



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Facultad de Ingeniería y Arquitectura
GICEP – GREDyP – GTT

Guía uso lwIP para enviar y recibir datos a través de TCP/IP

En esta guía se presentan los pasos necesarios para la transmisión de datos, a través de TCP/IP, para la tarjeta de desarrollo Hercules RM57Lx LaunchPad. Posteriormente se describe el proceso para realizar la interconexión con Matlab/Simulink.

Autoría

Ana Isabel Narváez Villota – Estudiante de maestría en Ingeniería Eléctrica
Director: Armando Jaime Ustariz Farfán
Codirector: Luis Fernando Díaz Cadavid

12/Enero/2020

Requerimientos de software

HALCoGen: mínimo versión 04.07.01

Code Composer Studio: mínimo versión 9.1.0

Matlab/Simulink®: versión 2018a

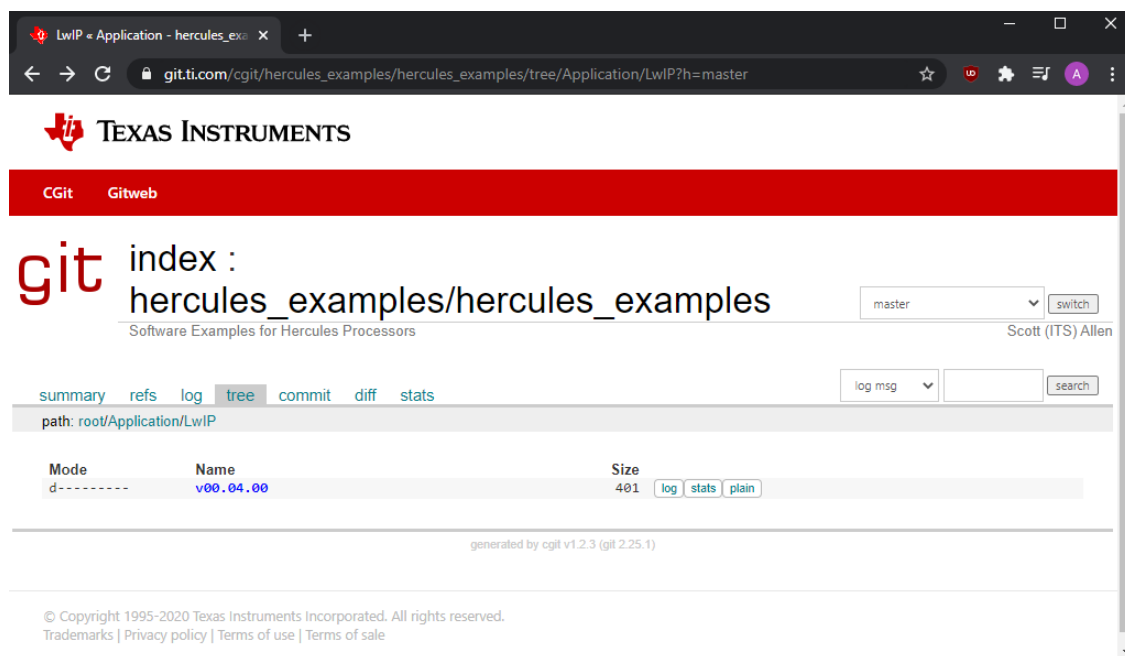
Descargar librerías

Se necesita la librería lwIP Demo Software. Se tiene la versión 3.0, pero el acceso a los códigos fuente (source) es oculto, así que se trabaja con la última versión: lwIP Demo Software versión 00.04.00.

Link de descarga:

http://git.ti.com/hercules_examples/hercules_examples/trees/master/Application/LwIP

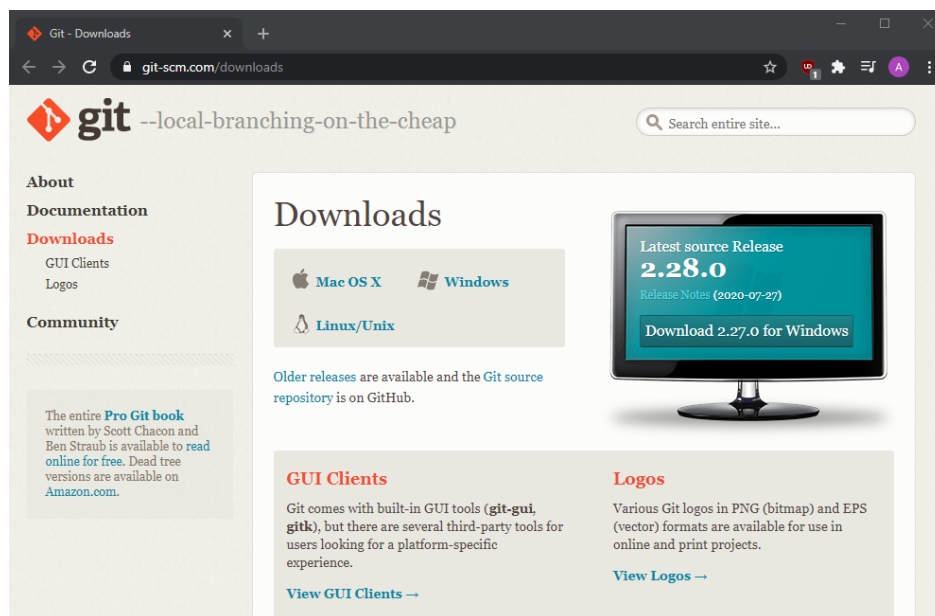
Se procede a acceder y visualizar la siguiente página:



Dado que el ejemplo está en Git, se debe clonar el repositorio. Para ello se tienen los pasos enunciados a continuación.

Paso 1: descargar git

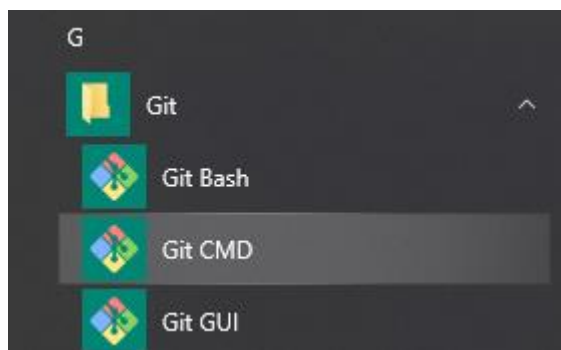
Se ingresa a: <https://git-scm.com/downloads>



Se descarga la versión más actualizada según el sistema operativo del computador bajo uso y se procede a instalar. También, se tiene la documentación disponible en la parte izquierda de la página.

Paso 2: ejecutar la consola

Para este caso se elige y ejecuta la consola de CMD:



Paso 3: configurar el usuario y el email

Se ejecutan los siguientes comandos en la consola de CMD:

```
Selecionar Git CMD

C:\>git config --global user.name "AINV"
C:\>git config --global user.email "ainarvaezv@una1.edu.co"
C:\>
```

Paso 4: creación de un proyecto

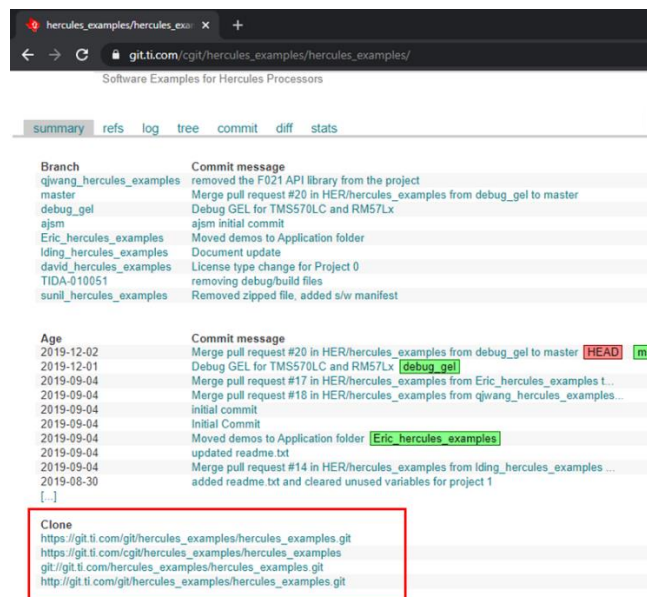
Se crea un proyecto en el disco de preferencia, en este caso se ha asignado el nombre de `texas_instruments_repositorio` y luego se entra a la carpeta creada:

```
Git CMD

C:\Users\LAURA MILENA>cd..
C:\Users>cd..
C:\>git init texas_instruments_repositorio
Initialized empty Git repository in C:/texas_instruments_repositorio/.git/
C:\>cd texas_instruments_repositorio
C:\texas_instruments_repositorio>
```

Paso 5: clonar el repositorio

Para ello, primero se obtiene el enlace para clonar el repositorio desde git:



Luego, en consola se ejecuta el comando: `git clone git: ...`

```
Git CMD
C:\Users\LAURA_MILENA>cd..
C:\Users>cd..
C:\>git init texas_instruments_repositorio
Initialized empty Git repository in C:/texas_instruments_repositorio/.git/
C:\>cd texas_instruments_repositorio
C:\texas_instruments_repositorio>git clone git://git.ti.com/hercules_examples/hercules_examples.git
Cloning into 'hercules_examples'...
remote: Counting objects: 4301, done.
remote: Compressing objects: 100% (1255/1255), done.
remote: Total 4301 (delta 2889), reused 4301 (delta 2889)
Receiving objects: 100% (4301/4301), 61.16 MiB | 604.00 KiB/s, done.
Resolving deltas: 100% (2889/2889), done.
Updating files: 100% (4919/4919), done.
C:\texas_instruments_repositorio>
```

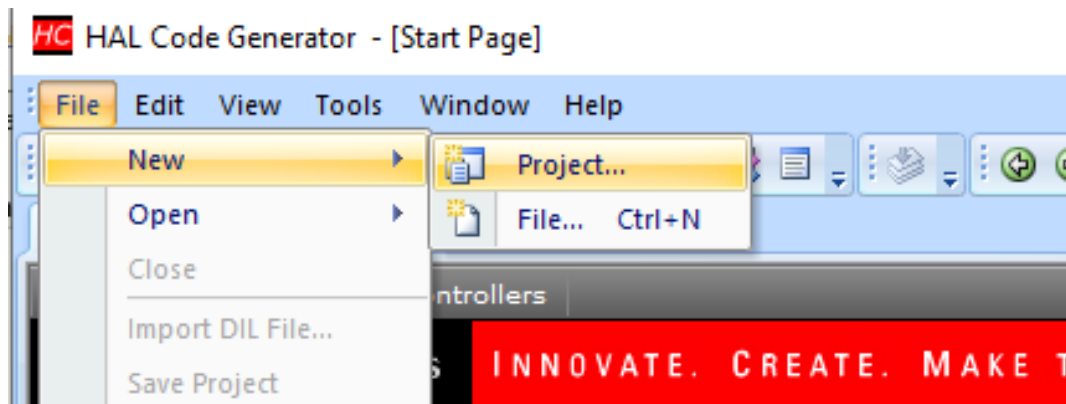
Programación en HALCoGen – HCG

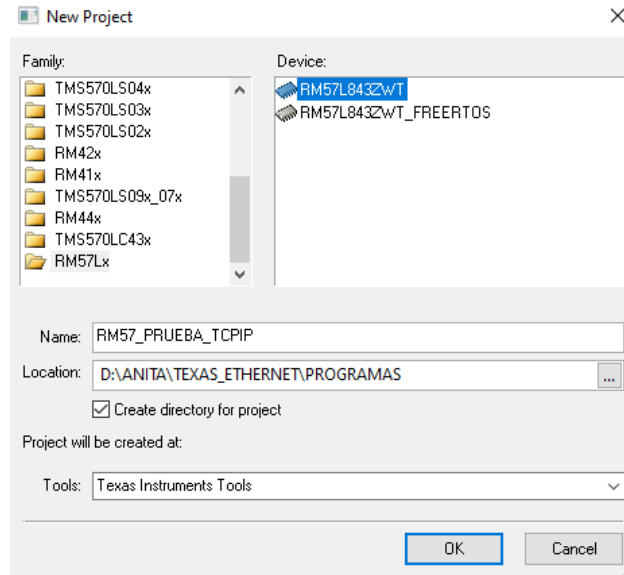
En esta sección se realiza la configuración de los drivers EMAC y MDIO en la RM57x Launchpad.

El siguiente proceso se basa en las referencias [1] y [2].

Paso 1: crear un nuevo proyecto en HCG

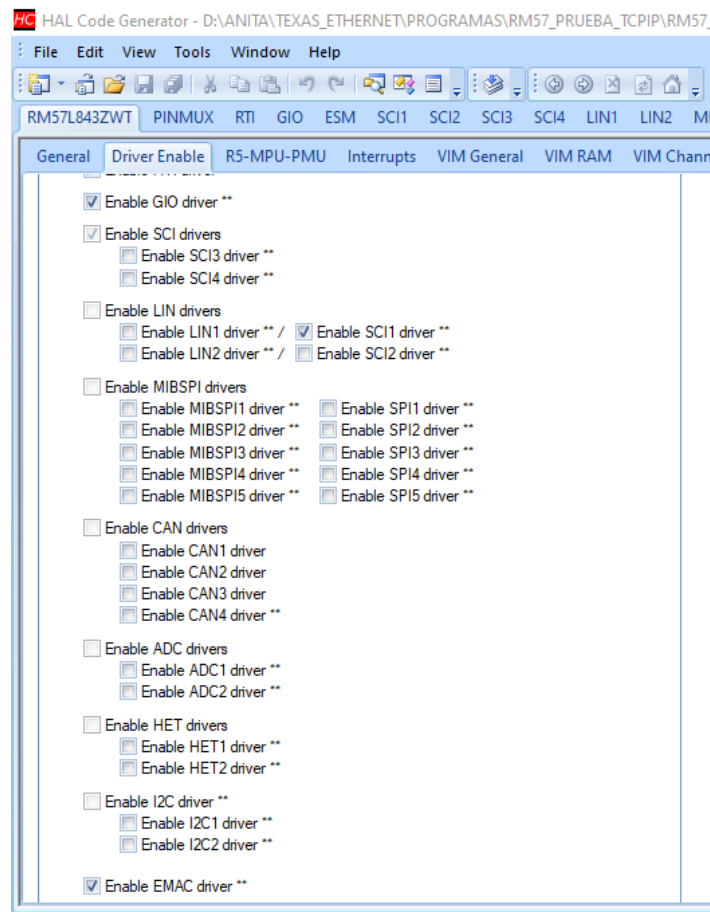
Se abre HALCoGen y se crea un nuevo proyecto teniendo en cuenta las siguientes pautas de configuración.





