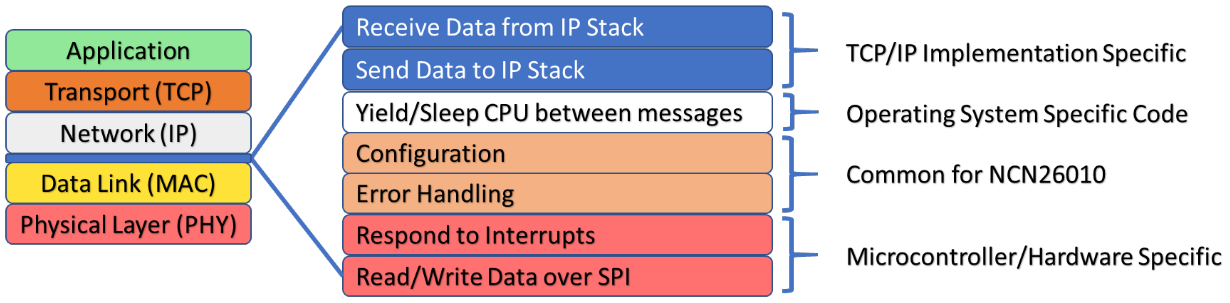
NCN26010 Example Software Users Guide

Version 1.0.0

The NCN26010 device is an Ethernet Transceiver designed for industrial multi-drop Ethernet. It provides all physical layer functions needed to transmit and receive data over a single unshielded twisted pair. The example software provided by onsemi is designed to provide the necessary firmware and application examples to help designers quickly integrate the NCN26010 into their systems.

Software Architecture

The provided software follows a modular design to allow the software to be quickly adapted to be used with any microcontroller or to be used with various implementations of the TCP and IP layers of the network stack. As illustrated below, the standard network stack consists of five layers: Physical (PHY), Data Link (MAC), Network (IP), Transport (TCP), and Application. The NCN26010 provides a hardware implementation of the MAC and PHY layers of the network stack. There are many existing open source or proprietary implantations of the TCP and IP layers that a designer may choose from. The software provided with the NCN26010 is designed to provide the connection between the IP and MAC layers of the stack.



The provided software includes three required modules and one optional module. The required modules include a module to handle interaction with the TCP/IP Implementation, a module to handle hardware specific details, and the core driver module that configures the NCN26010 and handles errors. The optional module handles interactions with the operating system to yield the CPU to other tasks when not processing messages. If no OS module is provided the system defaults to call functions in the hardware specific module to sleep the CPU until the NCN26010 signals data is available with an interrupt. The interfaces for these modules are described in T1S_TCP-IP.h, T1S_Hardware.h, NCN26010.h and T1S_OS.h, respectively.

The core driver module remains unchanged for any combination of hardware and TCP/IP implementations. The hardware specific module and the TCP/IP modules are intended to be implemented by the system designer to match their system. We provide the following example implementations to support systems with the following hardware and TCP/IP implementations:

		Microcontroller		
		Raspberry Pi (Zero, 3B)	STM32 (L4xx)	RSL10
TCP/IP	Linux TAP	✓		
	Lightweight IP (LwIP)		✓	✓
	FreeRTOS TCP-IP		✓	✓

If other microcontrollers or TCP/IP implementations are required for the system, only the functions defined in T1S_Hardware.h or T1S_TCP-IP.h need to be implemented.

Installing the software

Raspberry Pi

If you are using the NCN26010XMNEVK getting the system up and running is simple.

1. Install the latest version of Raspberry Pi OS onto a SD card as described at https://www.raspberrypi.com/software/
2. Download and extract the Raspberry Pi NCN26010 software package on the raspberry pi. This can be accomplished with the following command:
```
curl -o NCN26010Firmware.zip
"https://www.onsemi.com/pub/Collateral/EXAMPLE%20SOFTWARE:%20LINUX%20IMPLEMENTA
TION.ZIP" && \
unzip NCN26010Firmware.zip && \
rm NCN26010Firmware.zip
```
3. Move into the newly created directory
```
cd NCN26010_Example_Software
```
4. Run the install script and follow the prompts:
```
./install.sh
```
5. Execute the compiled program
```
sudo ./NCN26010_Driver
```
6. In a separate terminal test by pinging a device on the network (We recommend repeating these steps with a second evaluation kit)
```
ping 192.1.1.1
```

