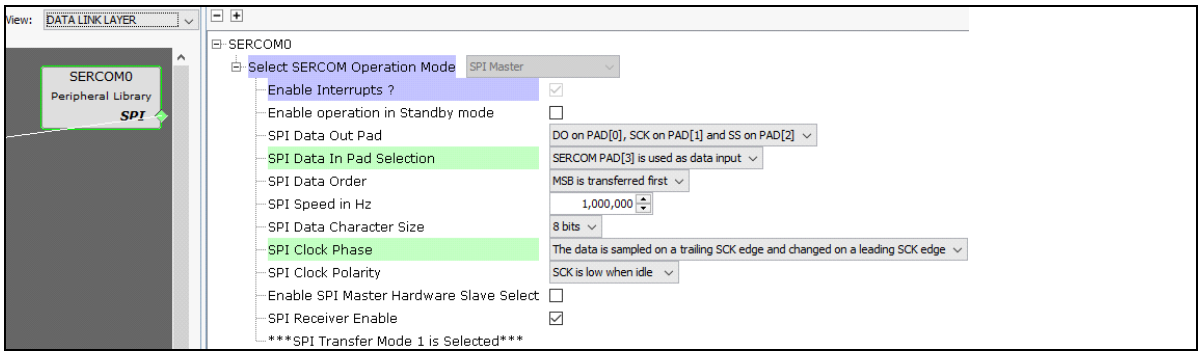
2. To modify the IP address or MAC address, select “Instance 0” of “NETCONFIG”.

   In the Configuration window you can see the IP Addresses and MAC Address. You can modify them according to your needs. Make sure IP Addresses and MAC Address must be unique, at least within the local network. An example is given below.



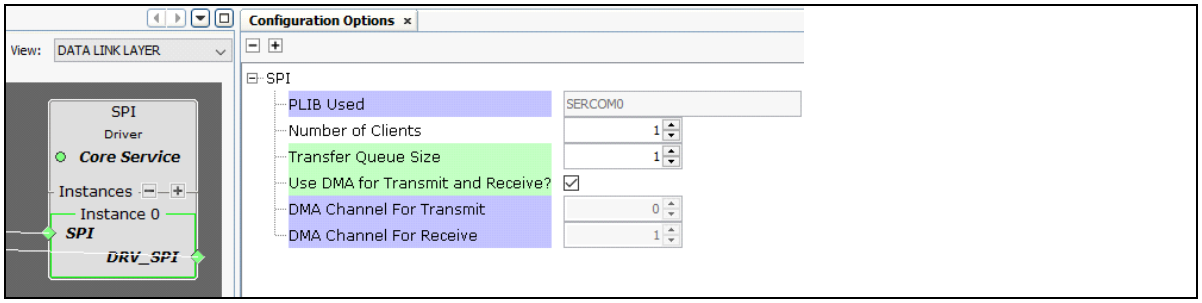
3. Select “SERCOM0”.
4. Set
   a) “SPI Data In Pad Selection” and
   b) “SPI Clock Phase”

   as shown below.

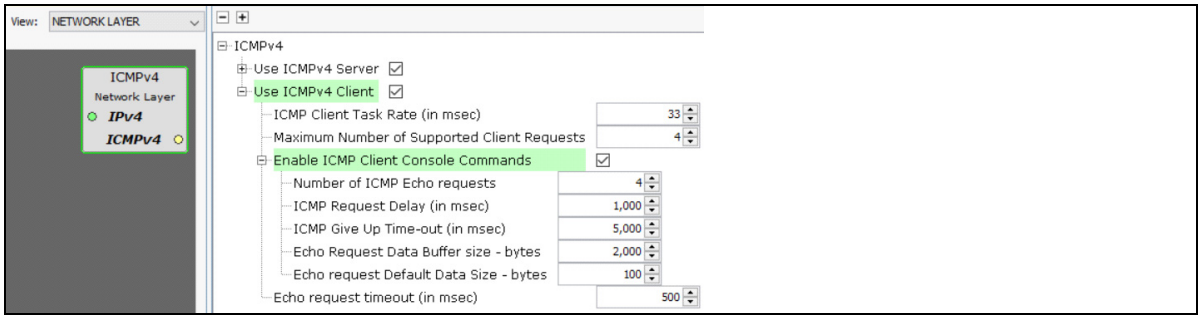


5. Select “SPI Driver” and ensure “SPI Driver” is in “Asynchronous” mode.

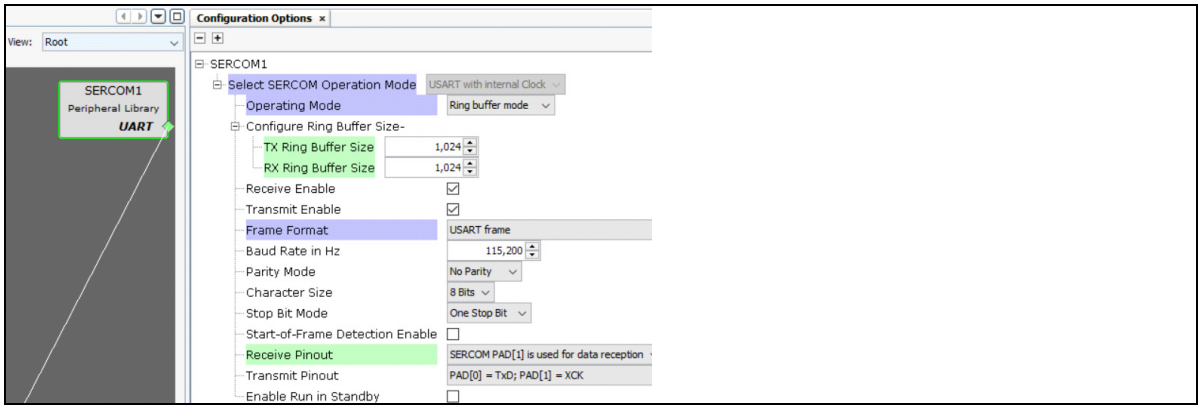
6. Select “Instance 0” of “SPI Driver”.
   a) Set “Transfer Queue Size” to “1”.
   b) Check the “Use DMA for Transmit and Receive?” option to enable DMA.



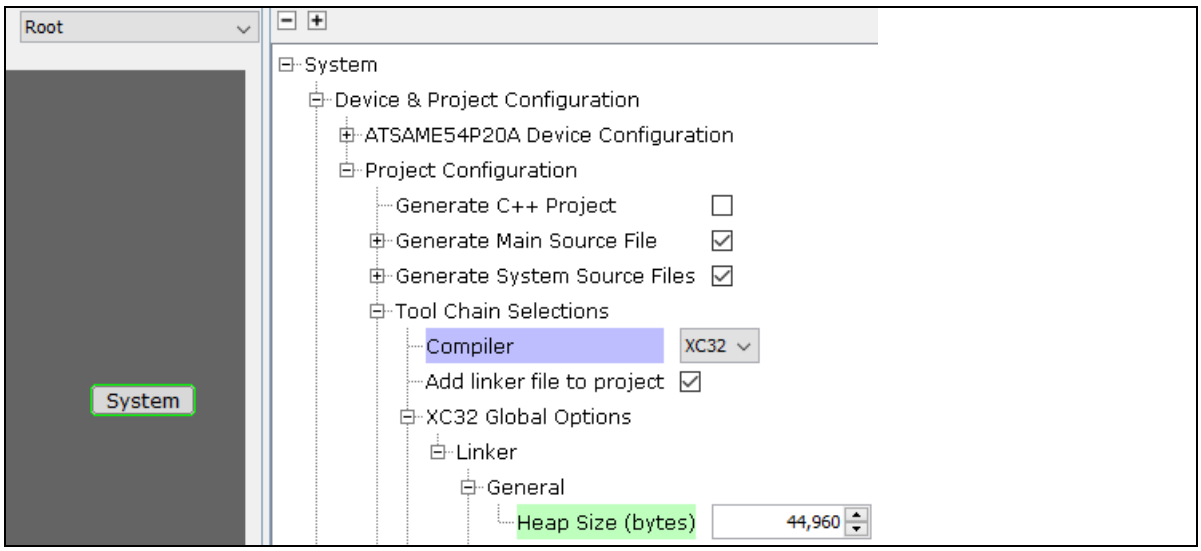
7. To enable ping operation from the SAM E54 Curiosity Ultra Development Board [4],
   a) in the “Project Graph” select “NETWORK LAYER” from the “View” drop-down menu,
   b) select “ICMPv4” and
   c) make the settings as shown below.



8. Configure “SERCOM1”.
   a) In the “Project Graph” select “Root” from the “View” drop-down menu.
   b) Select “SERCOM1”.
   c) Make the settings as shown below.