Paso 2: habilitar drivers

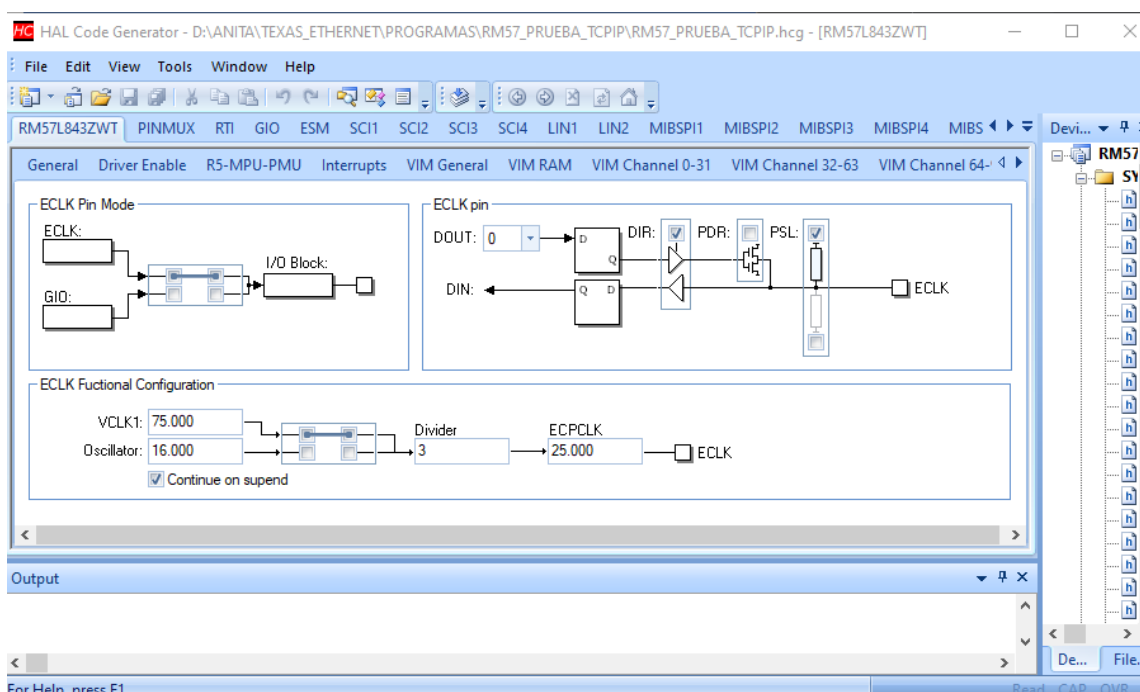
Se habilitan los drivers RM57l843ZWT>Driver Enable Tab: GIO, SCI1 y EMAC.



Paso 3: configuración del reloj

Se configura el reloj para proporcionar una frecuencia de 25 MHz para el PHY. Para ello se sigue la ruta RM57L843WT>ECLK Tab:

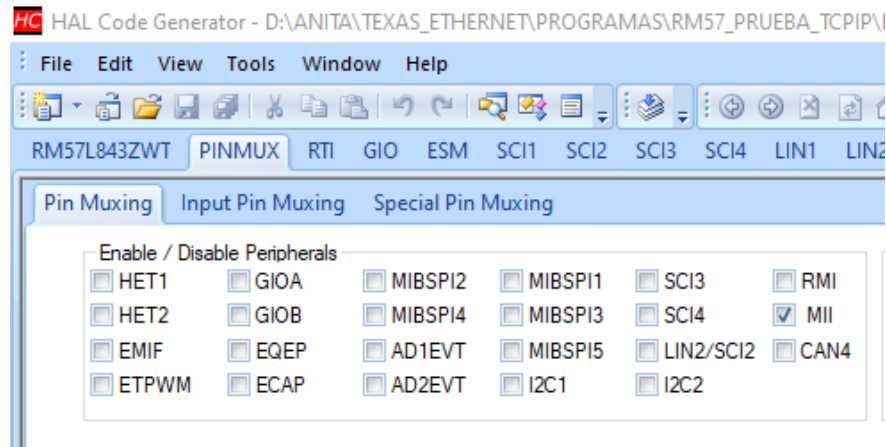
- En el ECLK Pin Mode Group se pone el ECLK pin Mode en ECLK.
- En el ECLK pin Group, la casilla DIR debe estar seleccionada.
- En el ECLK Functional Configuration Group, se cambia el Divider a 3, así que ECPClk es 25 MHz.
- Se selecciona la casilla “Continue on suspend”.



Paso 4: configuración de PINMUX

Se sigue la ruta PINMUX>PIN Muxing Tab. Se cambian los pines de interfaz MII y MDIO a sus ubicaciones predeterminadas.

- Se requiere que MII esté marcado en el Enable/Disable Peripherals Group. Es necesario poner la MAC en modo MII, aunque esto moverá los pines MII a su ubicación alternativa, que debe deshacerse manualmente.

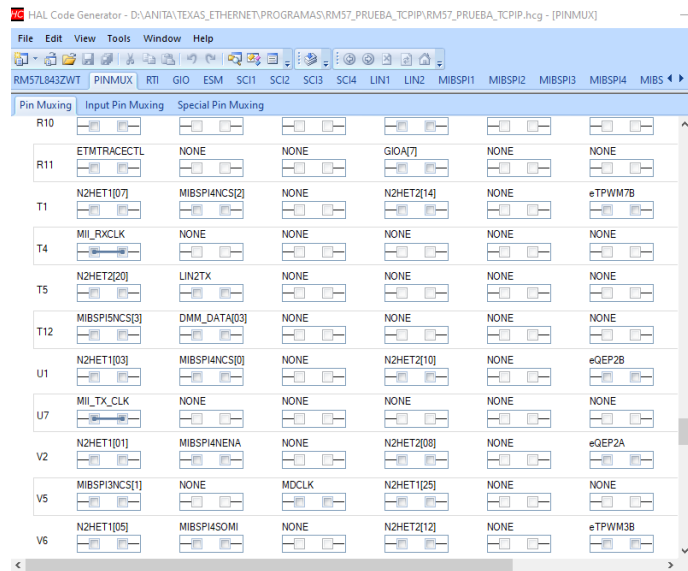


- Desmarcar las señales MII y MDIO en los Balls A14, B4, B11, D19, E18, F3, G3, G19, H18, H19, J18, J19, K19, N19, P1, R2 y V5. Ahora estas filas deben estar en blanco, aunque se puede seleccionar funciones que no sean de Ethernet, si así se desea.

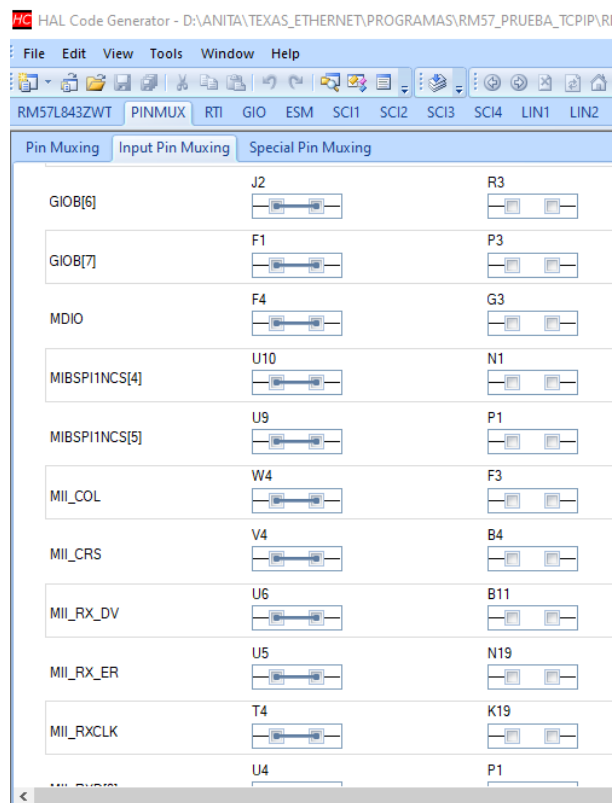
Ball	Default Mux	Mux Option 1	Mux Option 2	Mux Option 3
A4	N2HET1[16]	NONE	NONE	ETPWM1SYNCl
A13	N2HET1[17]	EMIF_nOE	SCI4RX	NONE
A14	N2HET1[26]	NONE	MII_RXD[1]	RMII_RXD[1]
B2	MIBSPI3NCS[2]	I2C1_SDA	NONE	N2HET1[27]
B3	N2HET1[22]	EMIF_nDQM[3]	NONE	NONE
B4	N2HET1[12]	MIBSPI4NCS[5]	MII_CRS	RMII_CRS_DV
D5	GIOA[5]	NONE	NONE	EXTCLKIN

- Cambiar la selección en los balls T4, U7 a las funciones por defecto (desde MII_RX_AVCLK4 a MII_RXCLK y desde MII_TX_AVCLK4 a MII_TX_CLK). El PHY proporcionará estos relojes a la MAC en el launchpad.

Nota: List Conflicts muestra un total de 2 conflictos para el Ball T4 y U7, solo ignorar esto.



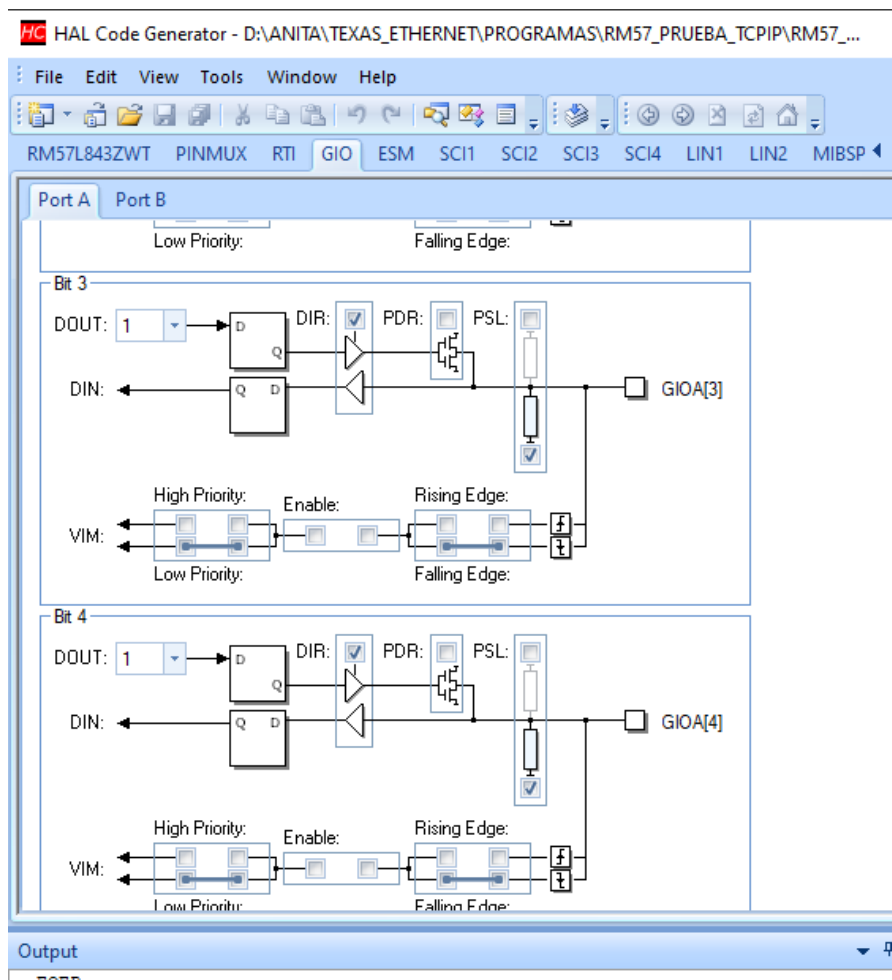
- Se sigue la ruta PINMUX>Input Pin Muxing Tab. Se cambian todas las entradas de las señales MII y MDIO a los estados por defecto (columna de la izquierda). Las señales son: MDIO=F4, MII_COL=W4, MII_CRS=V4, MII_RX_DV=U6, MII_RX_ER=U5, MII_RXCLK=T4, MII_RXD[0]=U4, MII_RXD[1]=T3, MII_RXD[2]=U3, MII_RXD[3]=V3, MII_TX_CLK=U7.



Paso 5: configuración del driver GIO

Se sigue la ruta GIO>Port A Tab. GIOA[3] y GIOA[4] necesitan ser configurados en alto (1), para liberar al PHY del reset y power down. Para ello:

- Marcar la caja “DIR” para el Bit 3 y el Bit 4.
- Cambiar DOUT a “1” para el Bit 3 y el Bit 4.

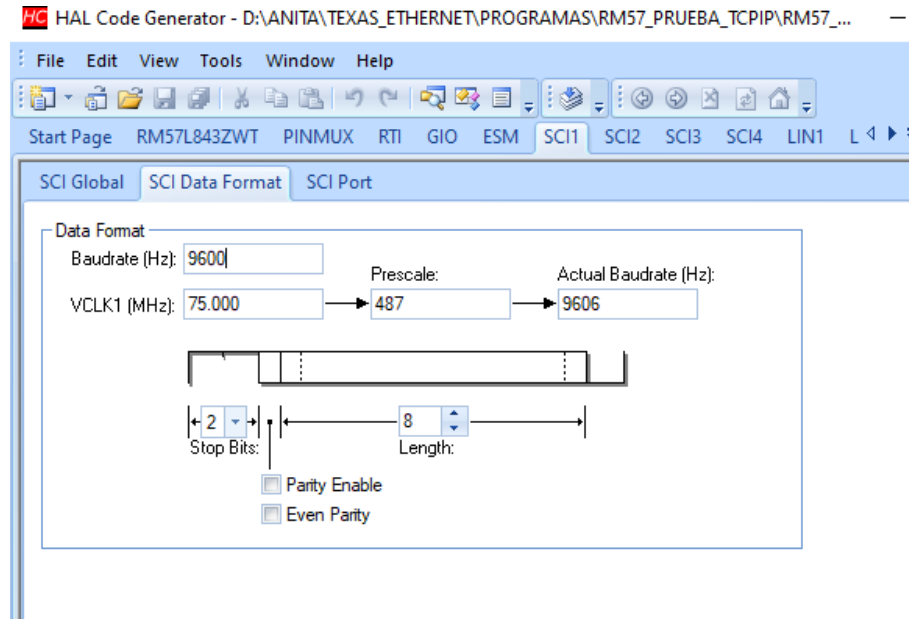


Paso 6: parámetros de configuración del puerto serial

Los mensajes de diagnóstico y debug son impresos en el puerto SCI. Así que la configuración para la consola es la siguiente:

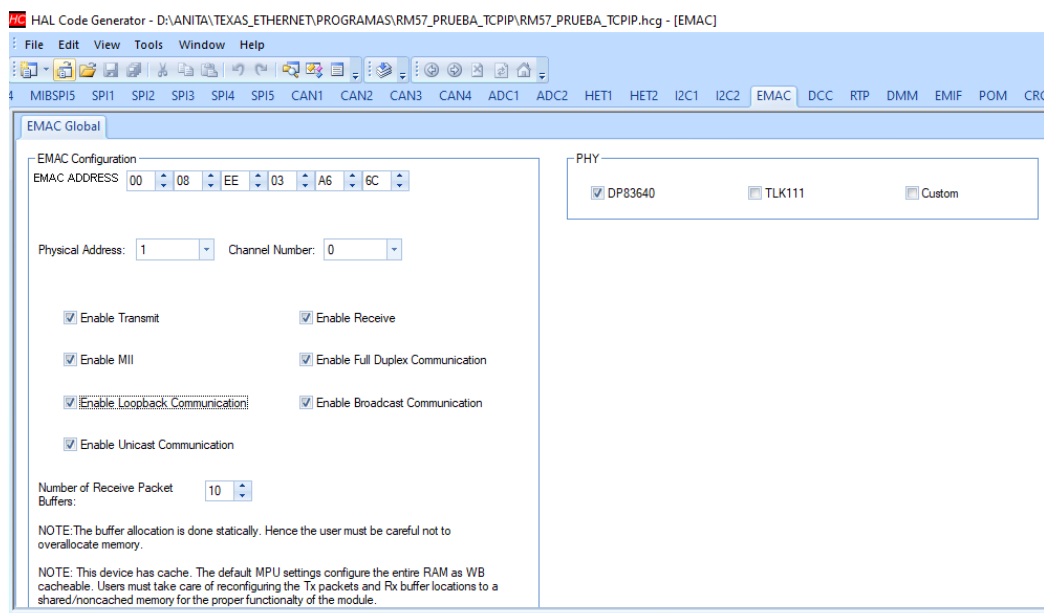
- Baud rate= 9600
- Data= 8 bits

- Parity: None
- Stop bit: 2
- Flow Control: None



Paso 7: configuración del driver EMAC

Se ingresa la dirección MAC de la imagen y las otras pautas de configuración. La dirección EMAC y la dirección física estarán disponibles en el archivo main.c source.