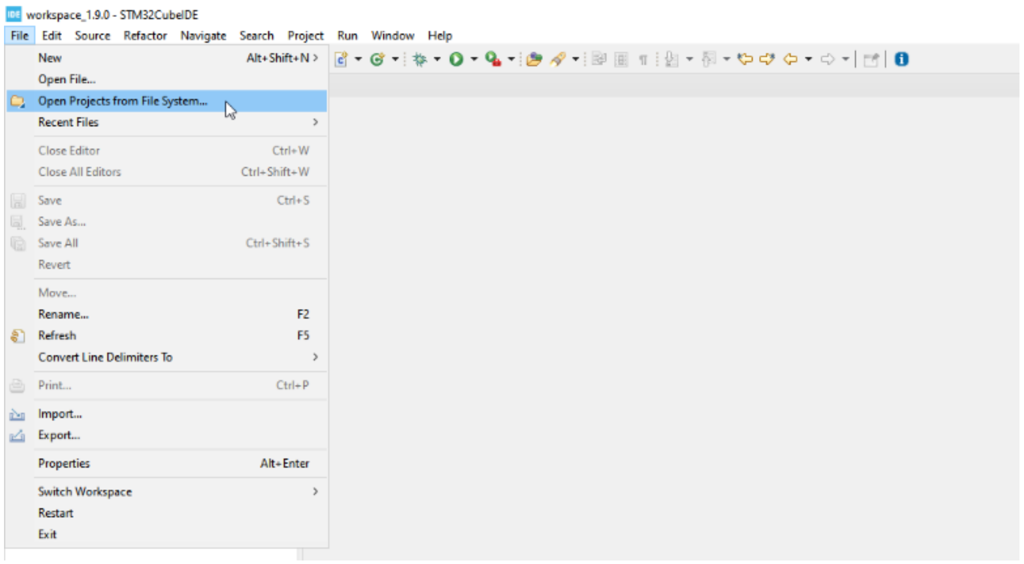
STM32

Install STM32CubeIDE

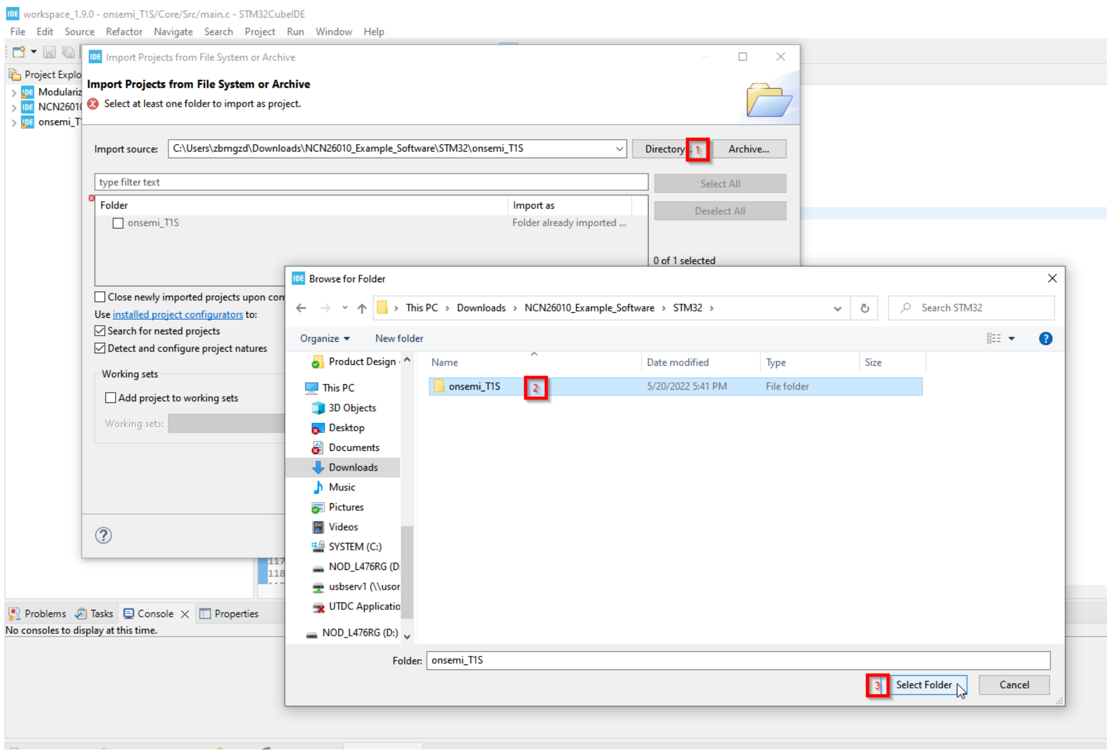
Follow the instructions provided by ST Micro here:
https://www.st.com/resource/en/user_manual/um2563-stm32cubeide-installation-guide-stmicroelectronics.pdf