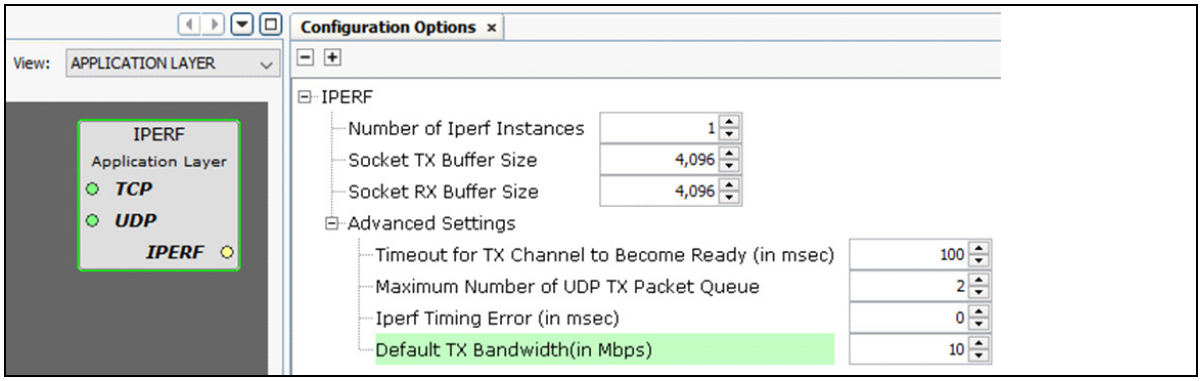
9. In the “Project Graph” window, select “System”.
10. In the “Configuration Options” window, set the “Heap Size (bytes)” to “44960”.
    The “Project Graph” window should look like shown below.



11. Configure the TX bandwidth for IPERF.

    In the “Project Graph” window:

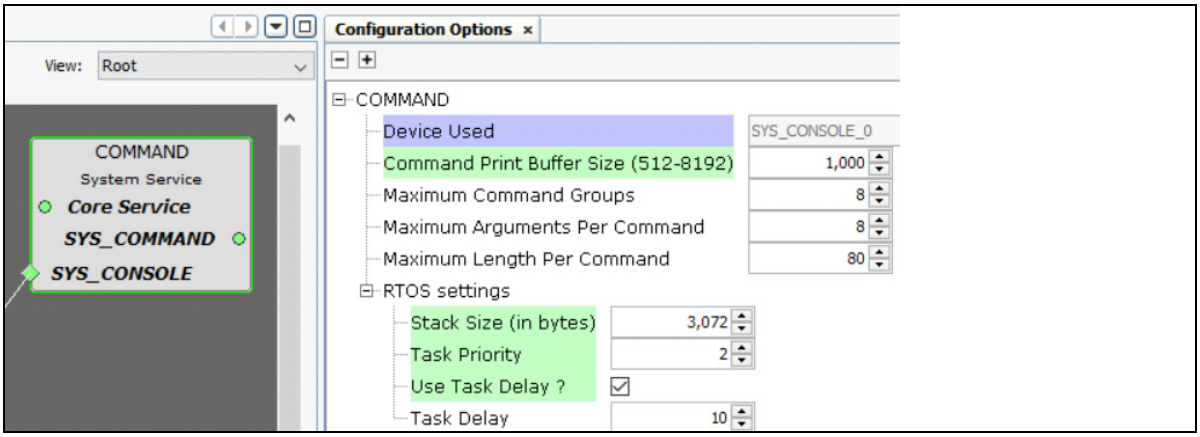
    a) Select “APPLICATION LAYER” from the “View” drop-down menu.
    b) Select “IPERF”.
    c) Make the settings as shown below.



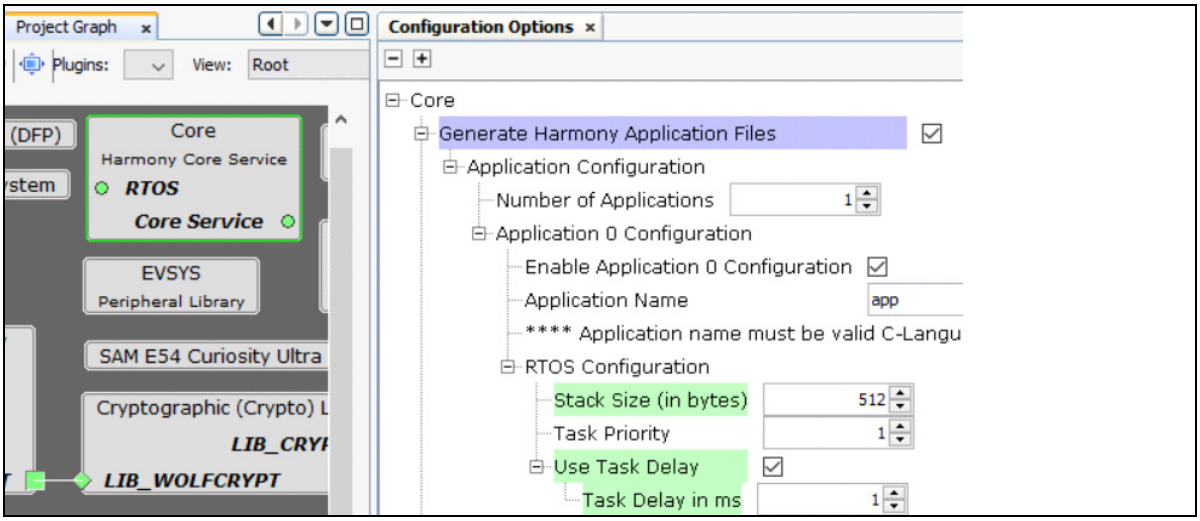
2.11 FreeRTOS Configuration

Note: The configuration settings described in this section are required only for a FreeRTOS project.

1. In the “Project Graph” window:
   a) Select “Root” from the “View” drop-down menu.
   b) Select “Command”.
2. In the “Configuration Options” window, set the “Stack Size (in bytes)” to “3072”.



3. Configure “Core” as shown below.

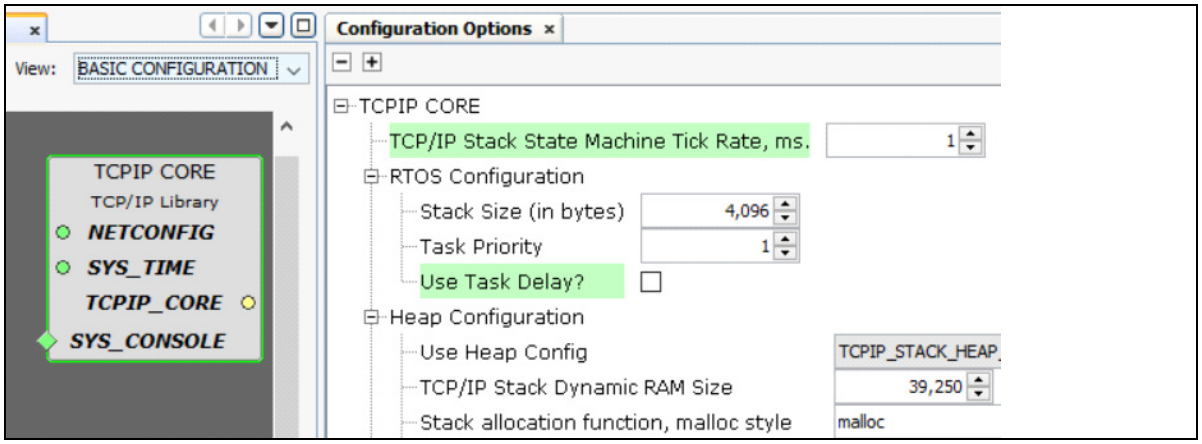


4. Configure “TCPIP Core” as shown below.

   In the “Project Graph” window:

   a) Select “BASIC CONFIGURATION” from the “View” drop-down menu.

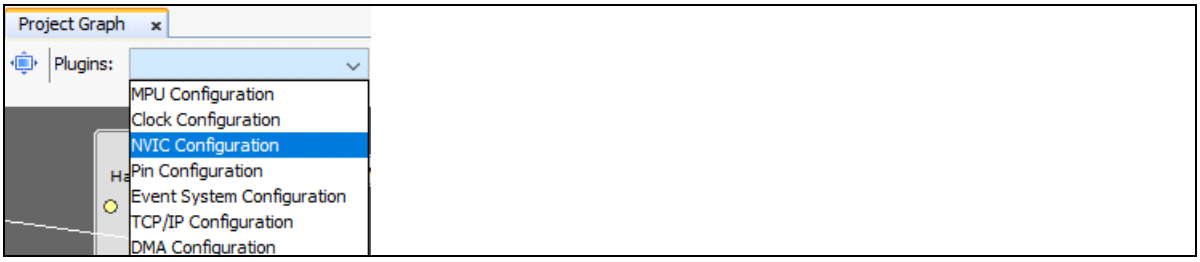
   b) Configure the RTOS settings as shown below.