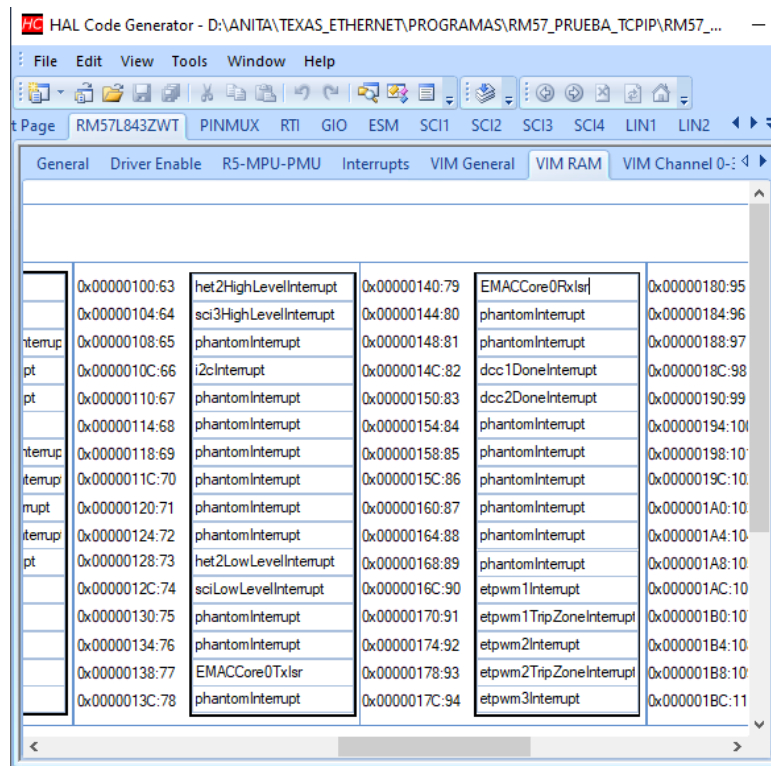


Paso 8: configuración VIM

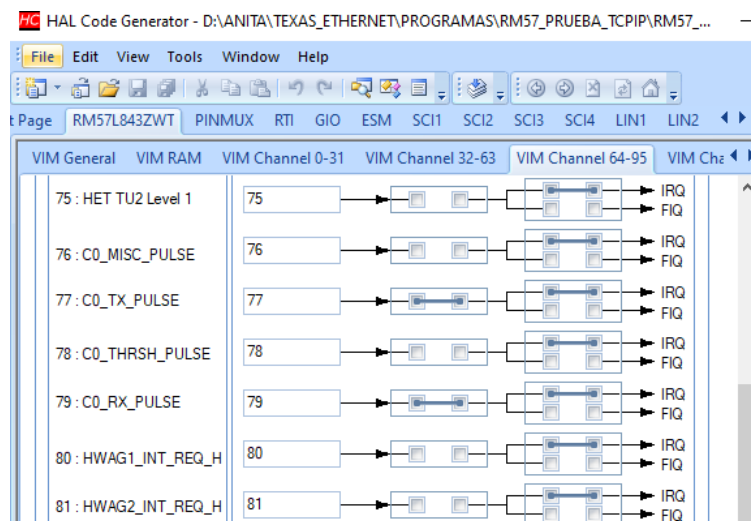
- Se sigue la ruta RM57L843ZWT>VIM RAM Tab y se configura los siguientes controladores de interrupciones:

77: EMACCore0TxIsr

79: EMACCore0RxIsr



- En el VIM Channel 64-95 se habilita el channel 77 y 79.



Paso 9: configuración R5-MPU-PMU

- Acceder a RM57L843ZWT>R5-MPU-PMU Tab y configurar la memoria [2]:

The screenshot shows the configuration tool for the RM57L843ZWT. The 'R5-MPU-PMU' tab is selected. The MPU section is expanded, showing three regions. Region 1 is configured with a base address of 0x00000000, size of 4GB, and permission of PRIV_NA_USER_NA_NOEXEC. Region 2 is configured with a base address of 0x00000000, size of 4MB, and permission of PRIV_RO_USER_RO_EXEC. Region 3 is configured with a base address of 0x00000000, size of 512KB, and permission of PRIV_RW_USER_RW_EXEC. The PMU section is also visible, showing Event Counter 0, 1, and 2 configurations.

Region 4 Configuration

☒ Enable Mpu Region Disable Sub-Region: ☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7

Base Address: Type:

Size: Permission:

End Address:

Region 5 Configuration

☒ Enable Mpu Region Disable Sub-Region: ☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7

Base Address: Type:

Size: Permission:

End Address:

Region 6 Configuration

☒ Enable Mpu Region Disable Sub-Region: ☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7

Base Address: Type:

Size: Permission:

End Address:

Region 7 Configuration

☒ Enable Mpu Region Disable Sub-Region: ☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7

Base Address: Type:

Size: Permission:

End Address:

Region 8 Configuration

☒ Enable Mpu Region Disable Sub-Region: ☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7

Base Address: Type:

Size: Permission:

End Address:

Region 9 Configuration

☒ Enable Mpu Region Disable Sub-Region: ☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7

Base Address: Type:

Size: Permission:

End Address:

Region 10 Configuration

☒ Enable Mpu Region Disable Sub-Region: ☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7

Base Address: Type:

Size: Permission:

End Address:

Region 11 Configuration

☒ Enable Mpu Region Disable Sub-Region: ☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7

Base Address: Type:

Size: Permission:

End Address:

Region 12 Configuration

☒ Enable Mpu Region Disable Sub-Region: ☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7

Base Address: Type:

Size: Permission:

End Address:

Region 13 Configuration

☒ Enable Mpu Region Disable Sub-Region: ☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7

Base Address: Type:

Size: Permission:

End Address:

Region 14 Configuration

☒ Enable Mpu Region Disable Sub-Region: ☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7

Base Address: Type:

Size: Permission:

End Address:

The image shows two configuration windows for memory regions. The top window is titled 'Region 15 Configuration' and the bottom one is 'Region 16 Configuration'. Both windows have a 'Disable Sub-Region' row with checkboxes for sub-regions 0 through 7, all of which are unchecked. In the 'Region 15 Configuration' window, the 'Enable Mpu Region' checkbox is checked, the 'Base Address' is 0xFF000000, the 'Size' is 16_MB, the 'End Address' is 0xFFFFFFFF, the 'Type' is DEVICE_NONSHAREABLE, and the 'Permission' is PRIV_RW_USER_RW_NOEXEC. In the 'Region 16 Configuration' window, the 'Enable Mpu Region' checkbox is checked, the 'Base Address' is 0xFF80000, the 'Size' is 512_KB, the 'End Address' is 0xFFFFFFFF, the 'Type' is DEVICE_NONSHAREABLE, and the 'Permission' is PRIV_RW_USER_RW_NOEXEC.

Region	Enable Mpu Region	Disable Sub-Region (0-7)	Base Address	Size	End Address	Type	Permission
Region 15	<input checked="" type="checkbox"/>	<input type="checkbox"/> 0 <input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5 <input type="checkbox"/> 6 <input type="checkbox"/> 7	0xFF000000	16_MB	0xFFFFFFFF	DEVICE_NONSHAREABLE	PRIV_RW_USER_RW_NOEXEC
Region 16	<input checked="" type="checkbox"/>	<input type="checkbox"/> 0 <input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5 <input type="checkbox"/> 6 <input type="checkbox"/> 7	0xFF80000	512_KB	0xFFFFFFFF	DEVICE_NONSHAREABLE	PRIV_RW_USER_RW_NOEXEC

Paso 10: generar el código

Se guarda el proyecto (File>Save Project) y se genera el código (File>Generate Code).

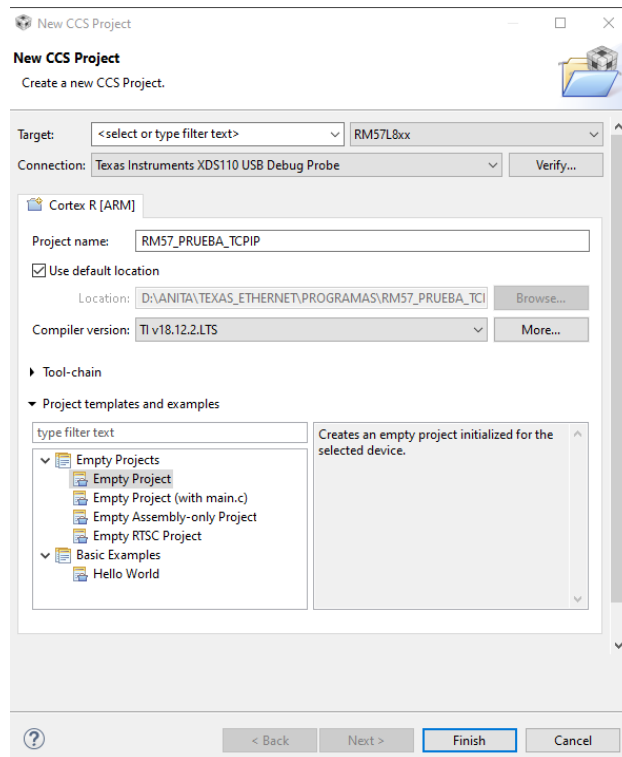
Programación en Code Composer Studio – CCS

Paso 1: ejecutar CCS

Se configura el mismo directorio (carpeta), en la que se guardó el proyecto de HCG.

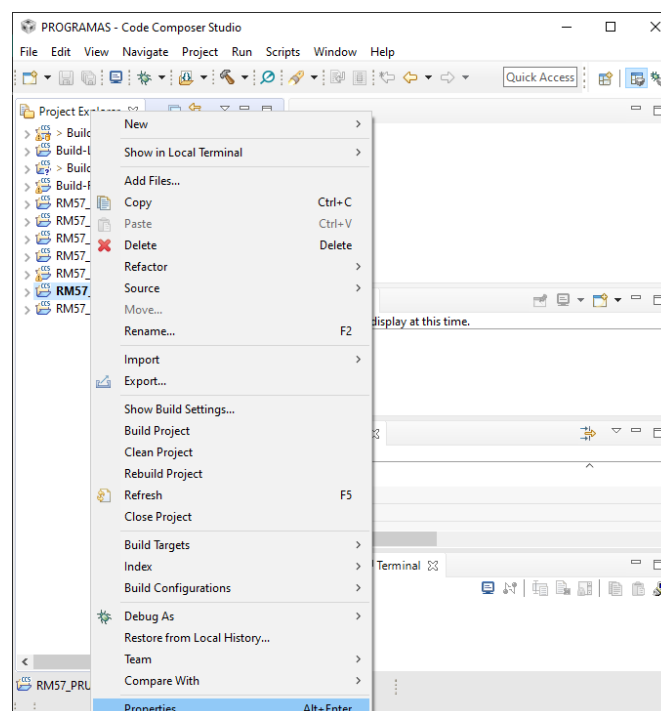
Paso 2: crear un nuevo proyecto

Se crea el proyecto: File>New>CCS Project. Se configura como se muestra en la imagen.

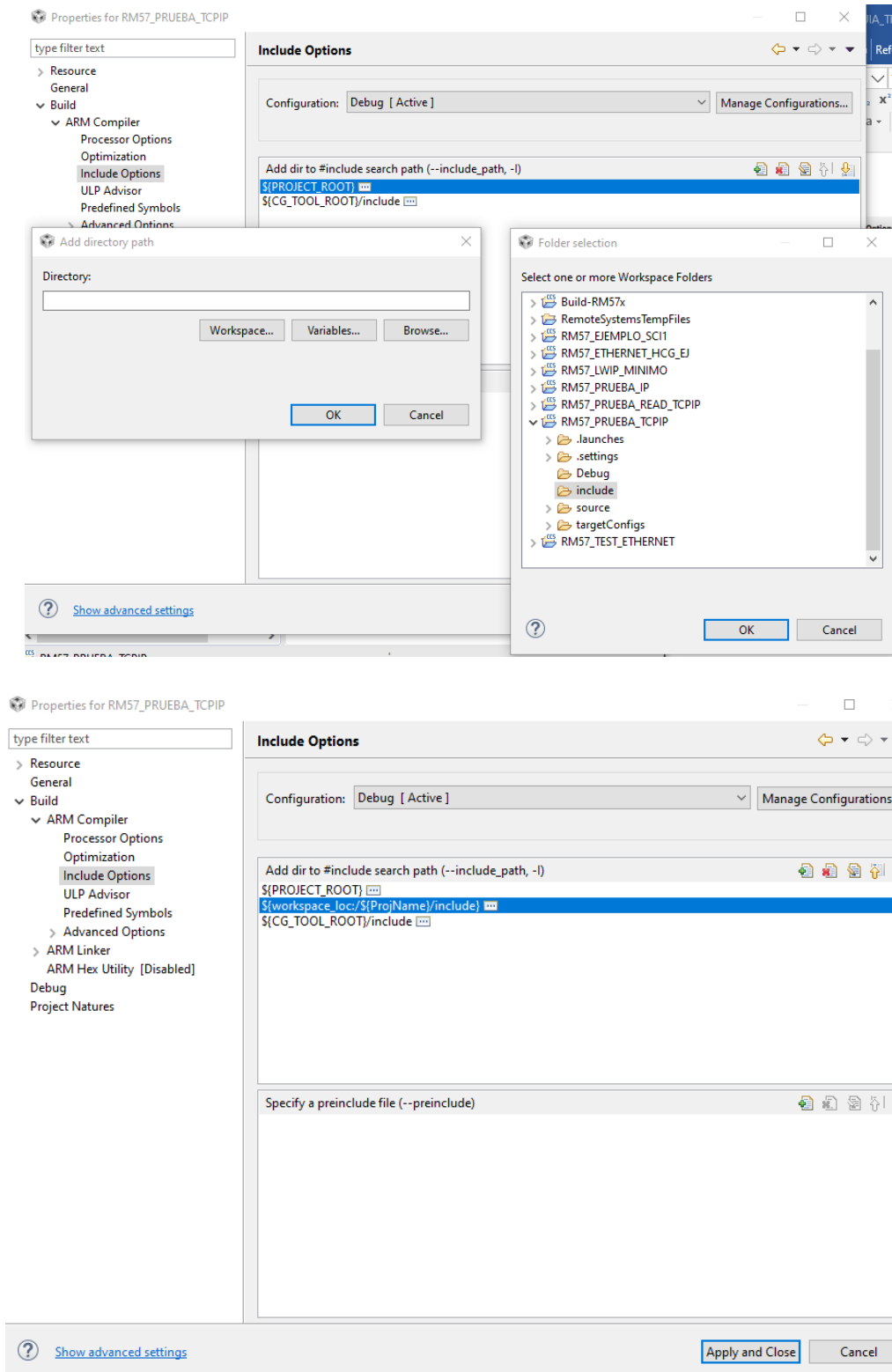


Paso 3: enlazar el proyecto de CCS con el de HCG

Ir a properties:



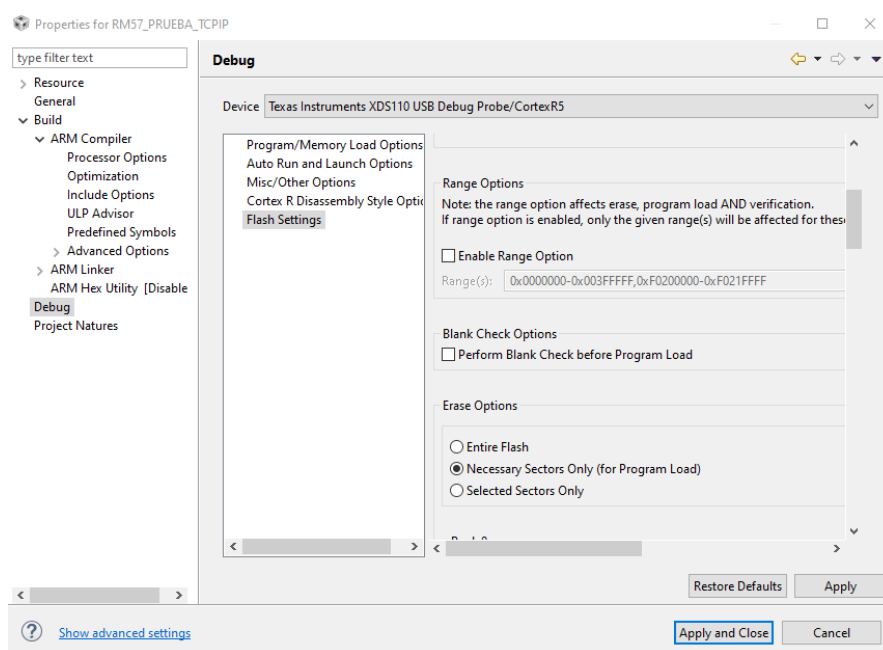
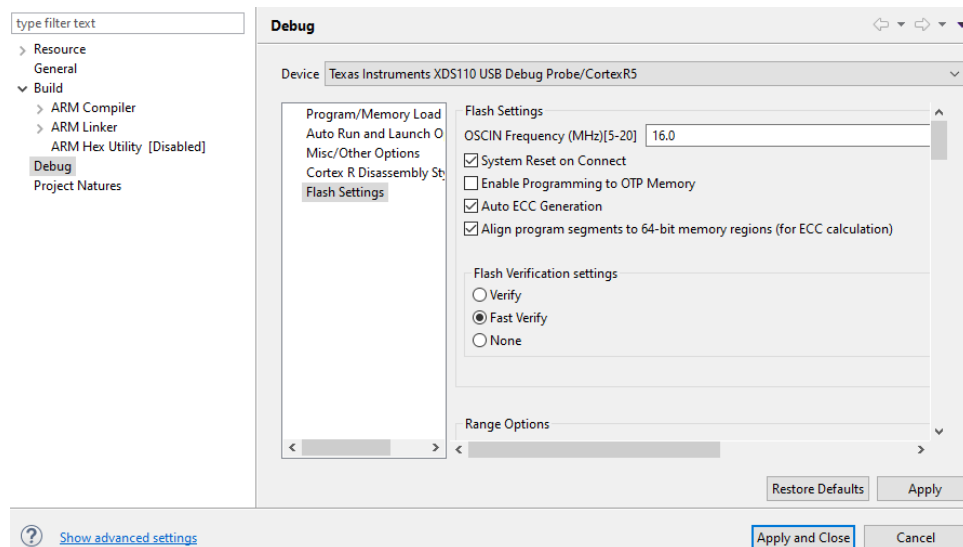
Ahora se realiza el enlace con HCG:



Finalmente dar en Apply and Close.

Paso 4: configurar más propiedades del proyecto

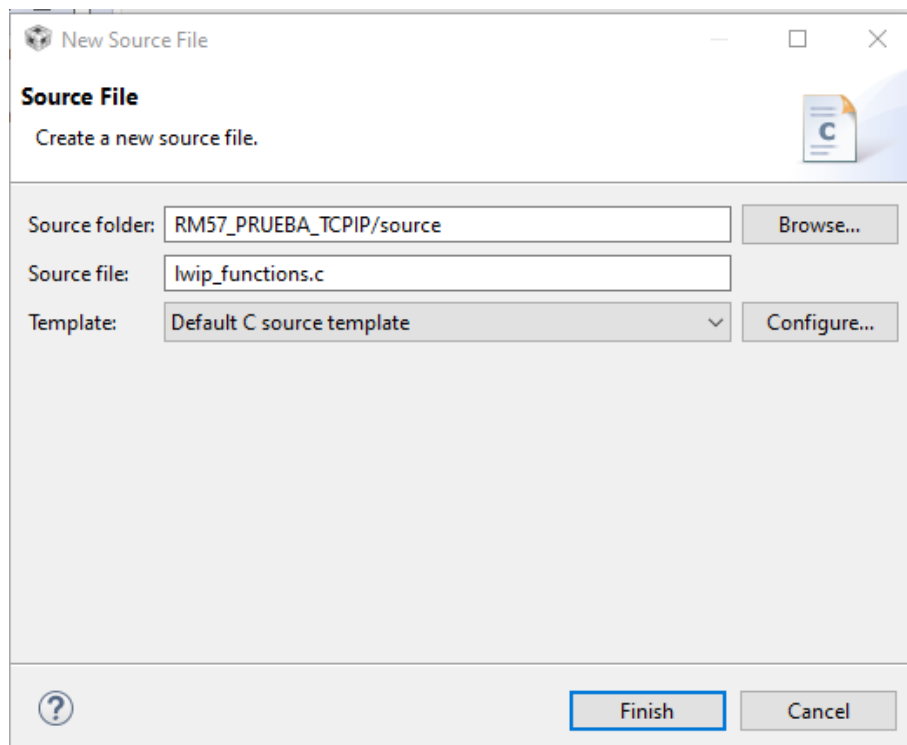
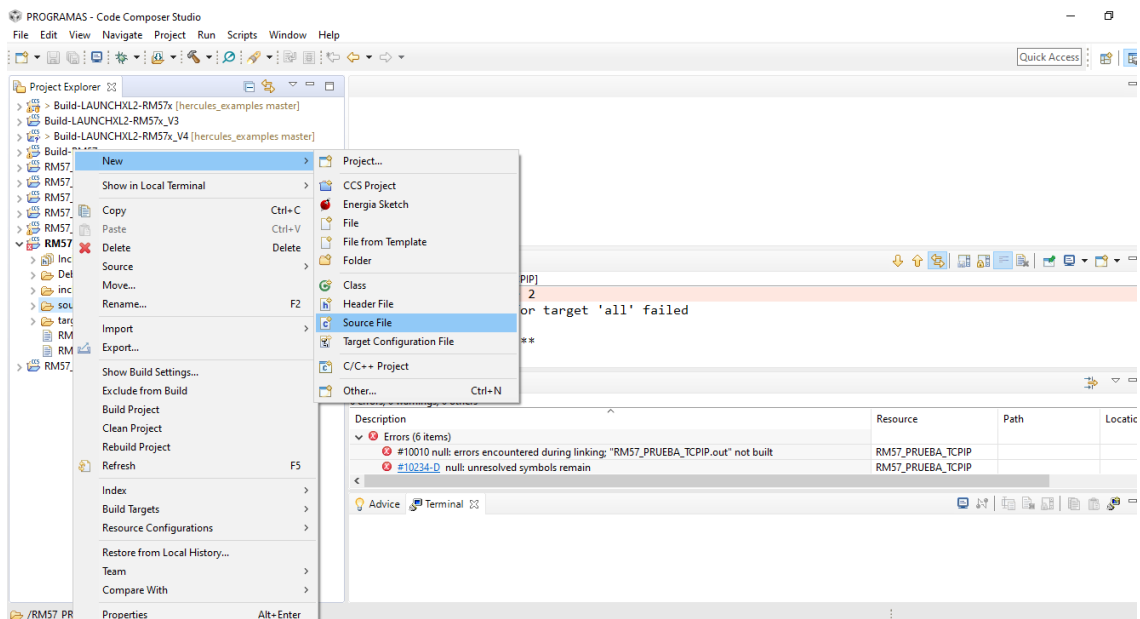
En Properties:



Finalmente dar en Apply and Close.

Paso 5: crear un source file

Se crea un nuevo source file llamado lwip_functions.c. Para ello, dar clic derecho sobre la carpeta source>new>source file.



Copiar y pegar el código fuente disponible en el archivo `lwip_functions.c`, que fue agregado en la carpeta junto a esta guía. Este código fue generado con base en los códigos fuentes disponibles en el repositorio de Texas Instruments [3] y al proyecto Savannah [4], [5].

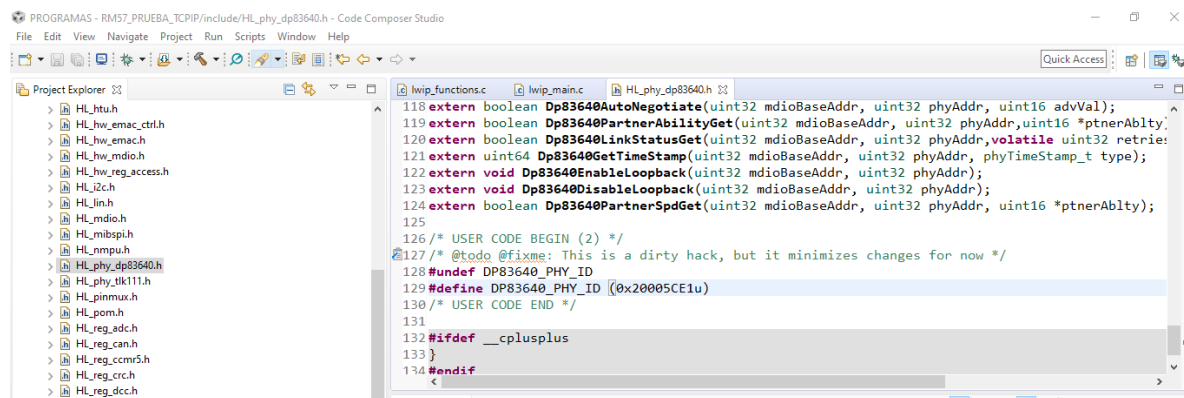
Paso 6: editar HL_phy_dp83640.h

Se requiere cambiar el contenido de este archivo para que este trabaje con el módulo on-board que es el DP83630 Precision PHYTER.

Ir a la ruta: RM57_PRUEBA_TCPIP>include> HL_phy_dp83640.h

Aquí se procede a cambiar el último bloque llamado “USER CODE” como sigue:

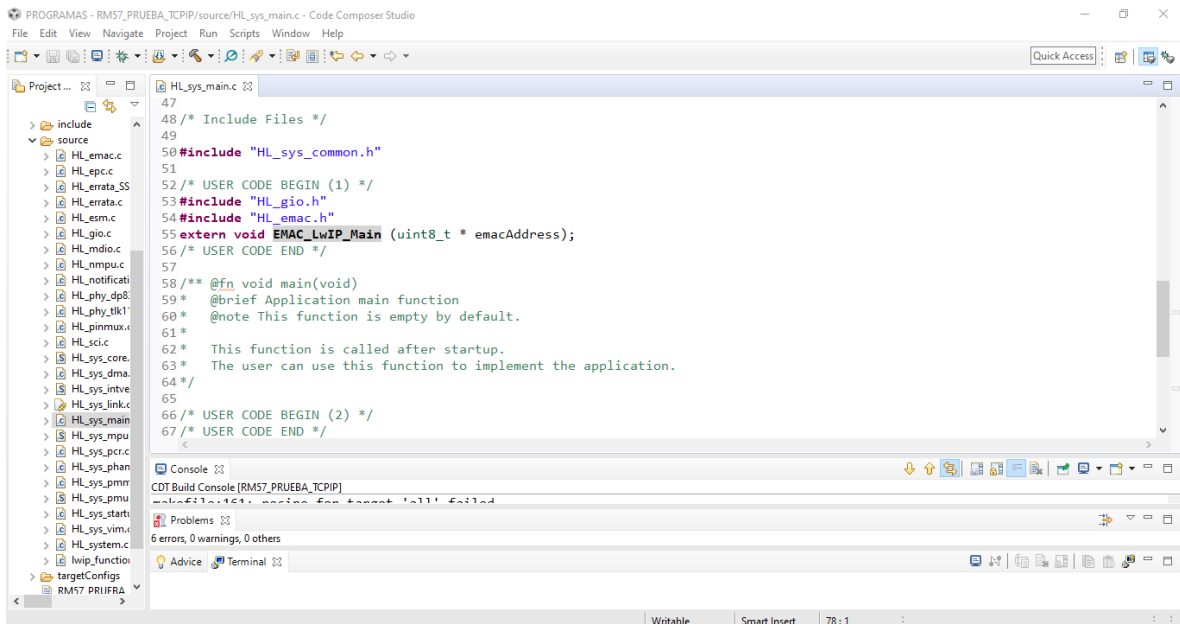
```
/* USER CODE BEGIN (2) */  
/* @todo @fixme: This is a dirty hack, but it minimizes changes for now */  
#undef DP83640_PHY_ID  
#define DP83640_PHY_ID (0x20005CE1u)  
/* USER CODE END */
```



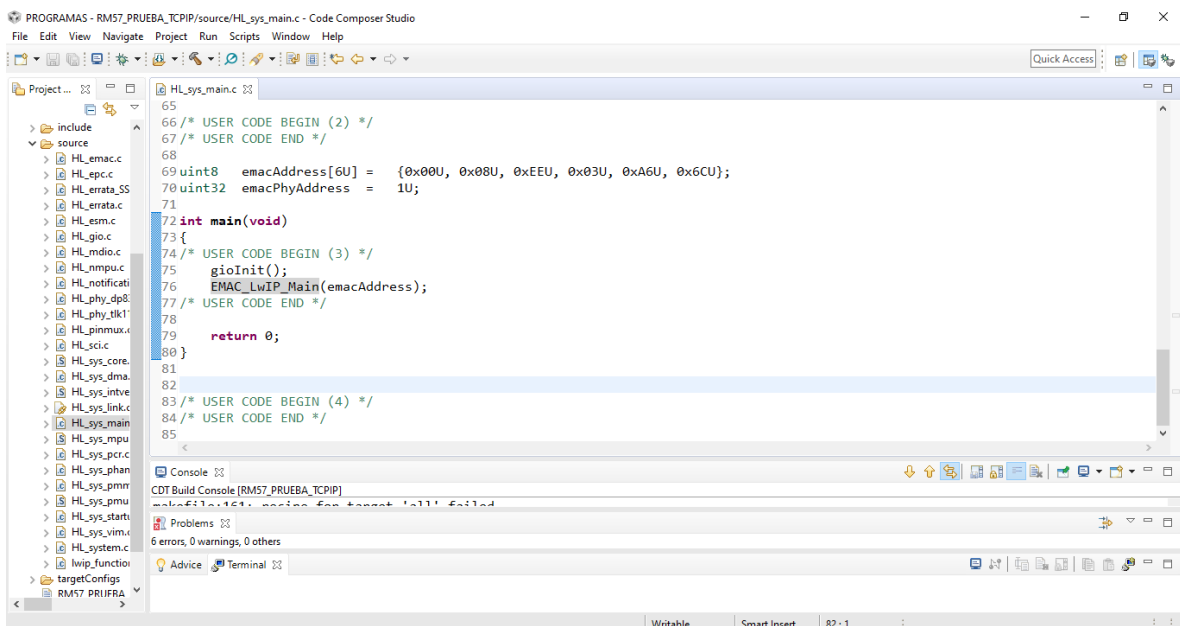
Paso 7: configurar el HL_sys_main.c

Se agrega la lógica al archivo principal HL_sys_main.c en la ruta RM57_PRUEBA_TCPIP>source> HL_sys_main.c

Se agregan los encabezados en el bloque 1:



Se inicializan las funciones en el bloque 3:

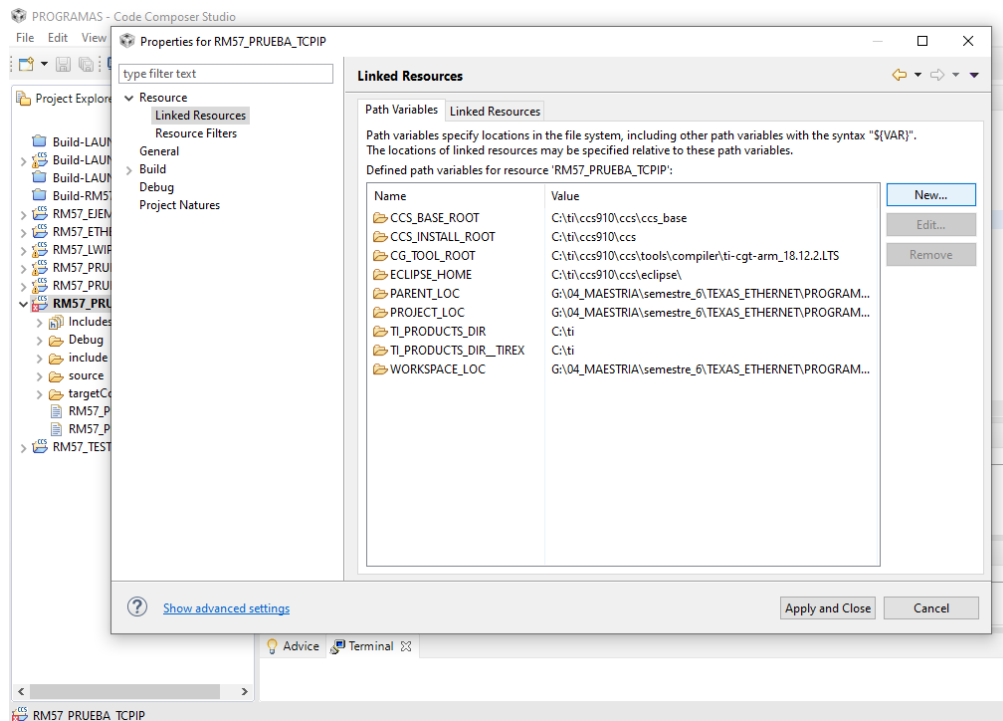
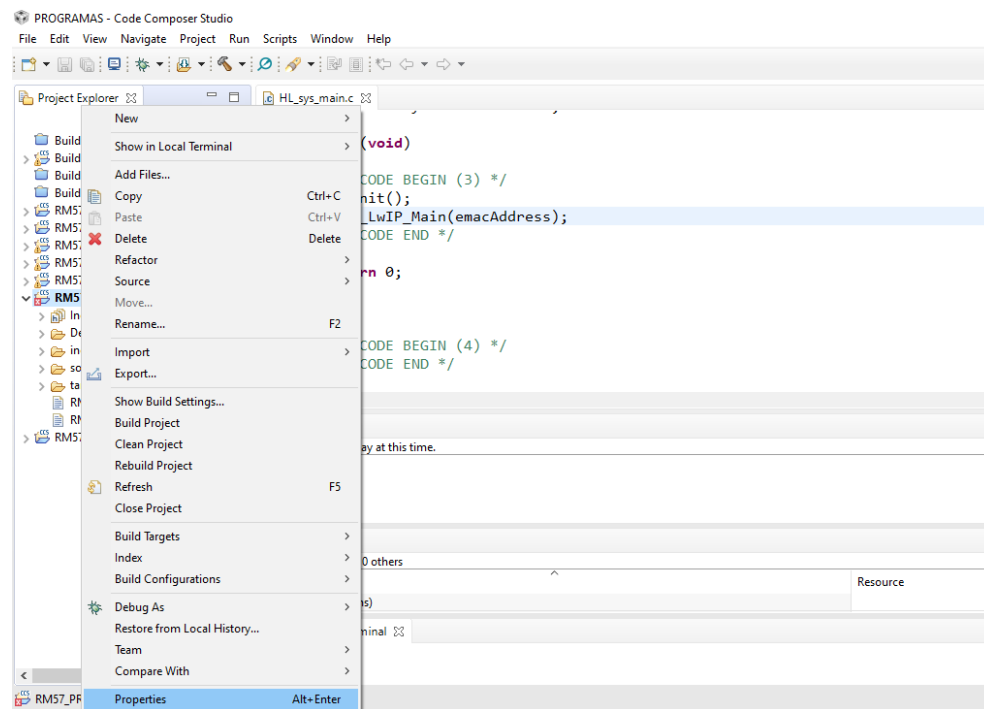


Paso 8: incluir las librerías de lwIP en el proyecto

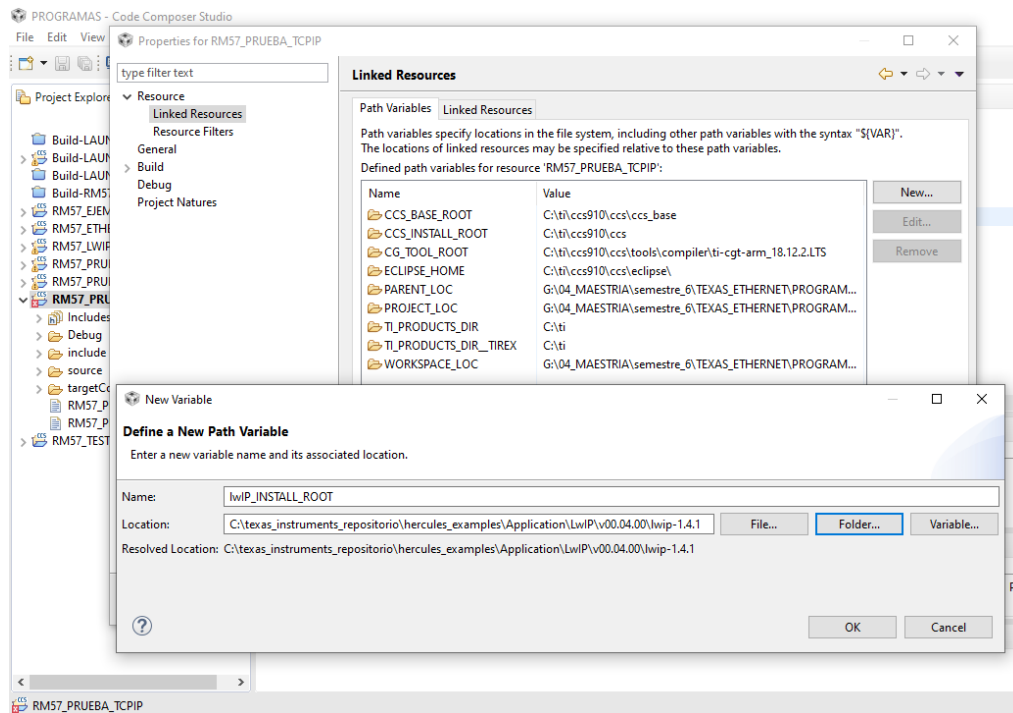
Se incluyen las librerías de lwIP, tanto .c como .h, además se excluyen algunos archivos para el build del proyecto.

Primero, se requiere configurar la ubicación de la instalación de la librería como una variable. Esta puede ser usada para agregar includes, etc.

En la página de propiedades del proyecto, se agrega la ubicación de la instalación de lwIP como una variable de ruta llamada lwIP_INSTALL_ROOT.



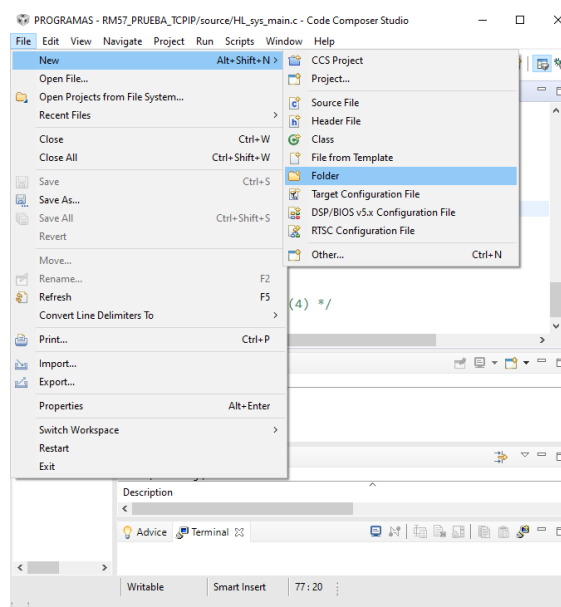
En nuevo, se agrega la carpeta de lwIP del repositorio que se había descargado en Paso 5: clonar el repositorio.



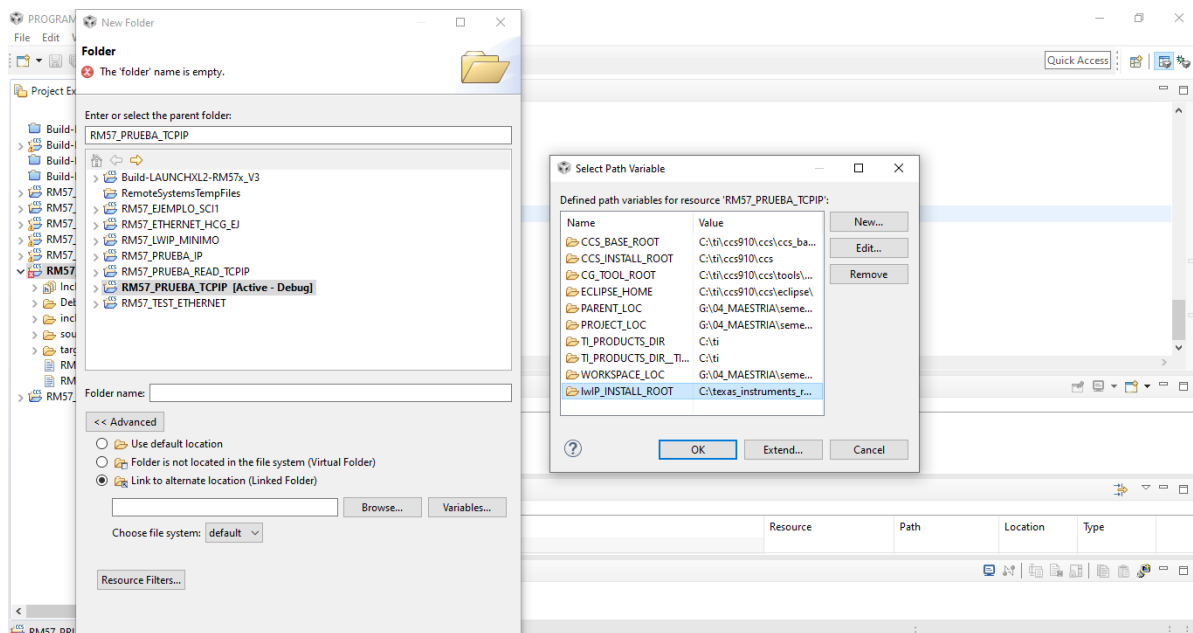
La ruta de la carpeta, para este caso, está en:

C:\texas_instruments_repositorio\hercules_examples\Application\LwIP\v00.04.00\lwip-1.4.1

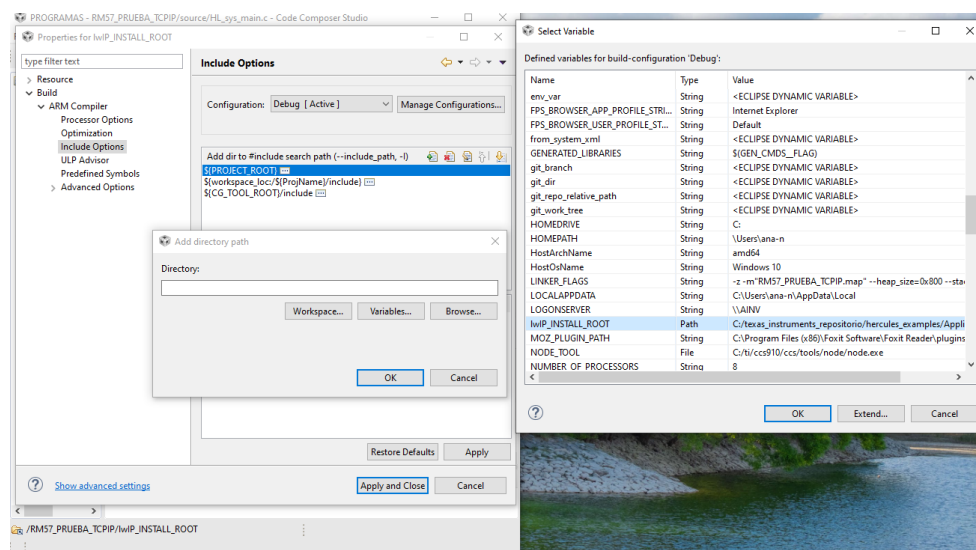
Luego se da en OK, apply and close. Posterior a esto, se agrega la carpeta de lwIP como una carpeta fuente. Para ello, se crea una nueva carpeta con clic derecho sobre el nombre del proyecto:

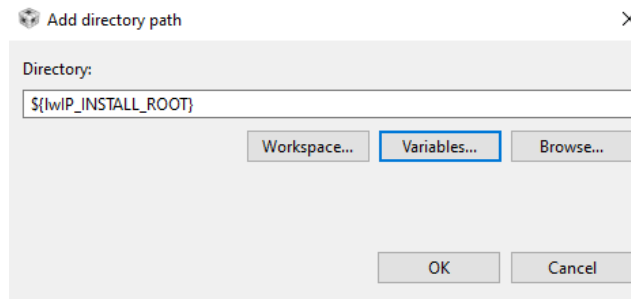


Seleccionar “Advanced”> Link to alternate location (Linked Folder)> Variables> lwIP_INSTALL_ROOT>Finish:

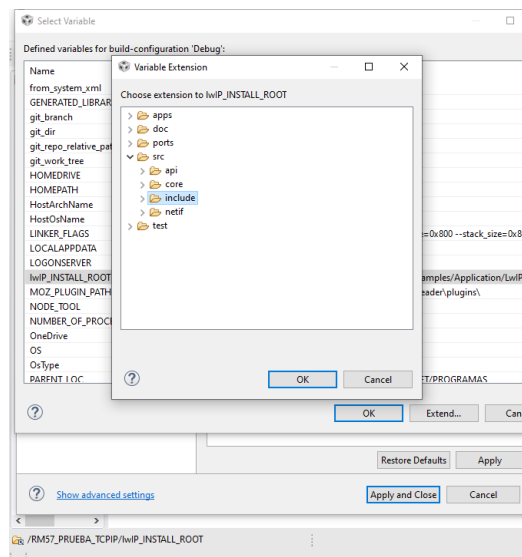


Agregar las ubicaciones necesarias de “include” en properties, usando la variable creada para lwIP:



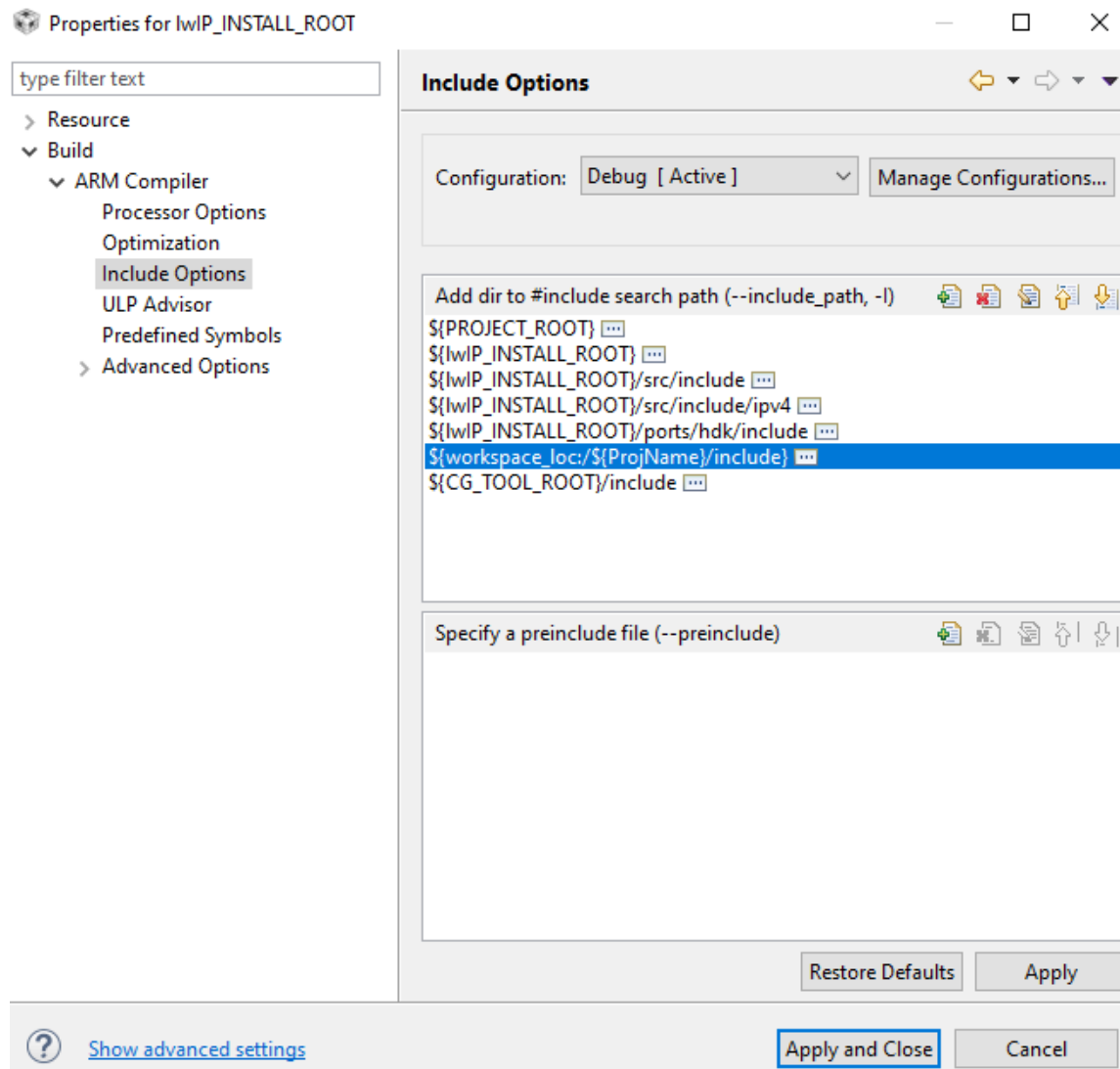


Para agregar la siguiente, nuevamente en Add>Variables>lwIP>Extend>src>include>OK.



Se realiza el mismo procedimiento para src>include>ipv4 y ports>hdk>include.

Con esto, se obtiene lo siguiente:



Finalmente dar clic en Apply, luego en Apply and Close.

Ahora, se procede a excluir los siguientes archivos y carpetas de la compilación. Para ello, hacer clic derecho sobre el archivo o carpeta y seleccionar Exclude from build. Se pueden seleccionar múltiples entradas presionando la tecla CTRL.

```
ports/hdk/netif/hdkif.c
ports/hdk/locator.c
ports/hdk/perf.c
ports/hdk/sys_arch.c
src/core/ipv4
src/core/snmp
src/core/dhcp.c
src/core/dns.c
```

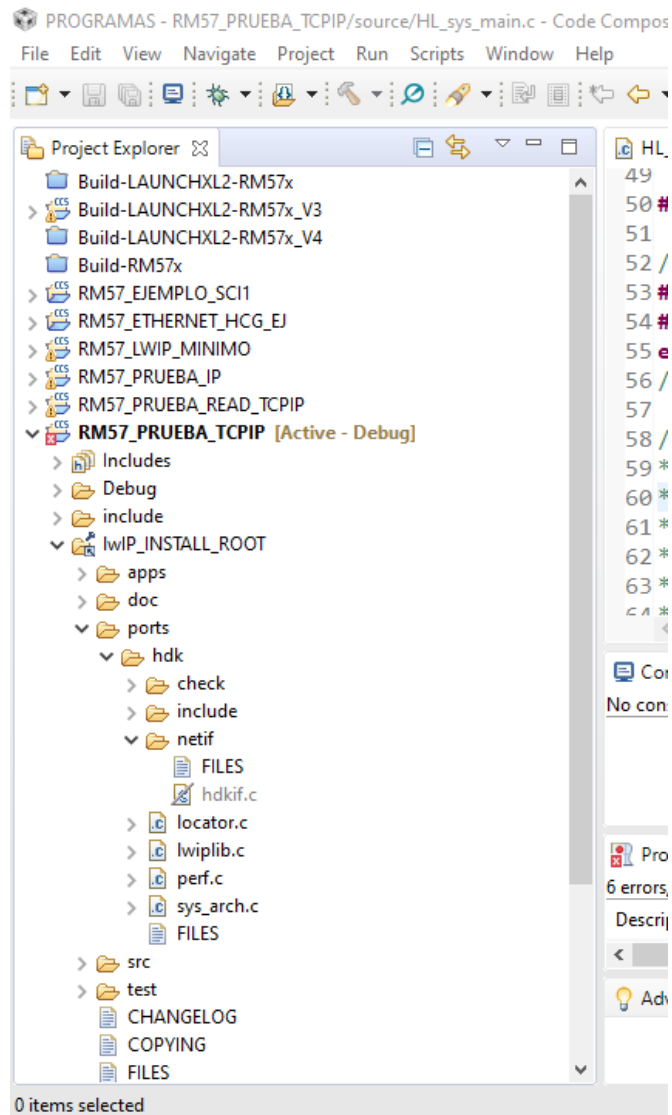
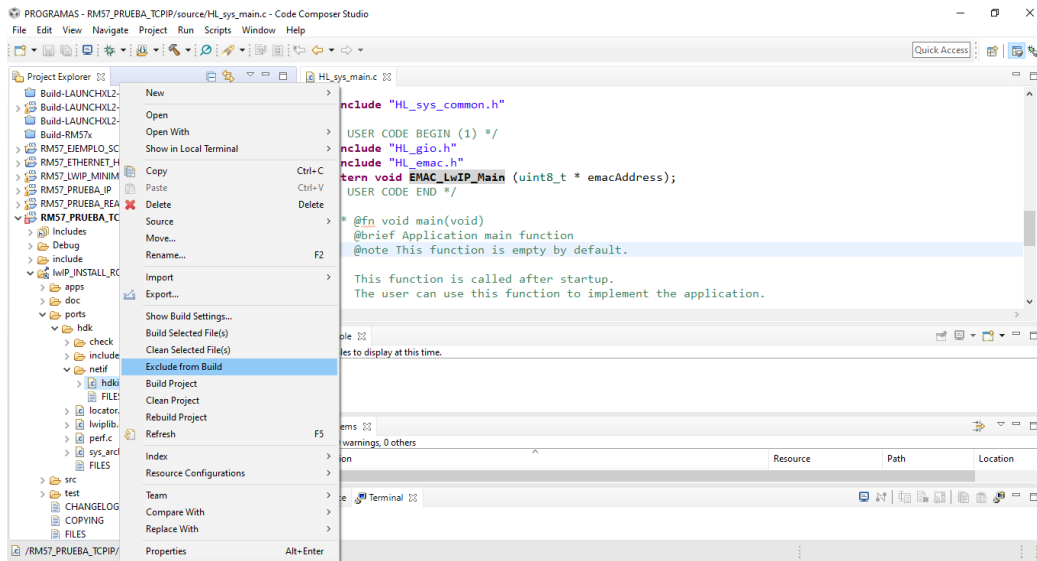
src/core/init.c
src/core/mem.c
src/core/memp.c
src/core/netif.c
src/core/pbuf.c
src/core/raw.c
src/core/stats.c
src/core/sys.c
src/core/tcp_in.c
src/core/tcp_out.c
src/core/tcp.c
src/core/udp.c
src/netif/ppp/auth.c
src/netif/ppp/chap.c
src/netif/ppp/chpms.c
src/netif/ppp/fsm.c
src/netif/ppp/ipcp.c
src/netif/ppp/lcp.c
src/netif/ppp/magic.c
src/netif/ppp/md5.c
src/netif/ppp/pap.c
src/netif/ppp/ppp_oe.c
src/netif/ppp/ppp.c
src/netif/ppp/randm.c
src/netif/ppp/vj.c
src/netif/etharp.c
src/netif/loopif.c
src/netif/slipif.c
src/api

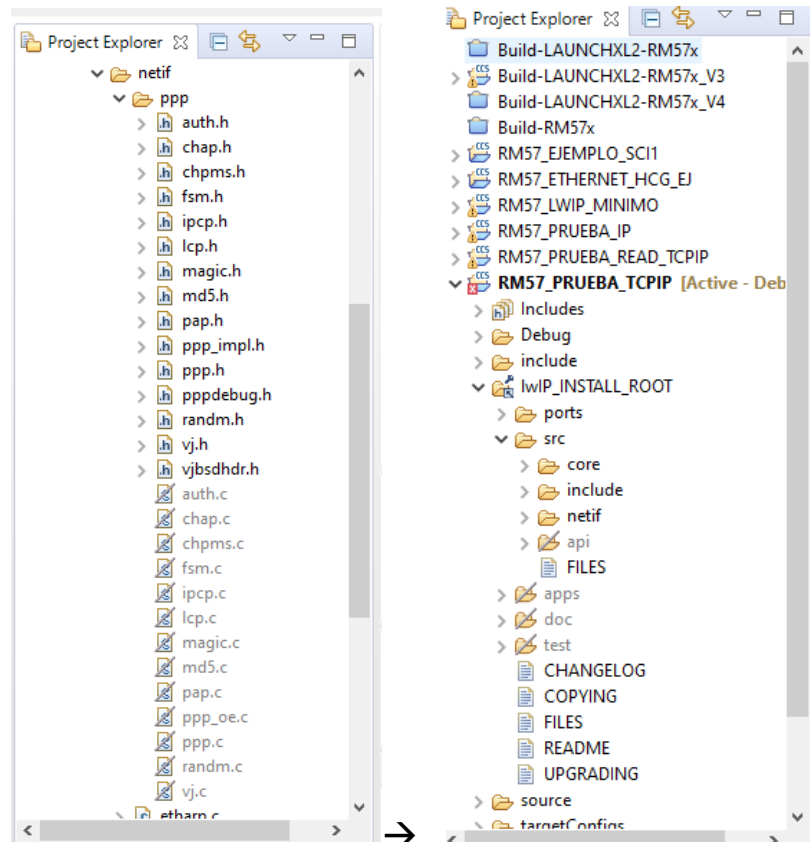
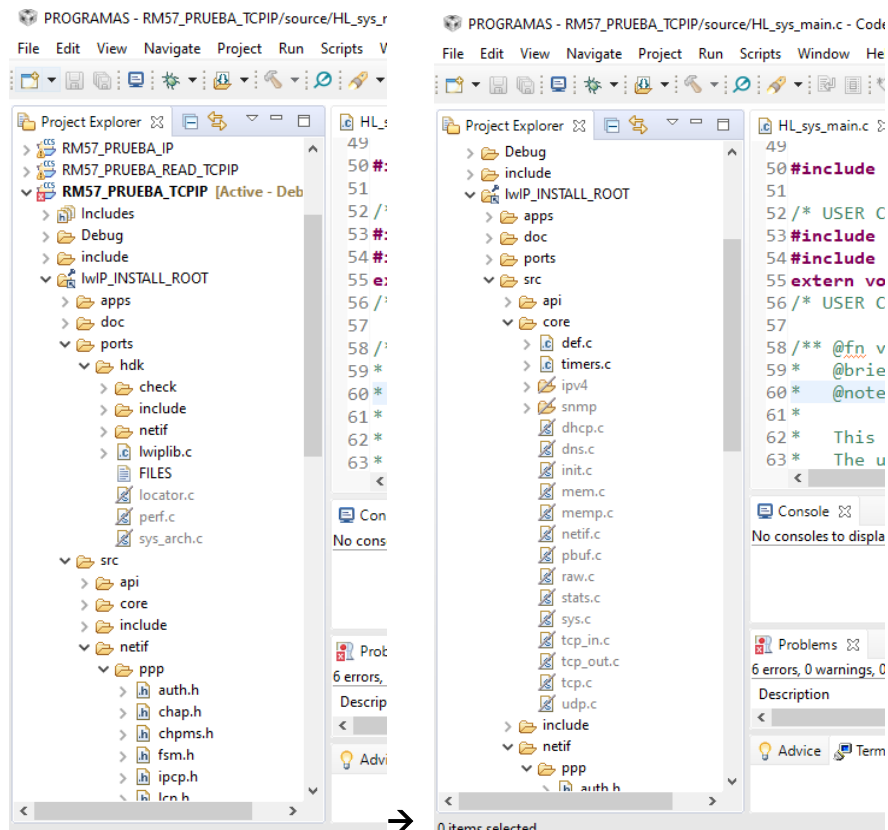
/apps → se recomienda eliminar esta carpeta con clic derecho>delete

/doc → se recomienda eliminar esta carpeta con clic derecho>delete

/test → se recomienda eliminar esta carpeta con clic derecho>delete

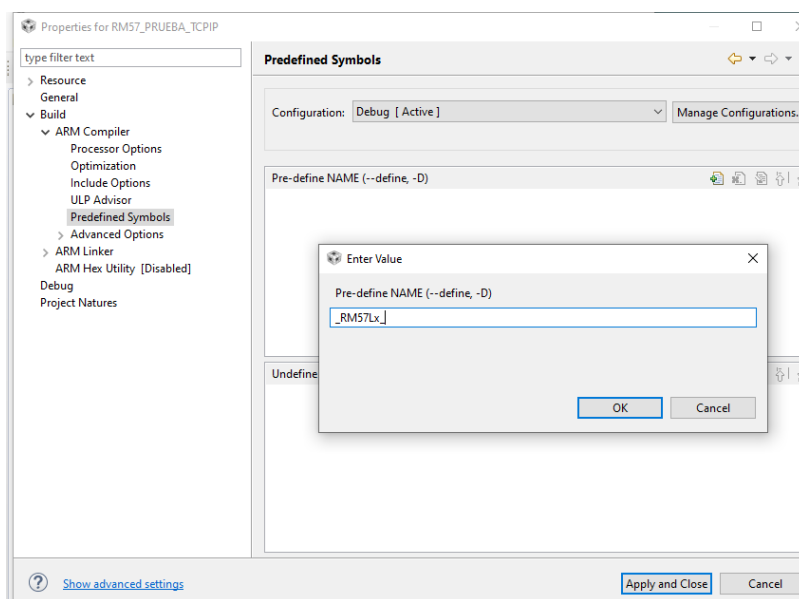
A continuación, se muestra un ejemplo con el primer archivo:



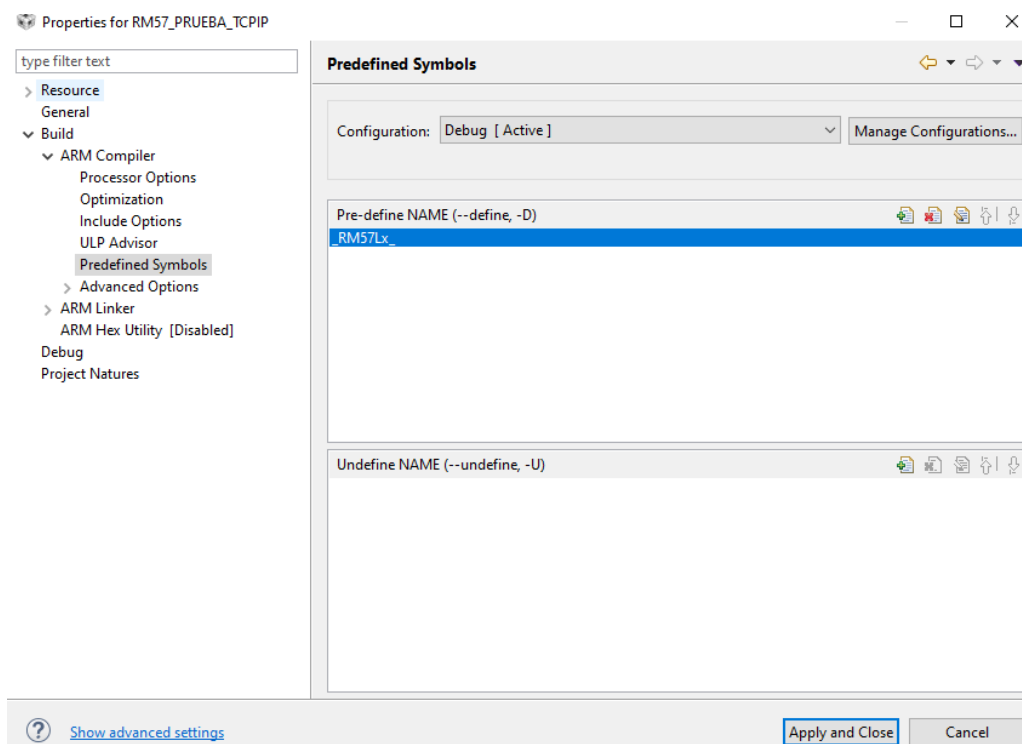


Paso 9: agregar el símbolo del procesador

Se define el procesador usado como un símbolo predefinido. Se realiza esta configuración en el dialogo de propiedades, agregando `_RM57Lx_`



Dar clic en OK, luego apply and close:



Paso 10: agregar lwipopts.h

En este paso se agregará el archivo lwipopts.h a la carpeta “include”. Para esto, se entrará a la carpeta del repositorio de Texas Instruments, que fue descargada en Paso 5: clonar el repositorio.

La dirección, para este caso es:

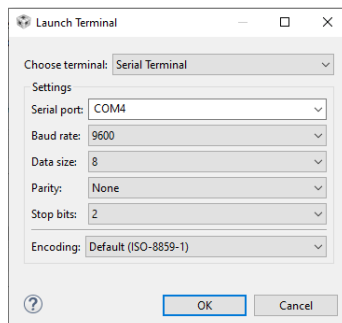
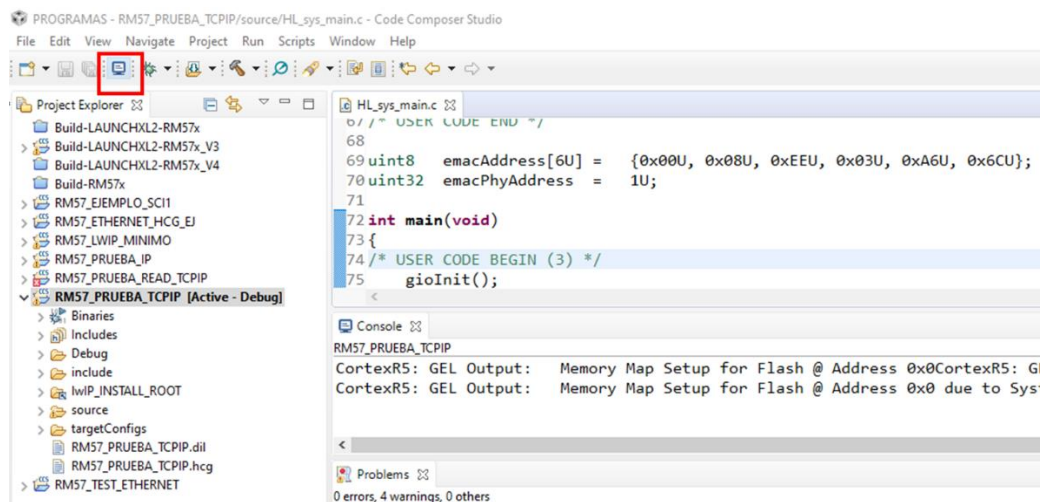
C:\texas_instruments_repositorio\hercules_examples\Application\LwIP\v00.04.00

Una vez, se encuentre en la carpeta, se ingresa a la ruta example\hfk\inc, dar clic derecho y copiar el archivo lwipopts.h

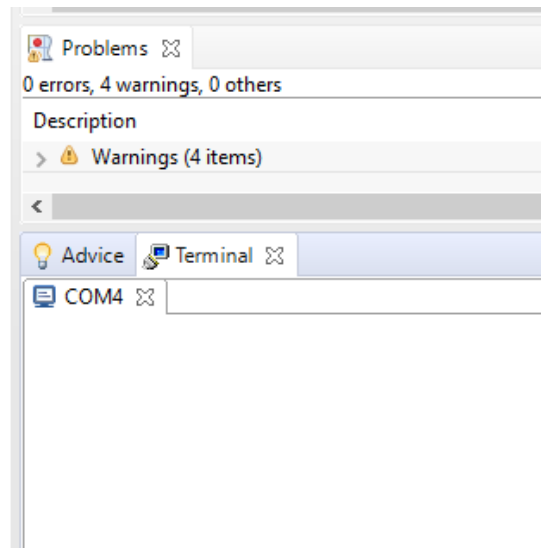
Ahora, en CCS, en el proyecto creado, se procede a pegar el archivo en la carpeta include, con clic derecho.

Paso 11: crear una terminal serial

En CCS, se da clic en el icono de terminal.



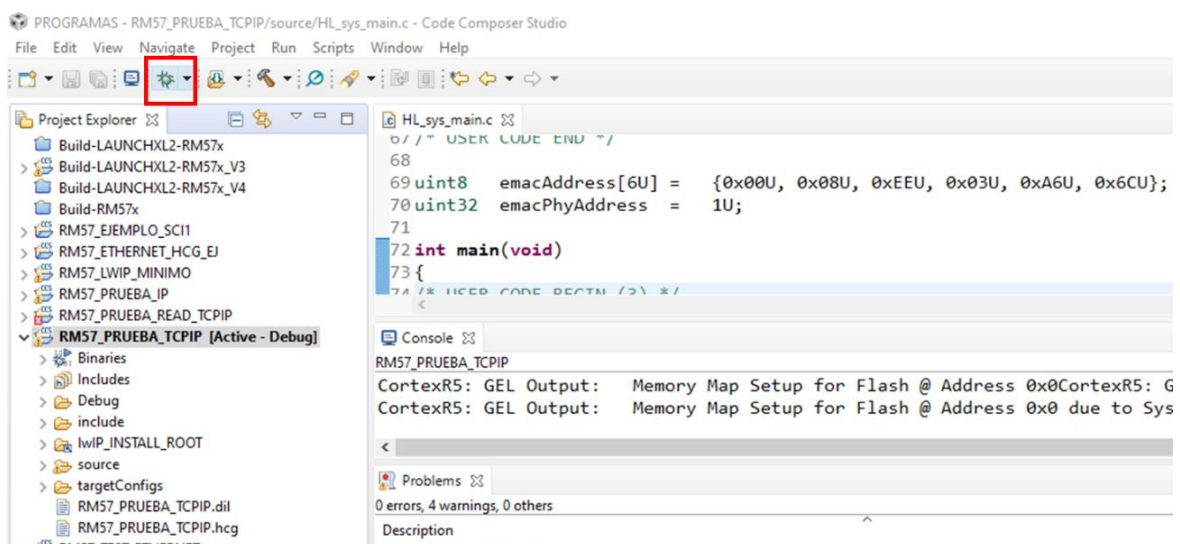
Se creará la terminal de la siguiente forma:



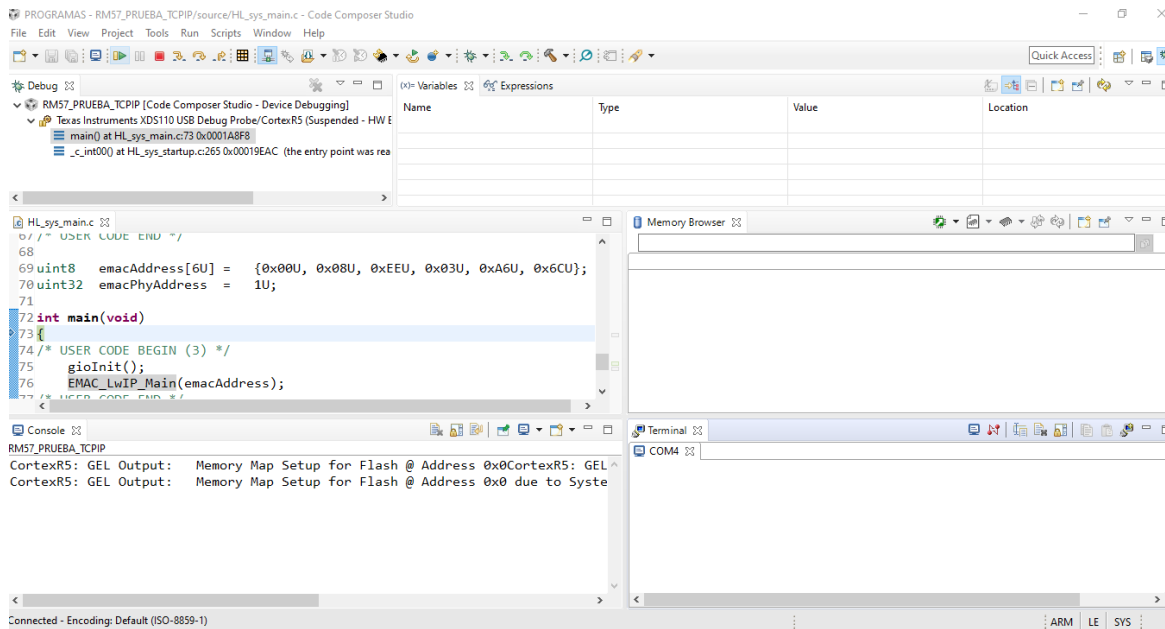
Paso 12: cargar el proyecto en la launchpad RM57

Antes de iniciar la carga del proyecto en la tarjeta, asegúrese de que la tarjeta está conectada al puerto USB de su computador. Además, se requiere que la tarjeta esté conectada con un cable ethernet al router o switch de la red doméstica. El computador y la tarjeta deben pertenecer a la misma red. En este código se ha configurado para que la tarjeta tome la dirección IP asignada por el router.

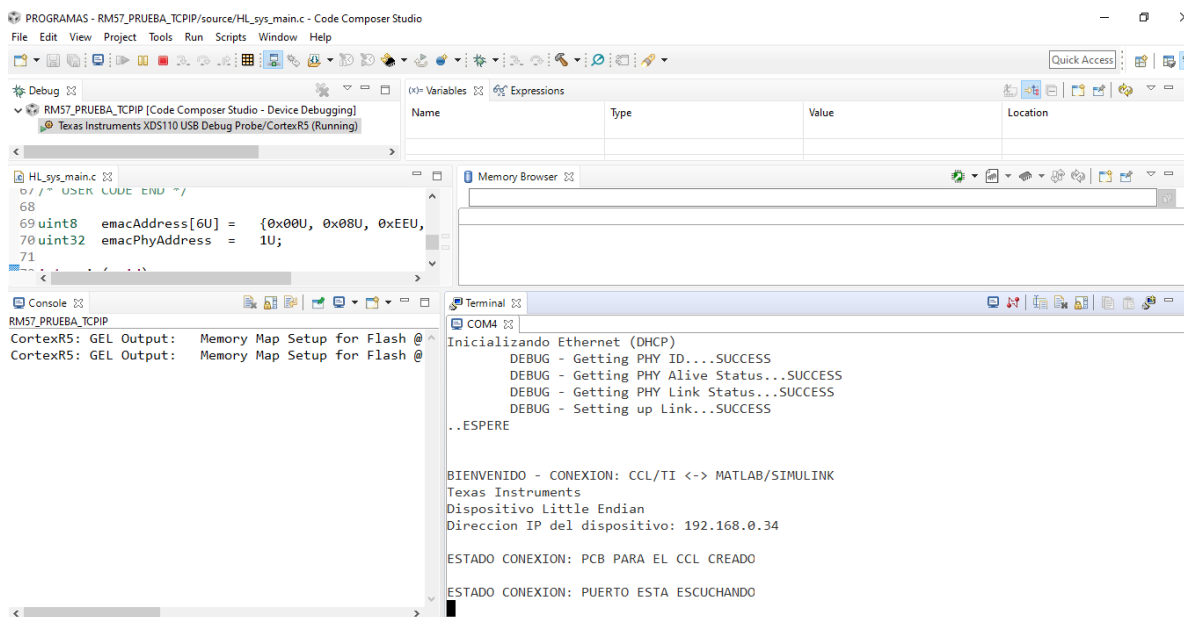
Posterior a estas consideraciones, en CCS, se da clic en el icono de debug.



Se abre la siguiente ventana.



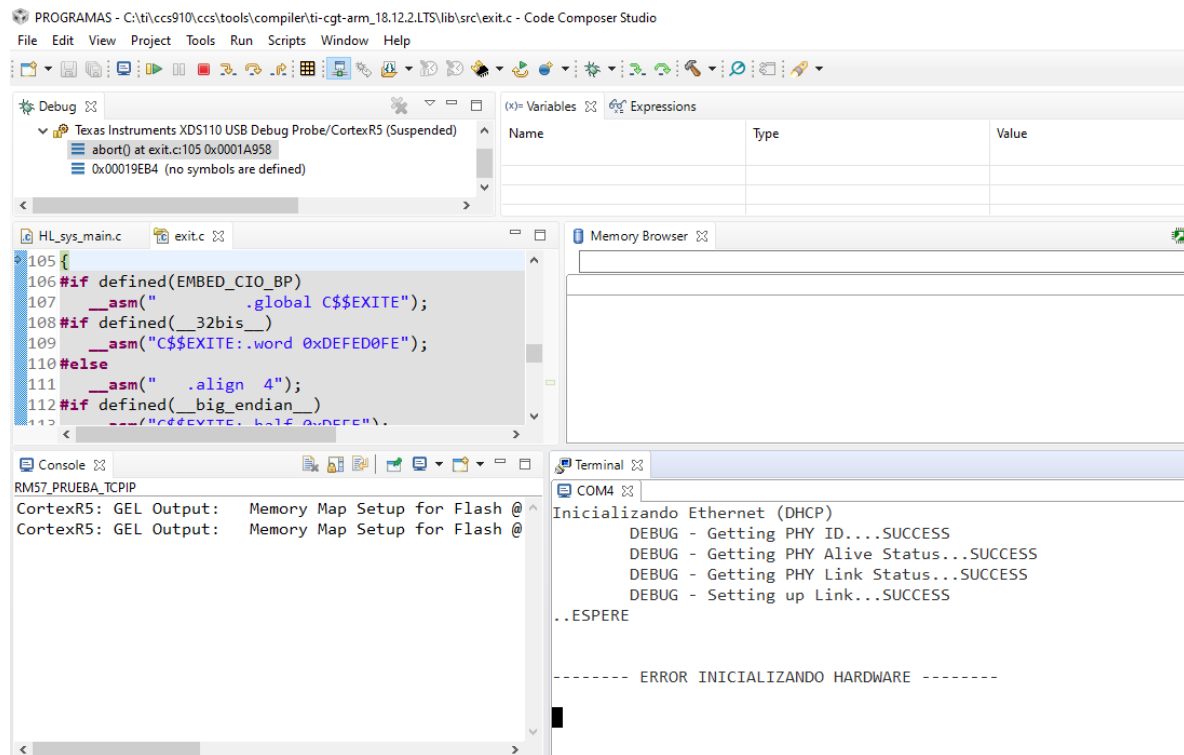
Cuando se da play se obtienen los siguientes mensajes en terminal:



La tarjeta está lista para iniciar la conexión con otro dispositivo. Para ello, la tarjeta muestra su dirección IP.