Open the Configured Example

1. Under file, select “Open Projects from File System…”



2. Select Directory and navigate to the folder called onsemi_T1S


workspace_1.9.0 - onsemi_T1S/Core/Src/main.c - STM32CubeIDE
Import Projects from File System or Archive
Directory...
1
Archive...
Select All
Deselect All
Browse for Folder
This PC › Downloads › NCN26010_Example_Software › STM32 ›
onsemi_T1S
2
3
Select Folder
Cancel

Import Projects from File System or Archive
Select at least one folder to import as project.
Import source: C:\Users\zbmgzd\Downloads\NCN26010_Example_Software\STM32\onsemi_T1S

Folder	Import as
☐ onsemi_T1S	Folder already imported ...

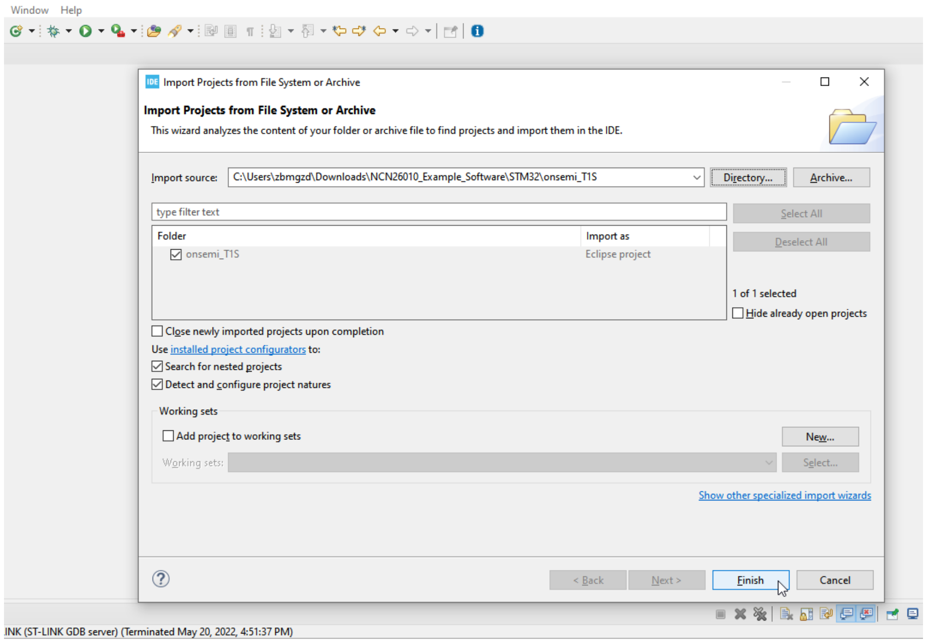
0 of 1 selected

- ☐ Close newly imported projects upon com
- Use installed project configurators to:
- ☑ Search for nested projects
- ☑ Detect and configure project natures
- Working sets
- ☐ Add project to working sets

Name	Date modified	Type	Size
onsemi_T1S	5/20/2022 5:41 PM	File folder	

Folder: onsemi_T1S

3. Leave the default values and select finish.



Import Projects from File System or Archive

Import Projects from File System or Archive

This wizard analyzes the content of your folder or archive file to find projects and import them in the IDE.