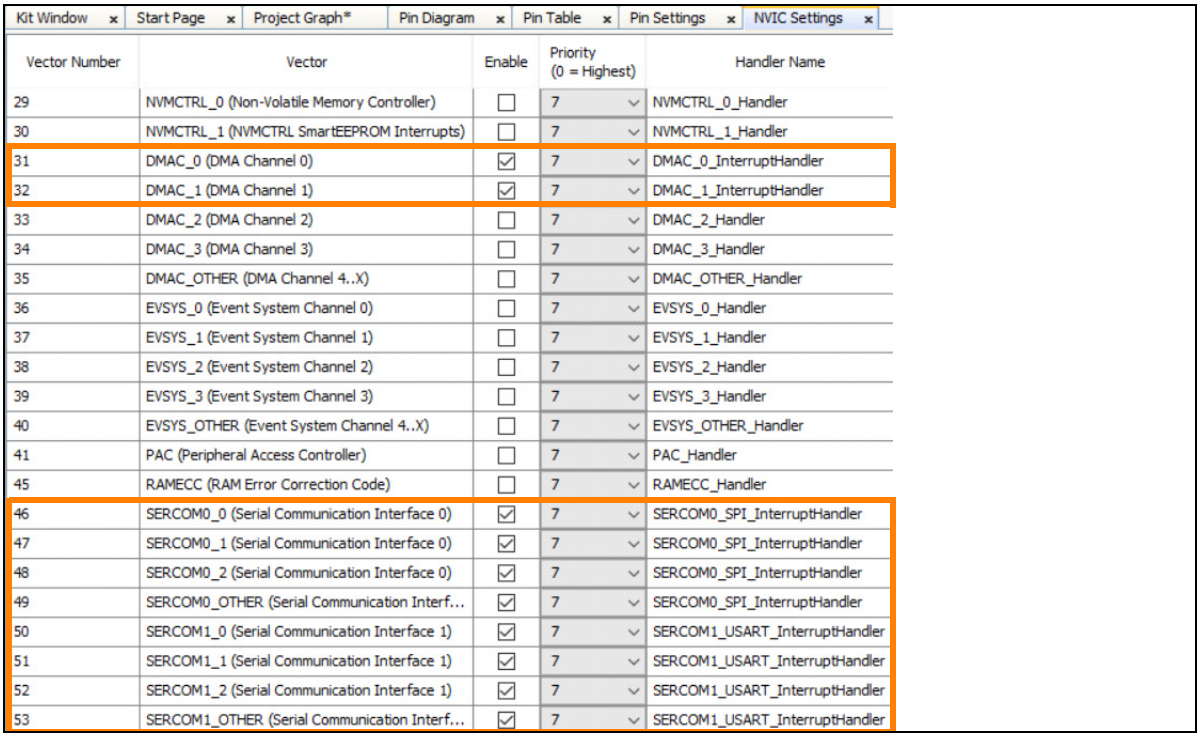
2.12 NVIC Configuration

The Nested Vectored Interrupt Controller (NVIC) is for the configuration of the interrupts. This configuration is applicable for SAM devices only.

1. In the “Plugins” drop-down list, select “NVIC Configuration”.



2. Make sure “DMAC” (#31 and #32) and “SERCOM” (#46-#53) interrupt handlers are enabled.

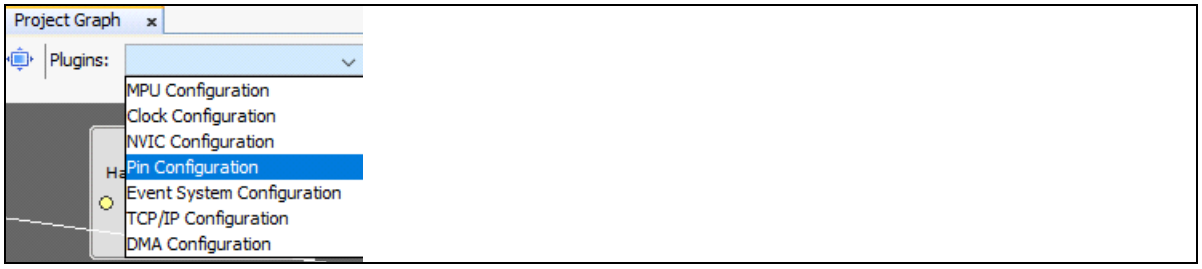


Vector Number	Vector	Enable	Priority (0 = Highest)	Handler Name
29	NVMCTRL_0 (Non-Volatile Memory Controller)	☐	7	NVMCTRL_0_Handler
30	NVMCTRL_1 (NVMCTRL SmartEEPROM Interrupts)	☐	7	NVMCTRL_1_Handler
31	DMAC_0 (DMA Channel 0)	☑	7	DMAC_0_InterruptHandler
32	DMAC_1 (DMA Channel 1)	☑	7	DMAC_1_InterruptHandler
33	DMAC_2 (DMA Channel 2)	☐	7	DMAC_2_Handler
34	DMAC_3 (DMA Channel 3)	☐	7	DMAC_3_Handler
35	DMAC_OTHER (DMA Channel 4..X)	☐	7	DMAC_OTHER_Handler
36	EVSYS_0 (Event System Channel 0)	☐	7	EVSYS_0_Handler
37	EVSYS_1 (Event System Channel 1)	☐	7	EVSYS_1_Handler
38	EVSYS_2 (Event System Channel 2)	☐	7	EVSYS_2_Handler
39	EVSYS_3 (Event System Channel 3)	☐	7	EVSYS_3_Handler
40	EVSYS_OTHER (Event System Channel 4..X)	☐	7	EVSYS_OTHER_Handler
41	PAC (Peripheral Access Controller)	☐	7	PAC_Handler
45	RAMECC (RAM Error Correction Code)	☐	7	RAMECC_Handler
46	SERCOM0_0 (Serial Communication Interface 0)	☑	7	SERCOM0_SPI_InterruptHandler
47	SERCOM0_1 (Serial Communication Interface 0)	☑	7	SERCOM0_SPI_InterruptHandler
48	SERCOM0_2 (Serial Communication Interface 0)	☑	7	SERCOM0_SPI_InterruptHandler
49	SERCOM0_OTHER (Serial Communication Interf...	☑	7	SERCOM0_SPI_InterruptHandler
50	SERCOM1_0 (Serial Communication Interface 1)	☑	7	SERCOM1_USART_InterruptHandler
51	SERCOM1_1 (Serial Communication Interface 1)	☑	7	SERCOM1_USART_InterruptHandler
52	SERCOM1_2 (Serial Communication Interface 1)	☑	7	SERCOM1_USART_InterruptHandler
53	SERCOM1_OTHER (Serial Communication Interf...	☑	7	SERCOM1_USART_InterruptHandler

2.13 Pin Configuration

The I/O pins on the device can have alternate functions other than the general purpose I/O function. Here, we will configure the functionality of the I/O pins required for this demo.

1. In the “Plugins” drop-down list, select “Pin Configuration”.



2. Configure the pins for “SERCOM1” as shown below.

111	PC27	USART_TX	SERCOM1_PAD0	Digital	High Impedance	n/a	☐	☐	NORMAL
112	PC28	USART_RX	SERCOM1_PAD1	Digital	High Impedance	n/a	☐	☐	NORMAL

3. Configure pins for the “LAN865x”.

Note: Check all columns for correct settings and make sure the LAN865x pins PC15 (Chip Select), PB24 (MOSI), PB25 (SCK) and PC25 (MISO) are set to “Drive Strength” “STRONG”.

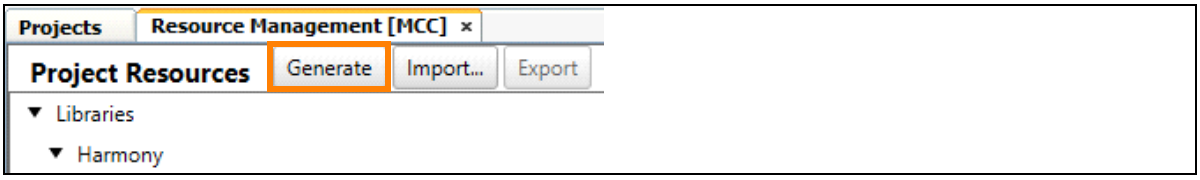
Pin Number	Pin ID	Custom Name	Function	Mode	Direction	Latch	Pull Up	Pull Down	Drive Strength
58	PC14	LAN865x_INT	GPIO	Digital	In	High	☑	☐	NORMAL
59	PC15	LAN865x_CS	GPIO	Digital	Out	High	☐	☐	STRONG
60	PA12		GMAC_GRX1	Digital	High Impedance	n/a	☐	☐	NORMAL
61	PA13		GMAC_GRX0	Digital	High Impedance	n/a	☐	☐	NORMAL
62	PA14		GMAC_GTXCK	Digital	High Impedance	n/a	☐	☐	NORMAL
63	PA15		GMAC_GRXER	Digital	High Impedance	n/a	☐	☐	NORMAL
64	GND			Digital	High Impedance	Low	☐	☐	NORMAL
65	VDDIO			Digital	High Impedance	Low	☐	☐	NORMAL
66	PA16		Available	Digital	High Impedance	Low	☐	☐	NORMAL
67	PA17		GMAC_GTXEN	Digital	High Impedance	n/a	☐	☐	NORMAL
68	PA18		GMAC_GTX0	Digital	High Impedance	n/a	☐	☐	NORMAL
69	PA19		GMAC_GTX1	Digital	High Impedance	n/a	☐	☐	NORMAL
70	PC16		Available	Digital	High Impedance	Low	☐	☐	NORMAL
71	PC17		Available	Digital	High Impedance	Low	☐	☐	NORMAL
72	PC18	LAN865x_RESET	GPIO	Digital	Out	High	☐	☐	NORMAL
73	PC19		Available	Digital	High Impedance	Low	☐	☐	NORMAL

Pin Number	Pin ID	Custom Name	Function	Mode	Direction	Latch	Pull Up	Pull Down	Drive Strength
100	PB24	LAN865x_MOSI	SERCOM0_PAD0	Digital	High Impedance	n/a	☐	☐	STRONG
101	PB25	LAN865x_SCK	SERCOM0_PAD1	Digital	High Impedance	n/a	☐	☐	STRONG
102	PB26		Available	Digital	High Impedance	Low	☐	☐	NORMAL
103	PB27		Available	Digital	High Impedance	Low	☐	☐	NORMAL
104	PB28		Available	Digital	High Impedance	Low	☐	☐	NORMAL
105	PB29		Available	Digital	High Impedance	Low	☐	☐	NORMAL
106	GND			Digital	High Impedance	Low	☐	☐	NORMAL
107	VDDIO			Digital	High Impedance	Low	☐	☐	NORMAL
108	PC24		Available	Digital	High Impedance	Low	☐	☐	NORMAL
109	PC25	LAN865x_MISO	SERCOM0_PAD3	Digital	High Impedance	n/a	☐	☐	STRONG
110	PC26		Available	Digital	High Impedance	Low	☐	☐	NORMAL