Paso 13: posible error en la ejecución del proyecto en la launchpad RM57

Es posible que al ejecutar el proyecto se muestre el mensaje de error en terminal.



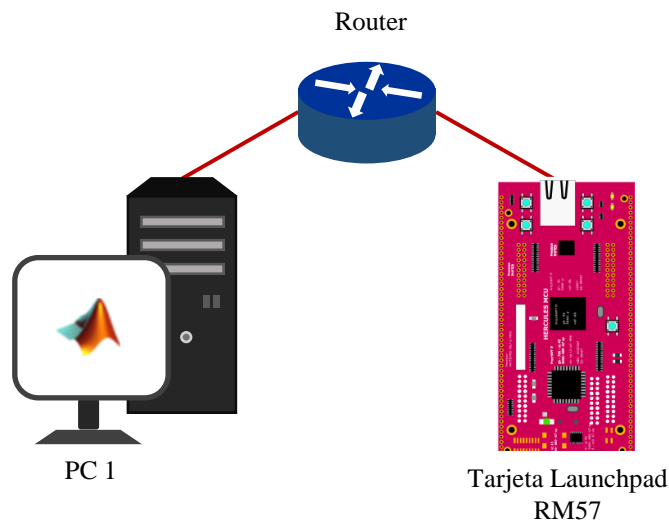
Este error es muy común, así que la recomendación es terminar el debug (ctrl+F2). Primero se debe asegurar que el cable ethernet entre los dos extremos está conectado adecuadamente. Si lo requiere, puede conectar y desconectar el cable, así el servicio DHCP del router puede asignar nuevamente una dirección IP. Posterior a este paso se procede a presionar el pulsador de RESET que posee la tarjeta e intentar cargar el proyecto de nuevo.

Realizar este paso tantas veces sea necesario.

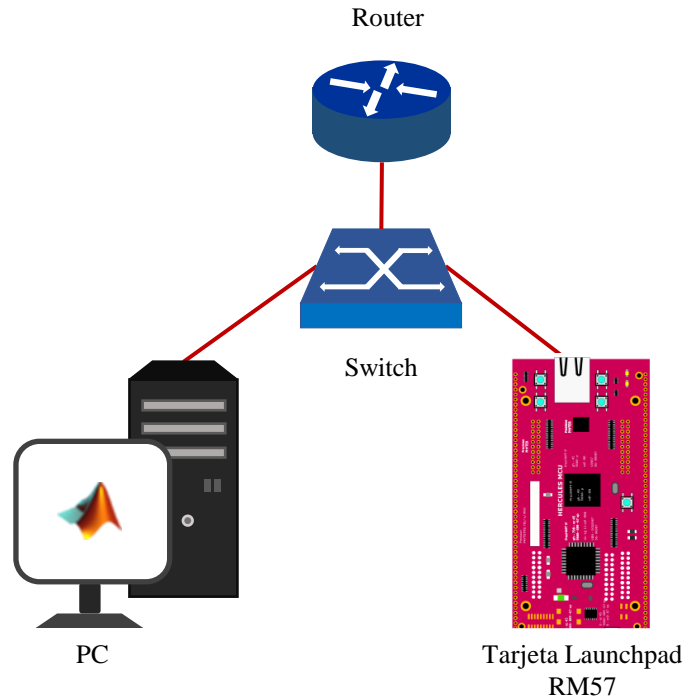
Interconexión con Matlab/Simulink

Paso 1: conexión entre el PC y la tarjeta TI

En la figura se realiza la conexión entre el computador y la tarjeta TI a través de un router fast/gigabit ethernet, que está disponible en las conexiones de internet del hogar. Los cables de red deben ser compatibles con gigabit ethernet, por lo tanto, pueden ser de categoría 5e o superior. Debido a que el router puede asignar dinámicamente las direcciones IPv4 a cada dispositivo conectado a él, no es necesario configurar una red LAN privada.



En la siguiente figura se realiza la conexión entre el computador y la tarjeta TI a través de un switch fast/gigabit ethernet, que está conectado a un router fast/gigabit ethernet. Esta conexión se realiza cuando el router solo cuenta con una interfaz para conectar el cable, así que el switch permite conectar más de dos equipos. Los cables de red deben ser compatibles con gigabit ethernet, por lo tanto, pueden ser de categoría 5e o superior. Debido a que el router puede asignar dinámicamente las direcciones IPv4 a cada dispositivo conectado a él, no es necesario configurar una red LAN privada.



Paso 2: ejecutar el proyecto en CCS

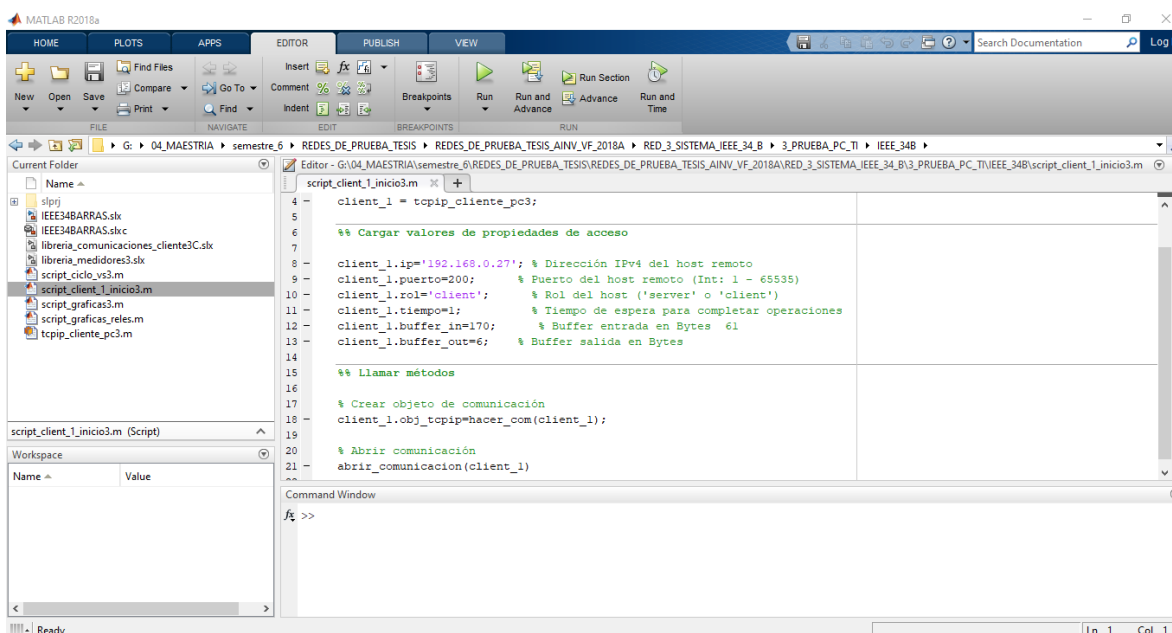
Se carga el proyecto en la tarjeta como se expone en Paso 12: cargar el proyecto en la launchpad RM57. Cuando la tarjeta haya habilitado la transmisión de datos, se procede a copiar la dirección IP del dispositivo.

The screenshot shows the Code Composer Studio (CCS) interface. The Project Explorer on the left lists the project files for 'RM57_TESIS_RED_1'. The main window displays the source code for 'HL_sys_main.c'. The execution log on the right shows the following messages:

```
ESTADO CONEXION: PCB PARA EL CCL CREADO
ESTADO CONEXION: PUERTO ESTA ESCUCHANDO
Iniciando Ethernet (DRCP)
DEBUG - Getting PHY ID...SUCCESS
DEBUG - Getting PHY Alive Status...SUCCESS
DEBUG - Getting PHY Link Status...SUCCESS
DEBUG - Setting up Link...SUCCESS
..ESPERE
BIENVENIDO - CONEXION: CCL/TI <-> MATLAB/SIMULINK
Texas Instruments
Dispositivo Little Endian
Direccion IP del dispositivo: 192.168.0.27
ESTADO CONEXION: PCB PARA EL CCL CREADO
ESTADO CONEXION: PUERTO ESTA ESCUCHANDO
```

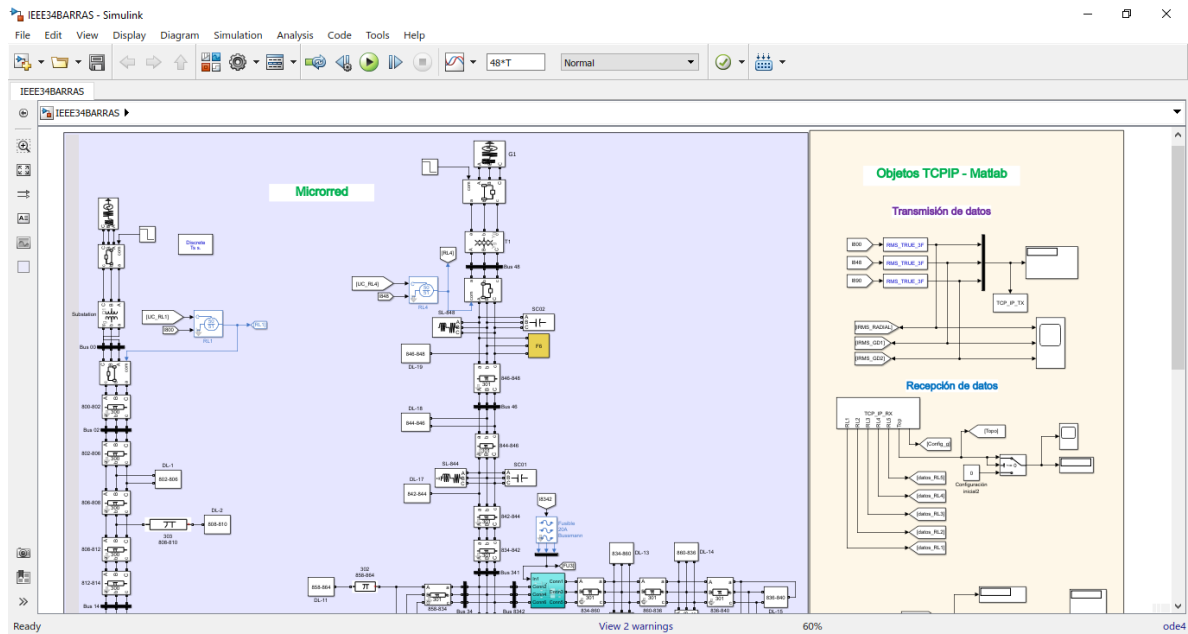
Paso 3: ajustes en Matlab

Junto a esta guía está la carpeta llamada IEEE_34B. Se abre el script de Matlab® titulado *script_client1_inicio3.m*. En la variable *client_1.ip* se debe ingresar la dirección IP de la tarjeta y la variable *puerto*, con el número 200.

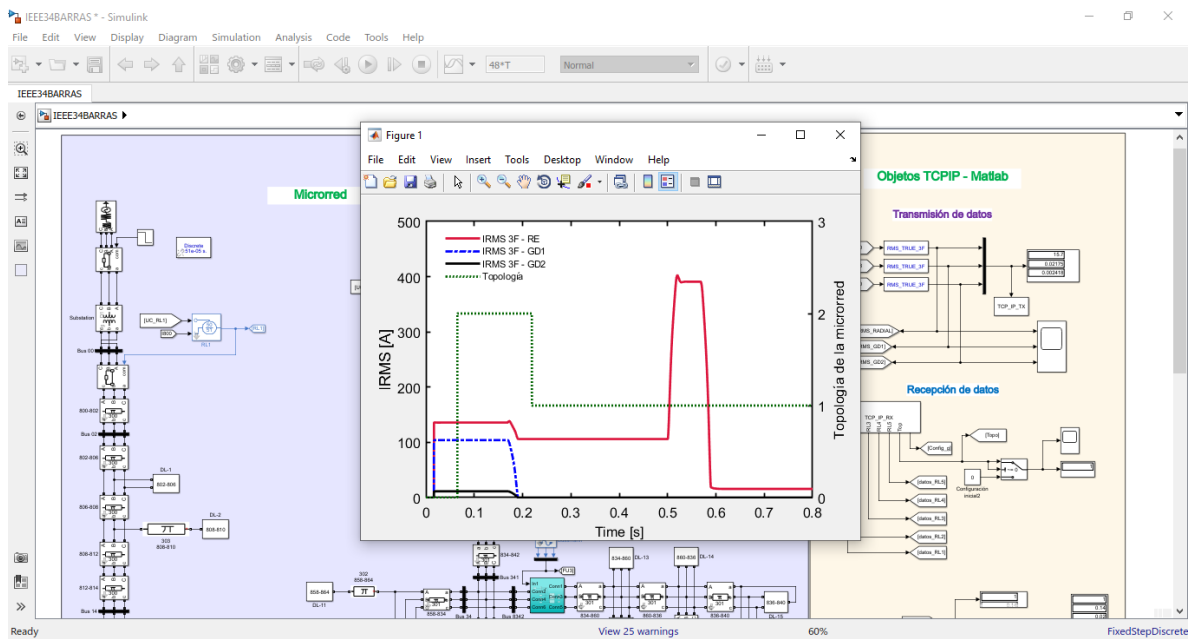


Paso 4: ejecutar Simulink

Dado que el servidor tiene en “ESTADO CONEXION: PUERTO ESTA ESCUCHANDO” ahora se procede a abrir y ejecutar el modelo *IEEE34BARRAS.slx*, presionando el botón *RUN* en Simulink®.



Cuando la simulación haya terminado, se puede obtener la gráfica ejecutando *script_graficas3.m*



Referencias

- [1] E. Ding, “HALCoGen Ethernet Driver With lwIP Integration Demo and Active Web Server Demo,” 2019. .
- [2] J. Cumps, “Minimal lwIP Project for Hercules LaunchPad,” 2018. <https://www.hackster.io/jancumps/minimal-lwip-project-for-hercules-launchpad-66d889> (accessed Jun. 12, 2020).
- [3] Texas Instruments, “Software Examples for Hercules Processors,” 2019. https://git.ti.com/cgit/hercules_examples/hercules_examples/tree/Application/LwIP/v00.04.00 (accessed Jun. 01, 2020).
- [4] S. Goldschmidt and D. Ziegelmeier, “lwIP - A Lightweight TCP/IP stack - Sumario.” <http://savannah.nongnu.org/projects/lwip/> (accessed Jun. 12, 2020).
- [5] Savannah Repository, “TCP Echo raw.” http://cvs.savannah.gnu.org/viewvc/lwip/contrib/apps/tcpecho_raw/ (accessed Jun. 12, 2020).