Import source: C:\Users\zbmgzd\Downloads\NCN26010_Example_Software\STM32\onsemi_T1S

Folder	Import as
☑ onsemi_T1S	Eclipse project

1 of 1 selected

☐ Hide already open projects

☐ Close newly imported projects upon completion

Use installed project configurators to:

☑ Search for nested projects

☑ Detect and configure project natures

Working sets

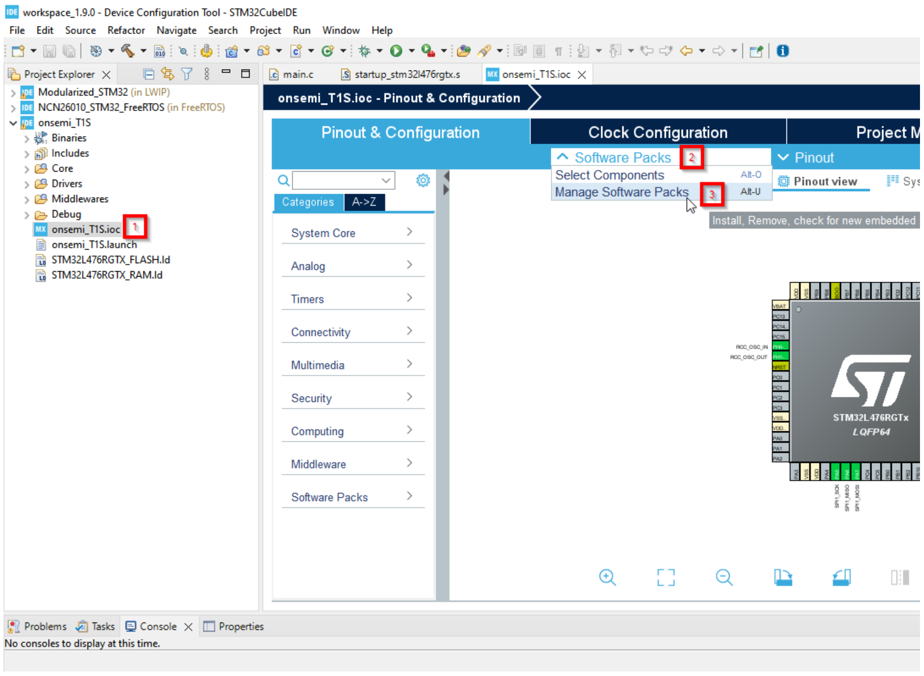
☐ Add project to working sets

Working sets:

INK (ST-LINK GDB server) (Terminated May 20, 2022, 4:51:37 PM)