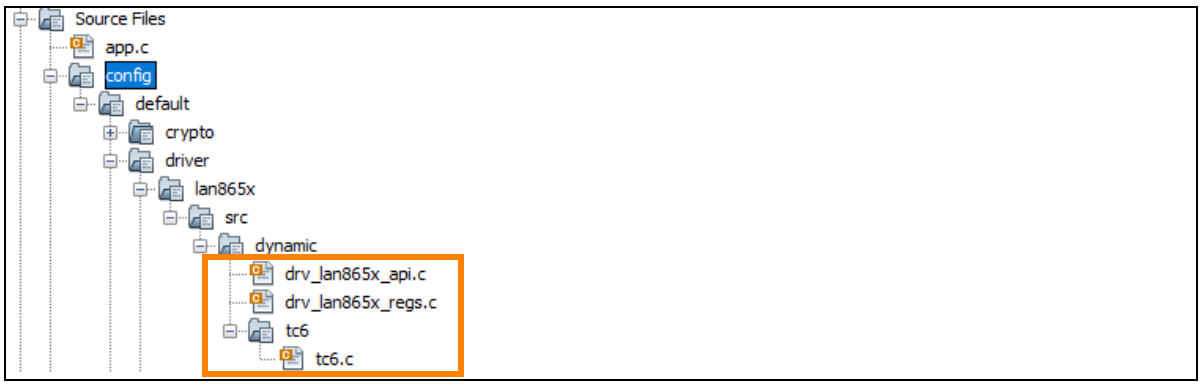
2.14 Code Generation

Once MPLAB Harmony components have been added by using the MCC “Project Graph”, it is time to generate the source files based on the configurations.

1. To start the code generation, click the **Generate** button (see below).

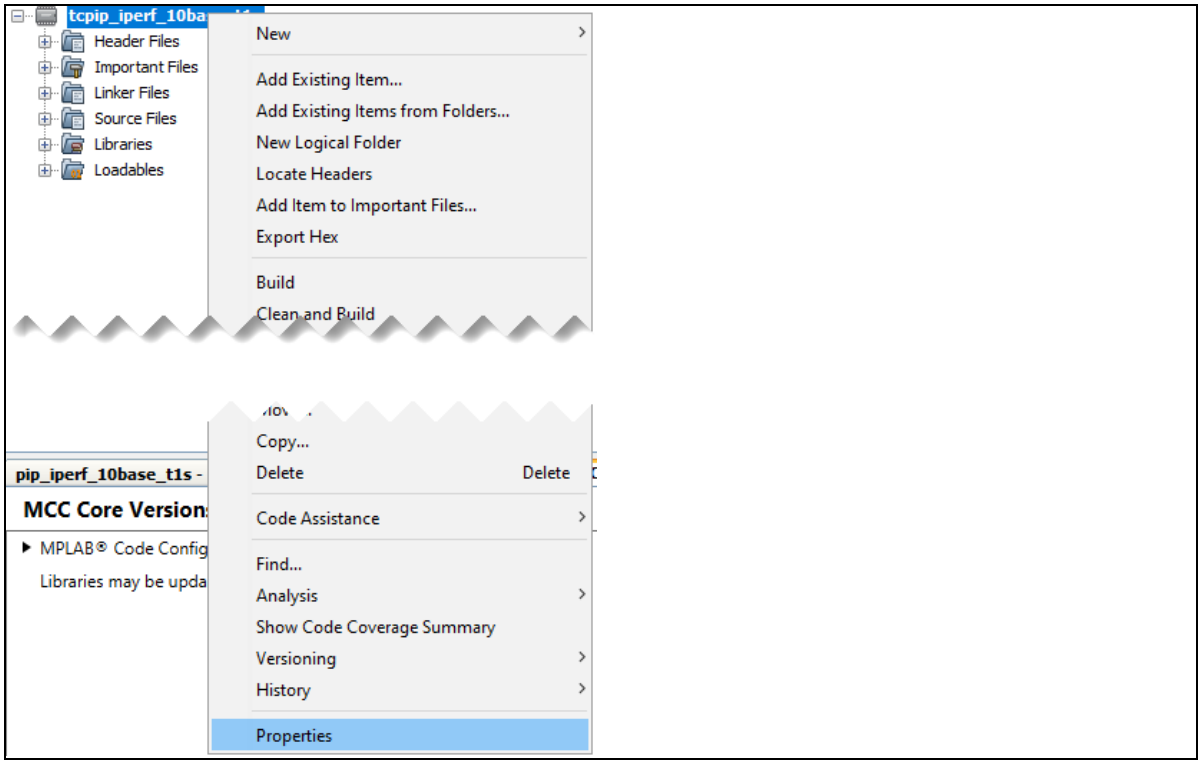


2. After successful code generation, verify in the “Projects” tab if the files *drv_lan865x_api.c*, *drv_lan865x_regs.c* and *tc6.c* have been added.
For this purpose, click the “Projects” tab on the left-hand side.
The result should look as follows:



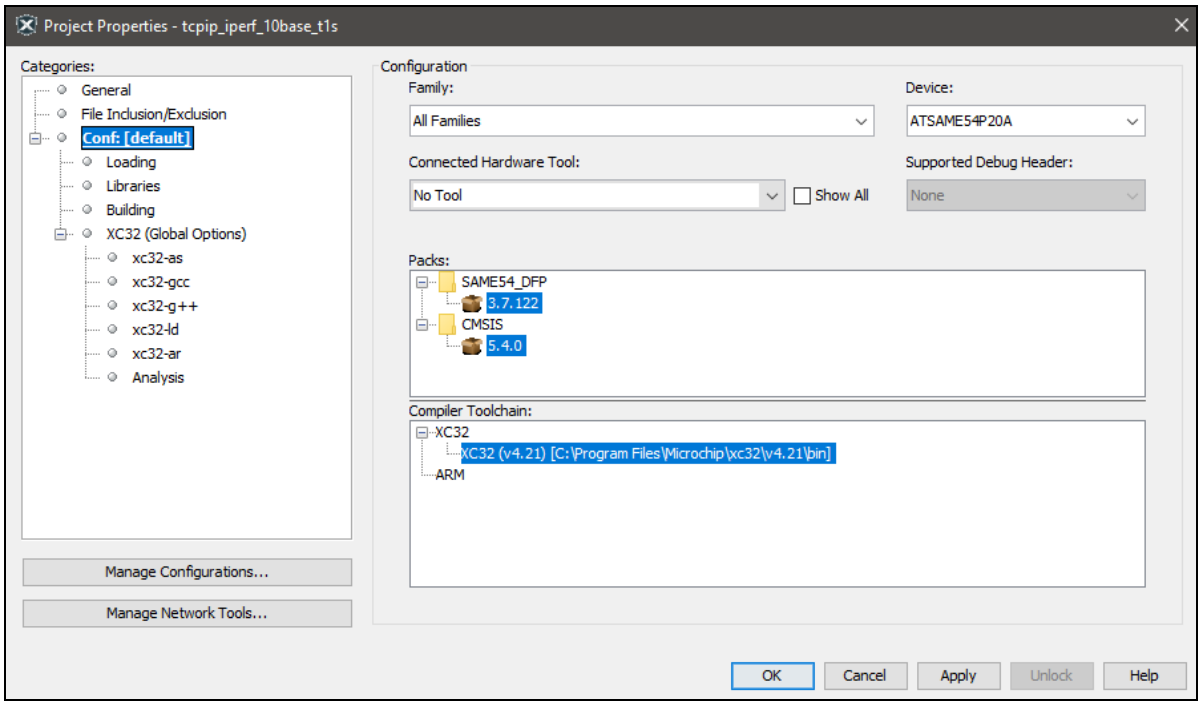
2.15 Build the Application

1. Open the project “Properties” by right clicking on the project name “tcpip_iperf_10base_t1s”.

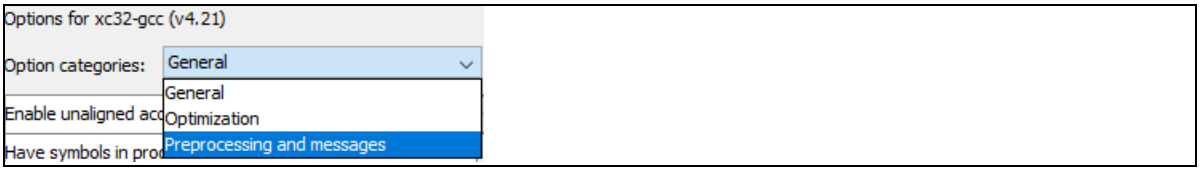


2. Make sure the “XC32” compiler tool chain is selected.

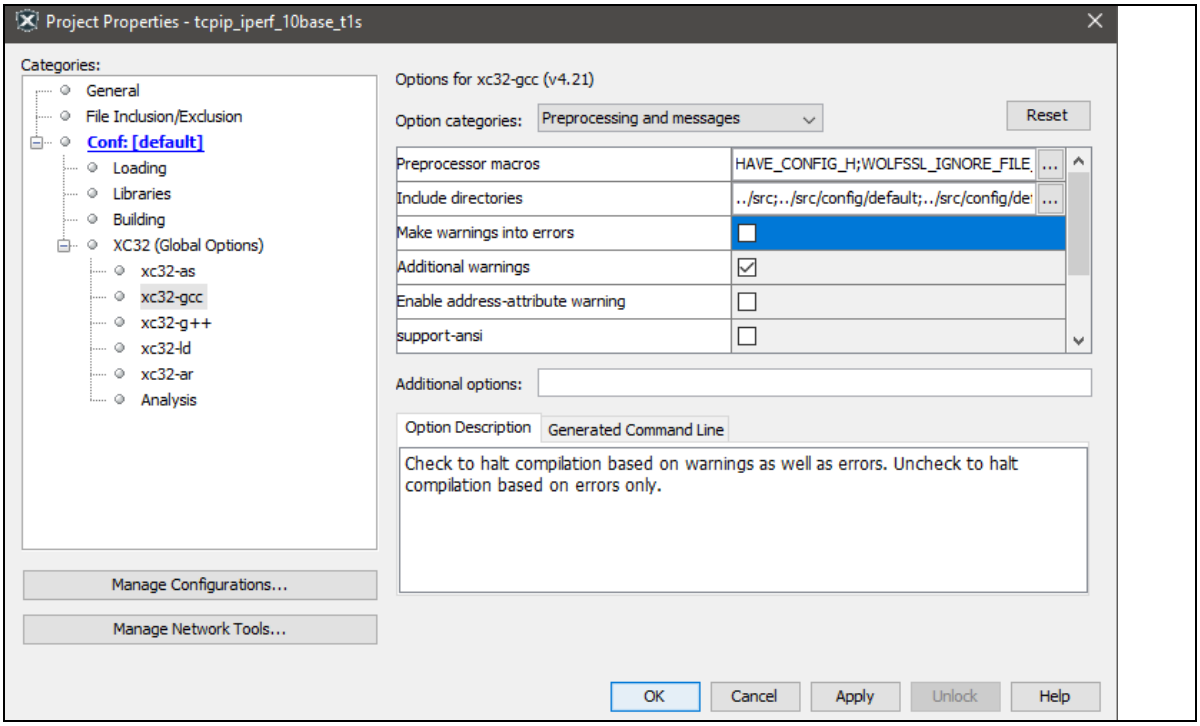
   Note: Version numbers may differ. Therefore, always use the latest versions.



3. Follow the steps below to disable the feature that turns warnings into errors.
   a) In the “Project Properties” window select “xc32-gcc”.
   b) From the “Option categories” drop down list select “Preprocessing and messages”.

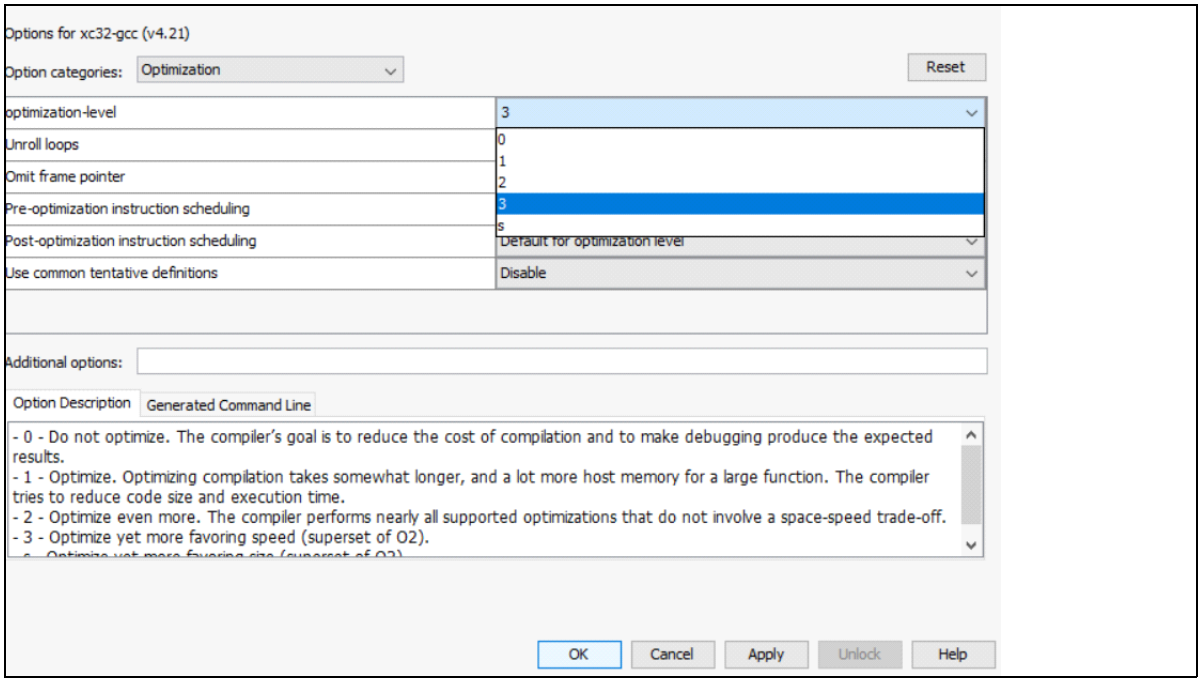


   c) Uncheck “Make warnings into errors”.

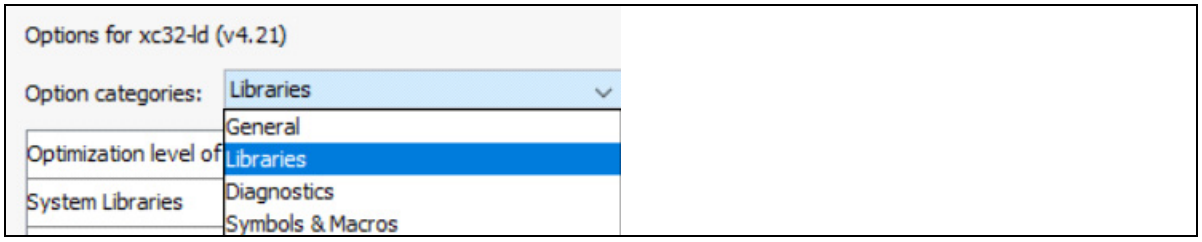


4. From the “Option categories” drop down list, select “Optimization”.
5. Enable optimization Level “3” for the FreeRTOS project.

   Note: Level “2” and “3” usage requires a paid compiler license that needs to be acquired.

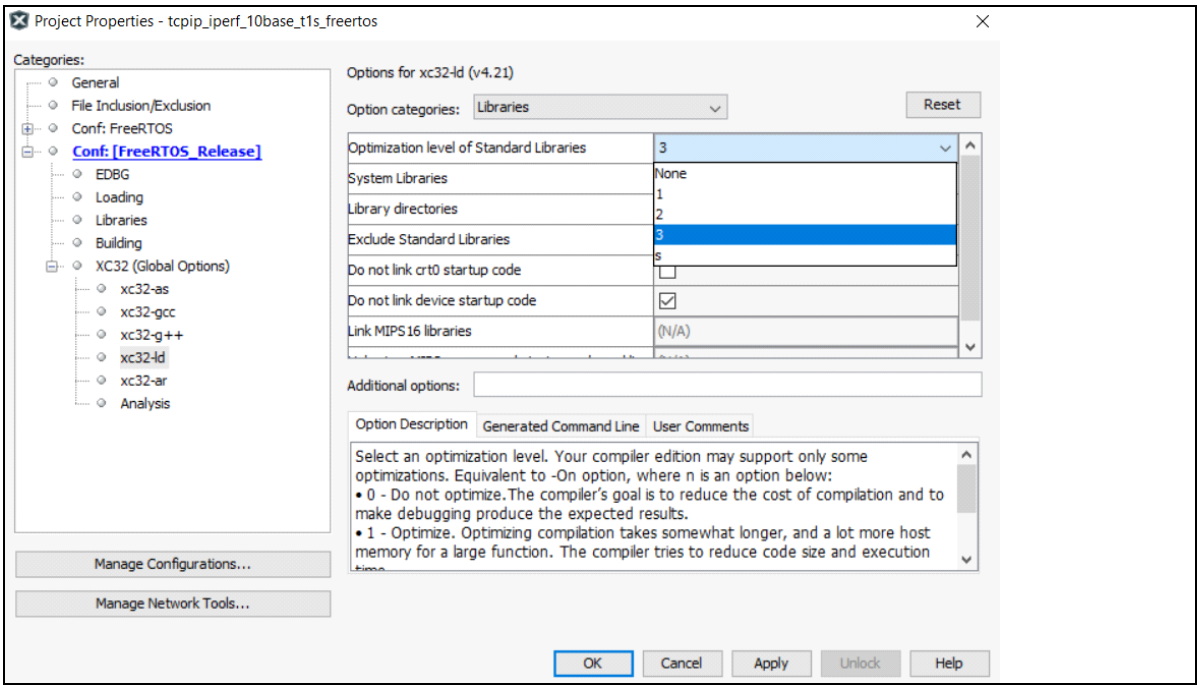


6. In the “Project Properties” window, select “xc32-ld”.
7. From the “Option categories” drop down list, select “Libraries”.



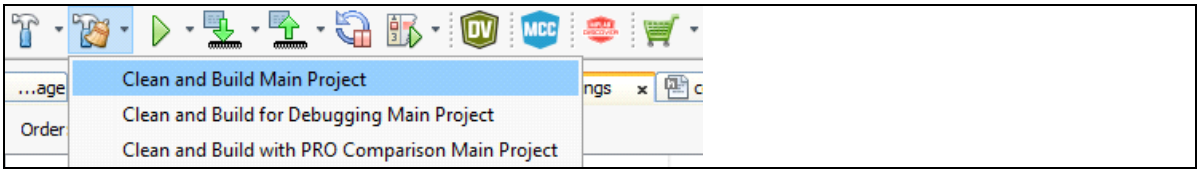
8. Enable optimization Level “3” for the FreeRTOS project.

   Note: Level “2” and “3” usage requires a paid compiler license that needs to be acquired.

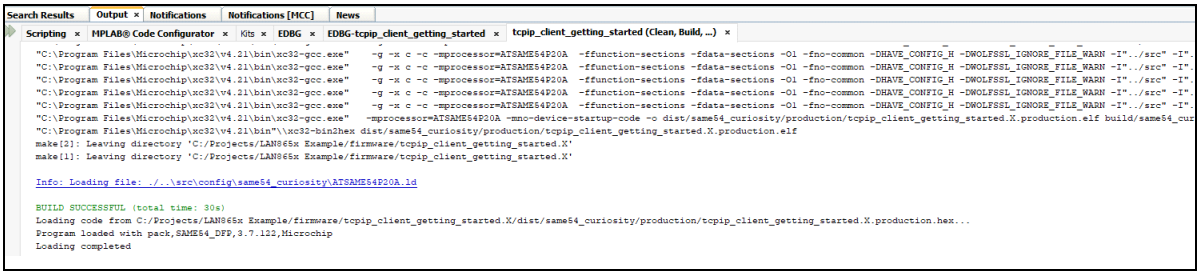


9. Click **OK**.

10. Build the application by clicking on “Clean and Build Main Project”.



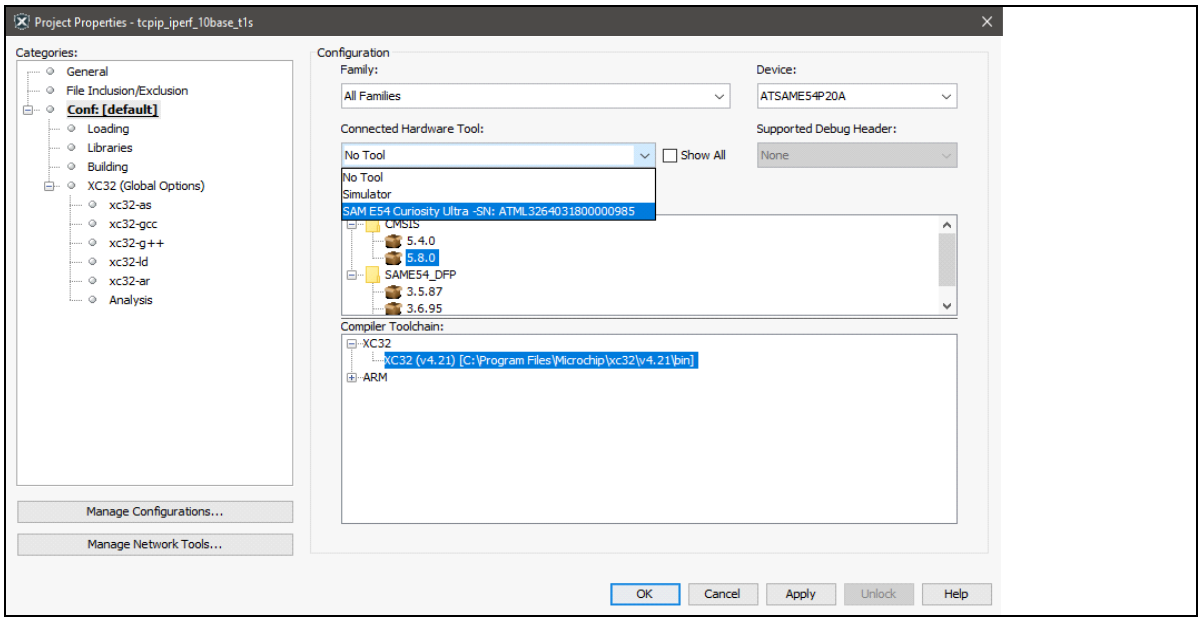
11. Verify the build results. Make sure that the build was successful.



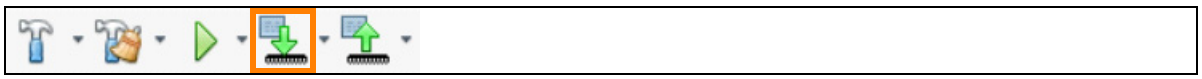
2.16 Program the Application

Now, the project is ready to be programmed on the hardware board. Refer to the hardware user's guide to see the different programming options supported.
The SAM E54 Curiosity Ultra Development Board [4] can be programmed using the on-board Embedded Debugger (EDBG).

1. Connect a Micro-USB cable from the computer to the DEBUG USB port on the SAM E54 Curiosity Ultra Development Board.
2. Open the "Project Properties".
3. Under Connected hardware, select "SAM E54 Curiosity Ultra".
4. Click **OK**.

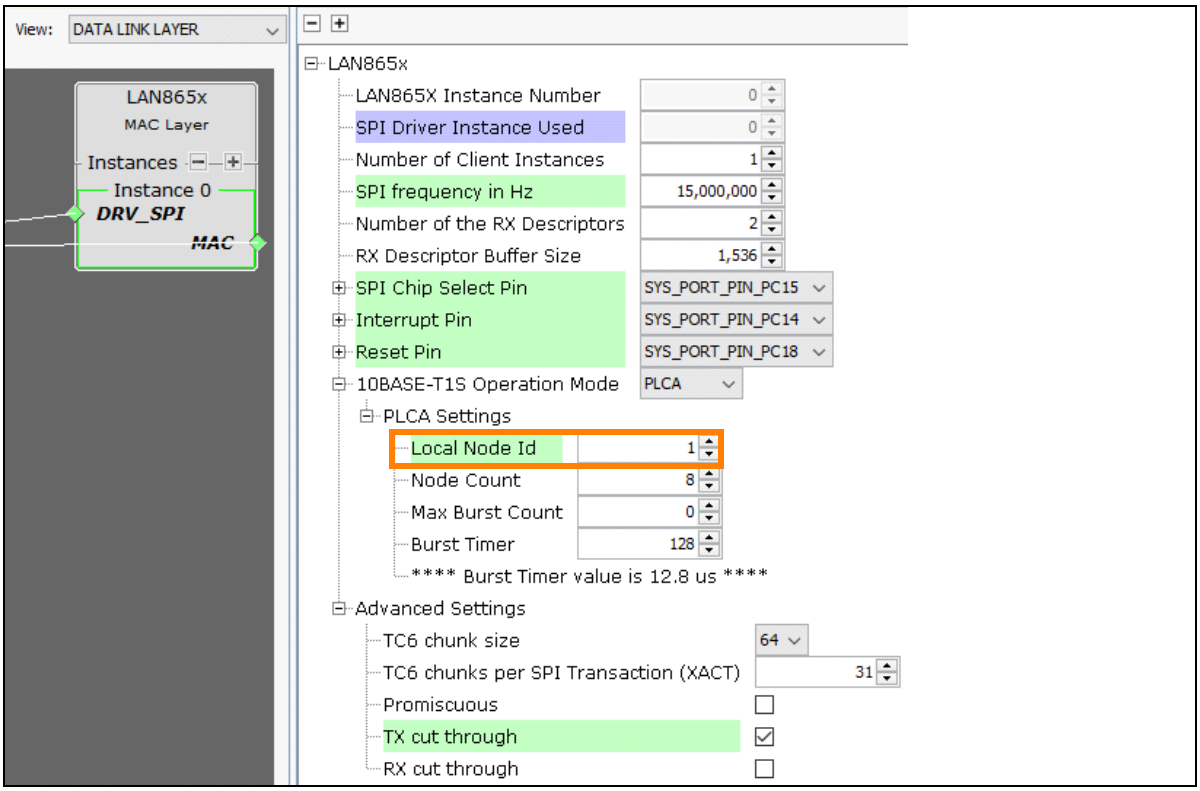


5. Program the application by clicking on the **Make and Program Device Main Project** button (see below).

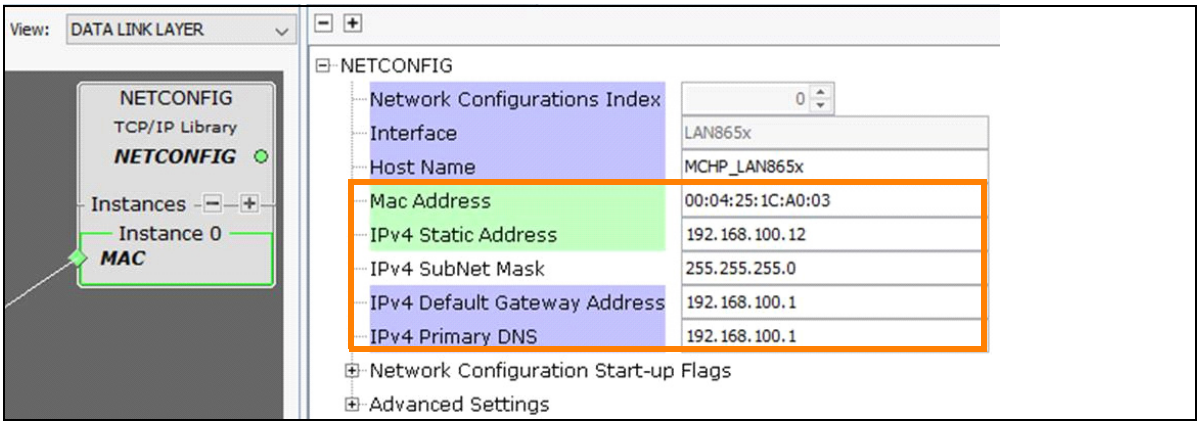


6. Make sure that the programming was successful.

7. Program the next SAME54 board to establish the two node 10BASE-T1S setup.
   Make sure to modify Node ID, IP Addresses and MAC Address of the second node.
   a) Set the Node ID of the second device as shown below.



   b) Set the IP Addresses and MAC Address of the second device as shown below.



   c) Under the “Resource Management” tab, click the **Generate** button to reflect relevant changes in the code.