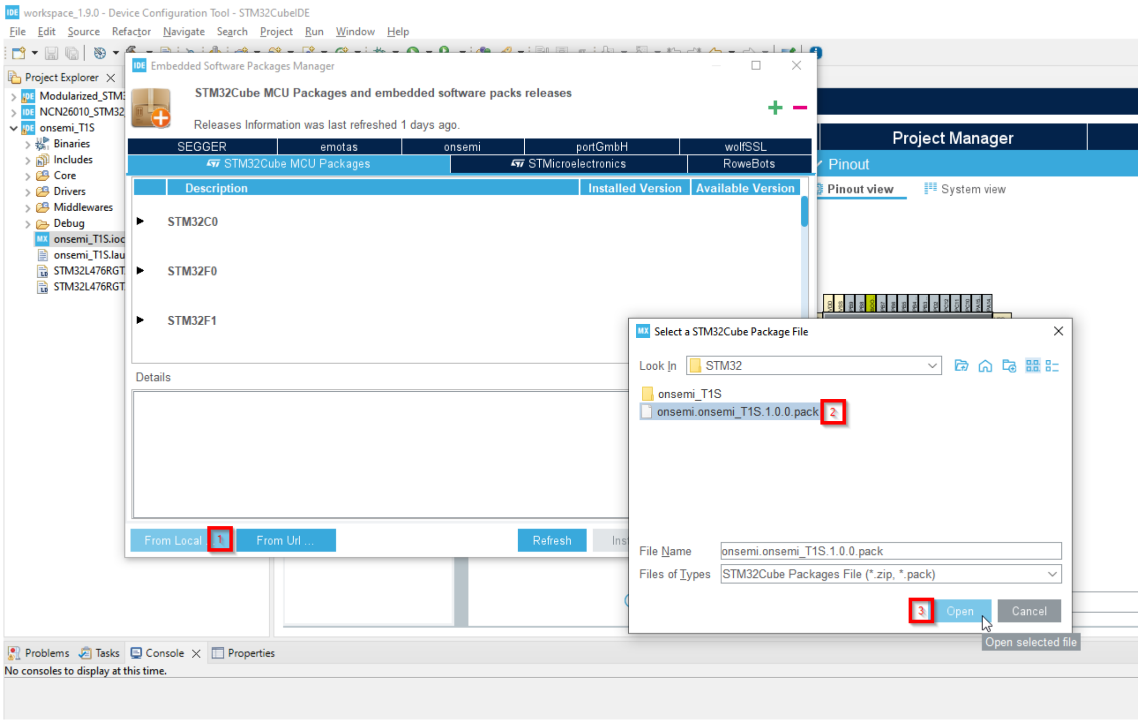
Install the onsemi_T1S Software Pack

1. Double click onsemi_T1S.ioc, once the GUI loads select “Software Packs,” then Manage Software Packs

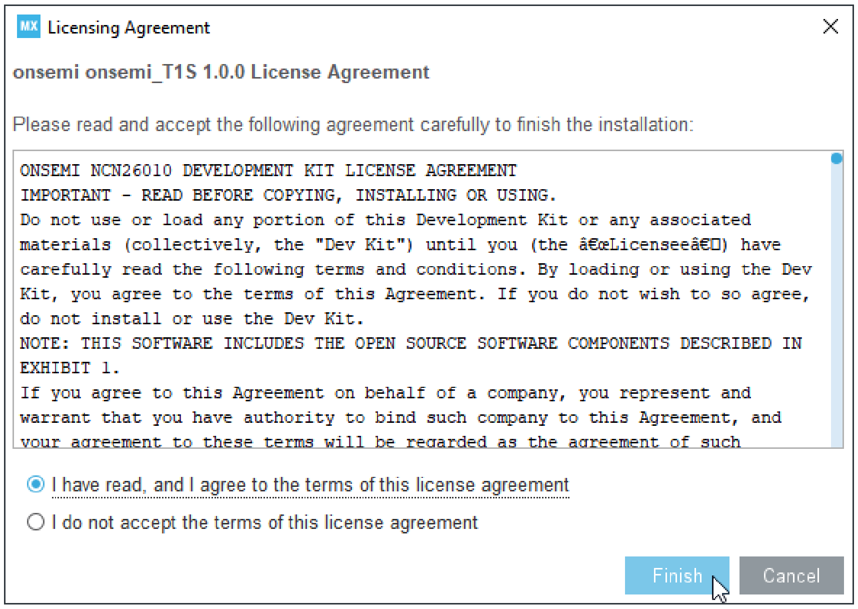


No consoles to display at this time.