8. Follow steps 1-6 to program the application on the second node.

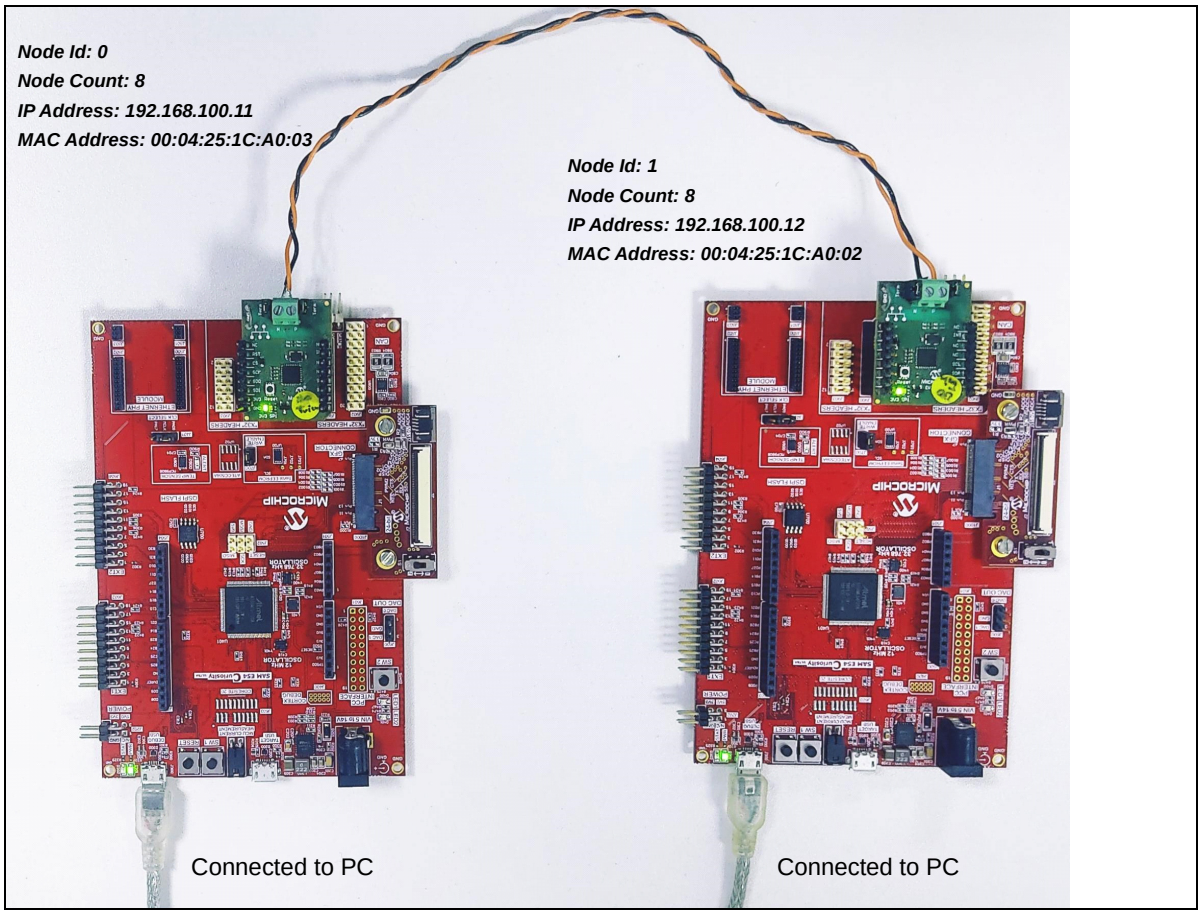
2.17 Test the Application

Testing the application is done by using different console commands.

Before you start, do the following:

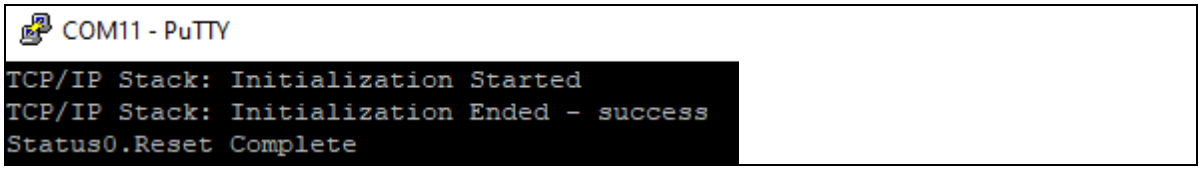
1. Establish a physical wiring with another 10BASE-T1S node as shown in the hardware setup picture below.

   Note: Both ends of the UTP cable must be connected to the PN port of the LAN865x and properly terminated with the jumper.
2. Make sure that PLCA settings are correct.
3. Make sure that Node Id, IP Addresses and MAC Addresses of the two nodes are different.

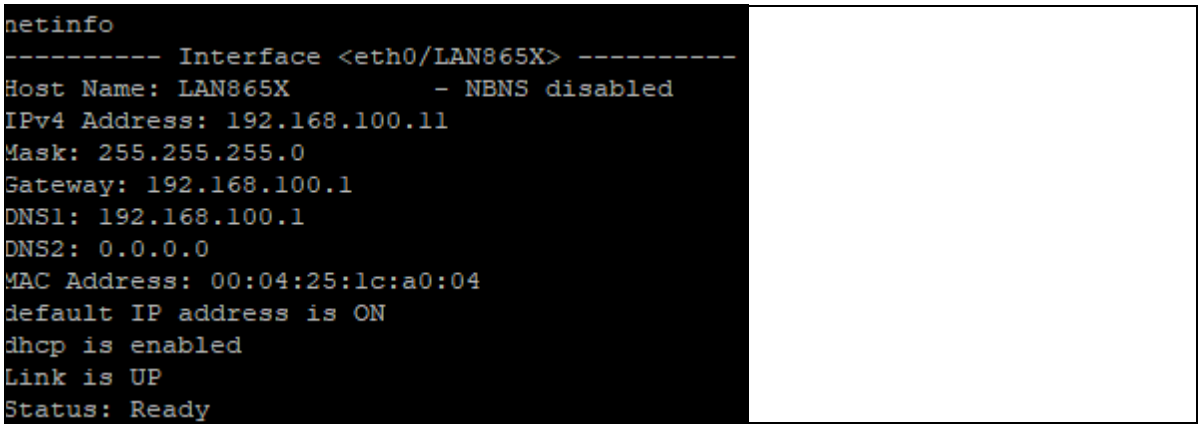


There will be a virtual COM port enumerated when a Micro-USB cable from the computer is connected to the DEBUG USB port on the SAM E54 Curiosity Ultra Development Board [4].

1. Open a terminal application (e.g., PuTTY).
2. Configure the baud rate for “115200”.
3. Press the **Reset** button on the SAM E54 Curiosity Ultra Development Board.
   The following messages become visible in the terminal window.



4. Type *netinfo* in the terminal window.
5. Press **Enter** to see the network status.



6. Type *macinfo* in the terminal window.
7. Press **Enter** to see the MAC status.

```
>macinfo
Interface: LAN865X Driver Statistics

 Receive Statistics
         nRxOkPackets: 0
         nRxPendBuffers: 0
         nRxSchedBuffers: 0
         nRxErrorPackets: 0
         nRxFragmentErrors: 0
         nRxBuffNotAvailable: 0

 Transmit Statistics
         nTxOkPackets: 22
         nTxPendBuffers: 0
         nTxErrorPackets: 0
         nTxQueueFull: 0

Interface: LAN865X Hardware Register Status
         PLCA-Enabled: 0x1
         PLCA-Status: 0x1
         PLCA-NodeId: 0x0
         PLCA-NodeCount: 0x8
         PLCA-BurstCount: 0x8
         PLCA-BurstTimer: 0x80
         TC6-ChunkSize: 0x40
         TX-CutThrough: 0x1
         RX-CutThrough: 0x0
         Promiscuous: 0x0
         SPI Frequency: 0xe4e1c0
```

8. Type *ping <Ip address of other node>* in the terminal window.
   You should get a reply for your request if the network was established.

```
>ping 192.168.100.12
>Ping: reply[1] from 192.168.100.12: time = 2ms
Ping: reply[2] from 192.168.100.12: time = 2ms
Ping: reply[3] from 192.168.100.12: time = 1ms
Ping: reply[4] from 192.168.100.12: time = 1ms
Ping: done. Sent 4 requests, received 4 replies.
```

You have successfully created and tested your first TCP/IP application.