2. Select “From Local…” then navigate to onsemi.onsemi_T1S.1.0.0.pack, and open it.



3. Read and Accept the License Agreement then close the manage dialog.



onsemi onsemi_T1S 1.0.0 License Agreement

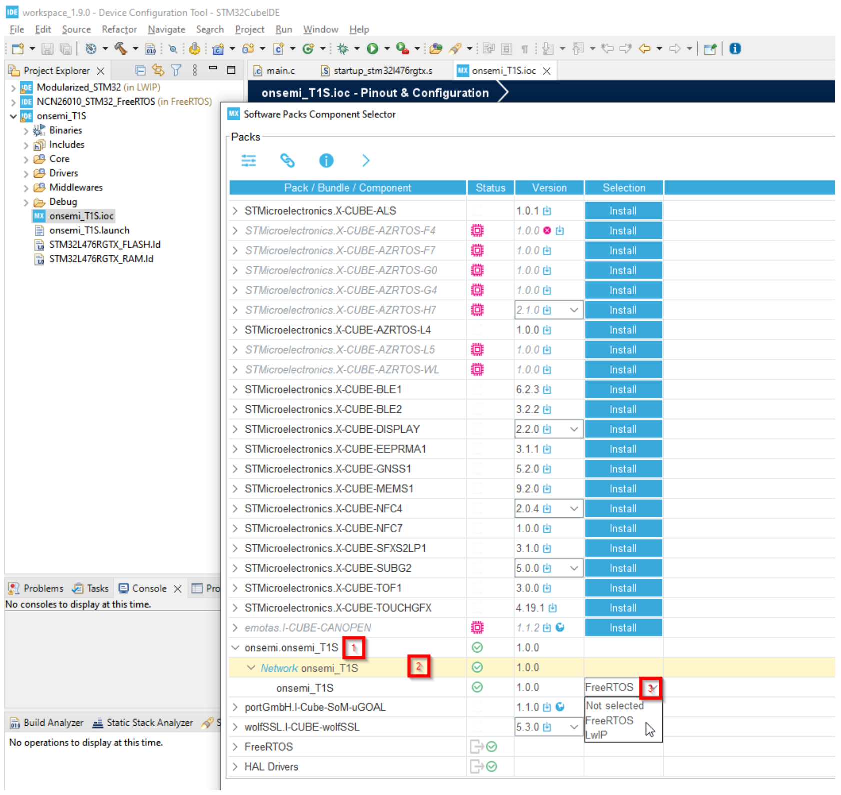
Please read and accept the following agreement carefully to finish the installation:

```
ONSEMI NCN26010 DEVELOPMENT KIT LICENSE AGREEMENT
IMPORTANT - READ BEFORE COPYING, INSTALLING OR USING.
Do not use or load any portion of this Development Kit or any associated
materials (collectively, the "Dev Kit") until you (the â€œLicenseeâ€□) have
carefully read the following terms and conditions. By loading or using the Dev
Kit, you agree to the terms of this Agreement. If you do not wish to so agree,
do not install or use the Dev Kit.
NOTE: THIS SOFTWARE INCLUDES THE OPEN SOURCE SOFTWARE COMPONENTS DESCRIBED IN
EXHIBIT 1.
If you agree to this Agreement on behalf of a company, you represent and
warrant that you have authority to bind such company to this Agreement, and
your agreement to these terms will be regarded as the agreement of such
```

[✓ I have read, and I agree to the terms of this license agreement]

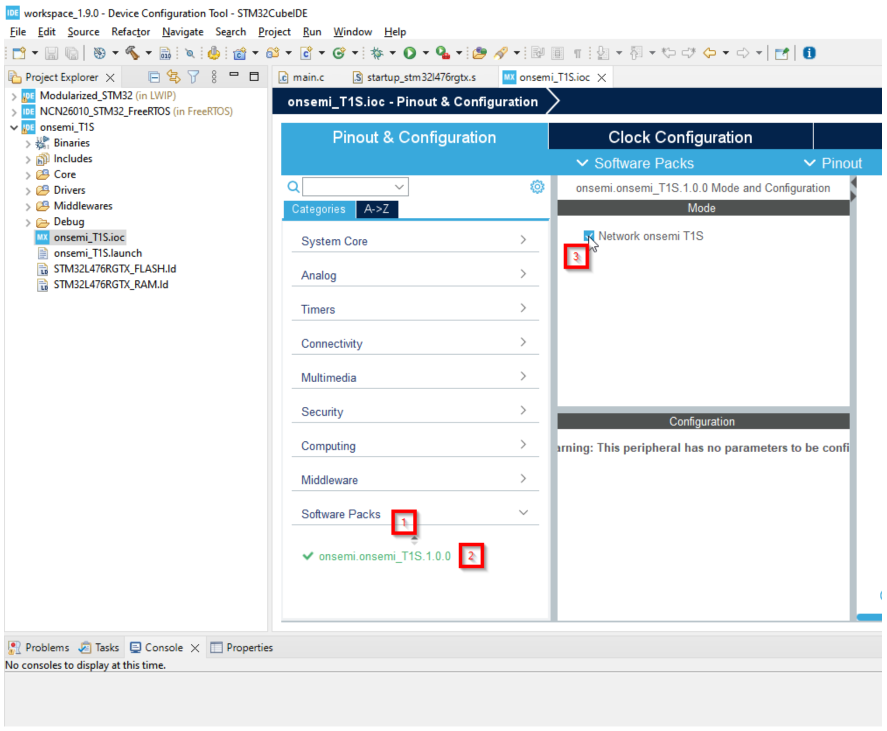
I do not accept the terms of this license agreement

4. Then select “Software Packs” again but this time click “Select Components.” Navigate to onsemi.onsemi_T1S, Network onsemi_T1S, then select either FreeRTOS or LwIP depending on which TCP/IP implementation you would like to use.


workspace_1.9.0 - Device Configuration Tool - STM32CubeIDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
Modularized_STM32 (in LWIP)
NCN26010_STM32_FreeRTOS (in FreeRTOS)
onsemi_T1S
Binaries
Includes
Core
Drivers
Middlewares
Debug
onsemi_T1S.ioc
onsemi_T1S.launch
STM32L476RGTX_FLASH.ld
STM32L476RGTX_RAM.ld
Problems Tasks Console Pro
No consoles to display at this time.
Build Analyzer Static Stack Analyzer
No operations to display at this time.
main.c
startup_stm32l476rgtx.s
onsemi_T1S.ioc
onsemi_T1S.ioc - Pinout & Configuration
Software Packs Component Selector
Packs
1
2
3

Pack / Bundle / Component	Status	Version	Selection
STMicroelectronics.X-CUBE-ALS		1.0.1	Install
STMicroelectronics.X-CUBE-AZRTOS-F4		*1.0.0*	Install
STMicroelectronics.X-CUBE-AZRTOS-F7		*1.0.0*	Install
STMicroelectronics.X-CUBE-AZRTOS-G0		*1.0.0*	Install
STMicroelectronics.X-CUBE-AZRTOS-G4		*1.0.0*	Install
STMicroelectronics.X-CUBE-AZRTOS-H7		*2.1.0*	Install
STMicroelectronics.X-CUBE-AZRTOS-L4		1.0.0	Install
STMicroelectronics.X-CUBE-AZRTOS-L5		*1.0.0*	Install
STMicroelectronics.X-CUBE-AZRTOS-WL		*1.0.0*	Install
STMicroelectronics.X-CUBE-BLE1		6.2.3	Install
STMicroelectronics.X-CUBE-BLE2		3.2.2	Install
STMicroelectronics.X-CUBE-DISPLAY		2.2.0	Install
STMicroelectronics.X-CUBE-EEPRMA1		3.1.1	Install
STMicroelectronics.X-CUBE-GNSS1		5.2.0	Install
STMicroelectronics.X-CUBE-MEMS1		9.2.0	Install
STMicroelectronics.X-CUBE-NFC4		2.0.4	Install
STMicroelectronics.X-CUBE-NFC7		1.0.0	Install
STMicroelectronics.X-CUBE-SFXS2LP1		3.1.0	Install
STMicroelectronics.X-CUBE-SUBG2		5.0.0	Install
STMicroelectronics.X-CUBE-TOF1		3.0.0	Install
STMicroelectronics.X-CUBE-TOUCHGFX		4.19.1	Install
emotas.I-CUBE-CANOPEN		*1.1.2*	Install
onsemi.onsemi_T1S	✓	1.0.0	
Network onsemi_T1S	✓	1.0.0	
onsemi_T1S	✓	1.0.0	FreeRTOS
portGmbH.I-Cube-SoM-uGOAL		1.1.0	Not selected
wolfSSL.I-CUBE-wolfSSL		5.3.0	FreeRTOS LwIP
FreeRTOS	✓		
HAL Drivers	✓		

5. Enable the onsemi_T1S software pack by selecting “Software Packs,” “onsemi_T1S,” then check the box next to “Network onsemi T1S”