9. Create an IPERF UDP server on one node *iperf -s -u* in the terminal window.

```
iperf -s -u

iperf: Starting session instance 0
------------------------------------------------------------
iperf: Server listening on UDP port 5001
>
iperf: instance 0 session started ...
    - Local  192.168.100.11 port 5001 connected with
    - Remote 192.168.100.12 port 49336
    - [ 0-  1 sec]    0/ 810 ( 0%)     9516 Kbps
    - [ 1-  2 sec]    0/ 811 ( 0%)     9528 Kbps
    - [ 2-  3 sec]    0/ 811 ( 0%)     9528 Kbps
    - [ 3-  4 sec]    0/ 810 ( 0%)     9516 Kbps
    - [ 4-  5 sec]    0/ 811 ( 0%)     9528 Kbps
    - [ 5-  6 sec]    0/ 811 ( 0%)     9528 Kbps
    - [ 6-  7 sec]    0/ 810 ( 0%)     9516 Kbps
    - [ 7-  8 sec]    0/ 811 ( 0%)     9528 Kbps
    - [ 8-  9 sec]    0/ 811 ( 0%)     9528 Kbps
    - [0.0- 10.0 sec]    0/ 8104 ( 0%)     9525 Kbps
iperf: instance 0 completed ...
iperf instance 0: Rx done. Socket closed.
iperf instance 0: Ready for the next session.
```

10. Create an IPERF UDP client on the other node *iperf -c <SERVER_IP_ADDR> -u -b 10M* in the terminal window.

```
iperf -c 192.168.100.11 -u -b 10M

iperf: Starting session instance 0

Given in BW: 10000000+0=10000000iperf: Using the default interface!
>    - RemoteNode MAC: 0 4 25 1c a0 4
------------------------------------------------------------
iperf: Client connecting to 192.168.100.11, UDP port 5001

iperf: instance 0 started ...
    - Local  192.168.100.12 port 49336 connected with
    - Remote 192.168.100.11 port 5001
    - Target rate = 10000000 bps, period = 1 ms
    - [ 0-  1 sec]    0/ 813 ( 0%)     9561 Kbps
    - [ 1-  2 sec]    0/ 810 ( 0%)     9526 Kbps
    - [ 2-  3 sec]    0/ 810 ( 0%)     9526 Kbps
    - [ 3-  4 sec]    0/ 810 ( 0%)     9526 Kbps
    - [ 4-  5 sec]    0/ 811 ( 0%)     9528 Kbps
    - [ 5-  6 sec]    0/ 810 ( 0%)     9526 Kbps
    - [ 6-  7 sec]    0/ 810 ( 0%)     9526 Kbps
    - [ 7-  8 sec]    0/ 811 ( 0%)     9528 Kbps
    - [ 8-  9 sec]    0/ 810 ( 0%)     9526 Kbps
    - [ 9- 10 sec]    0/ 810 ( 0%)     9526 Kbps
    - [0.0- 10.0 sec]    0/ 8104 ( 0%)     9530 Kbps
    ------------------------------------------
    - [0.0- 10.1 sec]    0/ 8113 ( 0%)     9445 Kbps
iperf: instance 0 completed ...iperf instance 0: Tx done. Socket closed.
iperf: instance 0 completed.
```

You have successfully achieved a 9.5Mbps throughput on the 10BASE-T1S network.

3.0 EXAMPLE FIRMWARE

The example project (included in the zip file) shows how to configure the LAN865x MAC-PHY and use the API to read from and write to MAC-PHY registers.

The example supports the following operations via terminal:

- MAC-PHY register read using *macinfo*
- Iperf support
- Ping support
- DNS support

3.1 Main File Description

3.1.1 LAN865x MAC-PHY DRIVER

The MAC-PHY driver is located in the example folder at *firmware\src\config\default\driver\lan865x and \src\dynamic*. It consists of two files, which are:

- *drv_lan865x.h*
- *drv_lan865x_api.c*

The files provide

- the functions to configure the MAC-PHY on initial startup,
- address and bit-mapping for each register of the MAC-PHY and
- APIs to access and modify MAC-PHY registers.

The following three methods are used to access and modify a register value of the MAC-PHY.

- This method is used to write a value to the register.

```
TCPIP_MAC_RES DRV_LAN865X_WriteRegister (uint8_t idx, uint32_t addr, uint32_t value,
bool protected, DRV_LAN865X_RegCallback_t txCallback, void *pTag)
```

- This method is used to read a value from the register.

```
TCPIP_MAC_RES DRV_LAN865X_ReadRegister (uint8_t idx, uint32_t addr, bool protected,
DRV_LAN865X_RegCallback_t rxCallback, void *pTag)
```

- This method is used to write a value to a register based on a bit mask.

```
TCPIP_MAC_RES DRV_LAN865X_ReadModifyWriteRegister (uint8_t idx, uint32_t addr,
uint32_t value, uint32_t mask, bool protected,
DRV_LAN865X_RegCallback_t modifyCallback, void *pTag)
```

3.1.2 app.c

The *app.c* file contains an example on how to access a MAC-PHY register and modify it if necessary.

1. To access the register, make sure the TCPIP stack is up and running, by reading the status *TCPIP_STACK_Status(sysObj.tcpip) == SYS_STATUS_READY.*

2. To read from a MAC-PHY register, you can use the method *DRV_LAN865X_ReadRegister*.
   In the example you can see this in method *APP_Task()*, under switch case *APP_GENERAL_INIT*.

```
TCPIP_MAC_RES result = DRV_LAN865X_ReadRegister (0u /* first instance */, 0x00000000 /
* ID-Register */, true /* protected */, OnIdRead, NULL);
```

   Here, it is reading the value of the *ID* register and invoking the callback *OnIdRead* to print the read value.

3. To write to a MAC-PHY register, you can use the method *DRV_LAN865X_WriteRegister*.

```
TCPIP_MAC_RES result = DRV_LAN865X_WriteRegister (0u /* first instance */,
0x00000000 /* ID-Register */, 0x1u /* reg_va l */, true /* protected */,
OnIdWrite, NULL);
```

   Here, it is writing the value “0x1u” to the *ID* register.

APPENDIX A: REVISION HISTORY

Revision	Date	Section/Figure/Entry	Correction
DS00004964A	2023-03-22	Initial version of this document	

Note the following details of the code protection feature on Microchip products:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner, within operating specifications, and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip product is strictly prohibited and may violate the Digital Millennium Copyright Act.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is "unbreakable" Code protection is constantly evolving. Microchip is committed to continuously improving the code protection features of our products.

Trademarks

ISBN: 978-1-6683-2153-9

For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Austin, TX
Tel: 512-257-3370

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453
Tel: 317-536-2380

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608
Tel: 951-273-7800

Raleigh, NC
Tel: 919-844-7510

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110
Tel: 408-436-4270

Canada - Toronto
Tel: 905-695-1980
Fax: 905-695-2078

ASIA/PACIFIC

Australia - Sydney
Tel: 61-2-9868-6733

China - Beijing
Tel: 86-10-8569-7000

China - Chengdu
Tel: 86-28-8665-5511

China - Chongqing
Tel: 86-23-8980-9588

China - Dongguan
Tel: 86-769-8702-9880

China - Guangzhou
Tel: 86-20-8755-8029

China - Hangzhou
Tel: 86-571-8792-8115

China - Hong Kong SAR
Tel: 852-2943-5100

China - Nanjing
Tel: 86-25-8473-2460

China - Qingdao
Tel: 86-532-8502-7355

China - Shanghai
Tel: 86-21-3326-8000

China - Shenyang
Tel: 86-24-2334-2829

China - Shenzhen
Tel: 86-755-8864-2200

China - Suzhou
Tel: 86-186-6233-1526

China - Wuhan
Tel: 86-27-5980-5300

China - Xian
Tel: 86-29-8833-7252

China - Xiamen
Tel: 86-592-2388138

China - Zhuhai
Tel: 86-756-3210040

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444

India - New Delhi
Tel: 91-11-4160-8631

India - Pune
Tel: 91-20-4121-0141

Japan - Osaka
Tel: 81-6-6152-7160

Japan - Tokyo
Tel: 81-3-6880- 3770

Korea - Daegu
Tel: 82-53-744-4301

Korea - Seoul
Tel: 82-2-554-7200

Malaysia - Kuala Lumpur
Tel: 60-3-7651-7906

Malaysia - Penang
Tel: 60-4-227-8870

Philippines - Manila
Tel: 63-2-634-9065

Singapore
Tel: 65-6334-8870

Taiwan - Hsin Chu
Tel: 886-3-577-8366

Taiwan - Kaohsiung
Tel: 886-7-213-7830

Taiwan - Taipei
Tel: 886-2-2508-8600

Thailand - Bangkok
Tel: 66-2-694-1351

Vietnam - Ho Chi Minh
Tel: 84-28-5448-2100

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4485-5910
Fax: 45-4485-2829

Finland - Espoo
Tel: 358-9-4520-820

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Garching
Tel: 49-8931-9700

Germany - Haan
Tel: 49-2129-3766400

Germany - Heilbronn
Tel: 49-7131-72400

Germany - Karlsruhe
Tel: 49-721-625370

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Germany - Rosenheim
Tel: 49-8031-354-560

Israel - Ra'anana
Tel: 972-9-744-7705

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Italy - Padova
Tel: 39-049-7625286

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Norway - Trondheim
Tel: 47-7288-4388

Poland - Warsaw
Tel: 48-22-3325737

Romania - Bucharest
Tel: 40-21-407-87-50

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

Sweden - Gothenberg
Tel: 46-31-704-60-40

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820