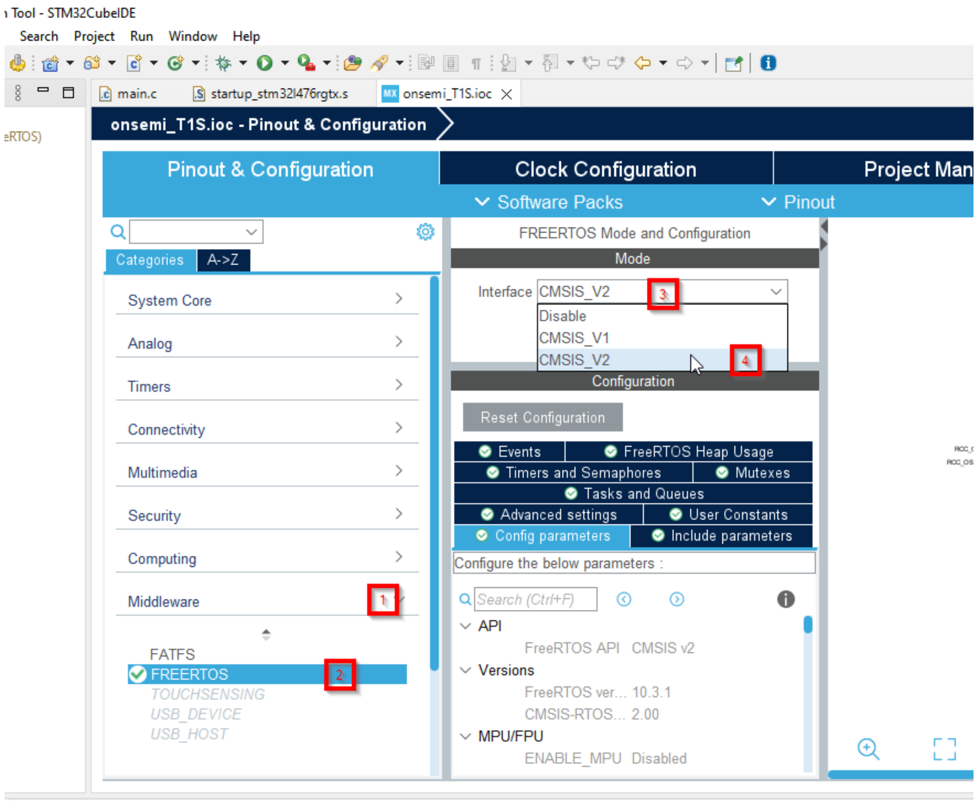
[✓ Network onsemi T1S]

rning: This peripheral has no parameters to be confi

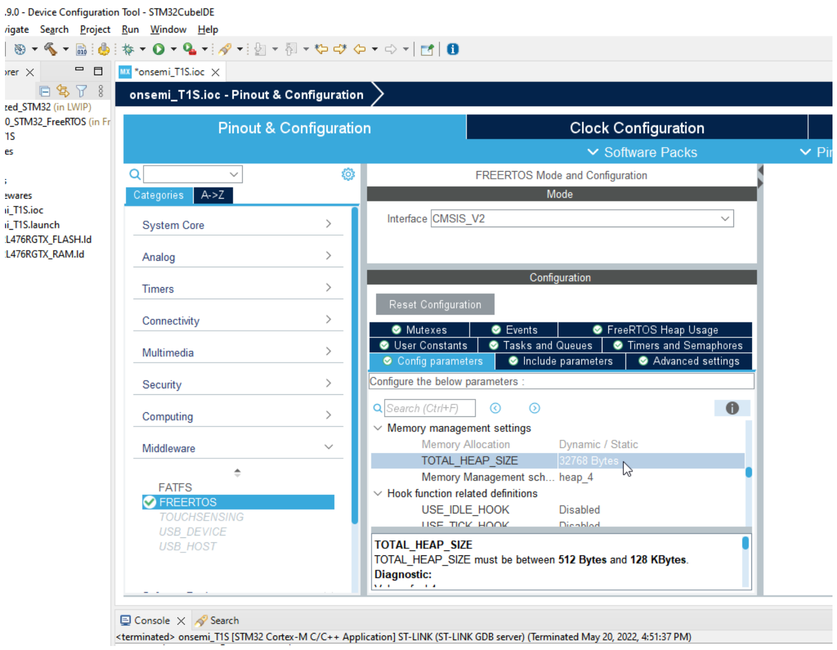
Setup FreeRTOS

If you selected the LwIP TCP/IP stack, you can skip this section.

1. Enable FreeRTOS under "Middleware" "FreeRTOS." Then select "CMSIS_V2."

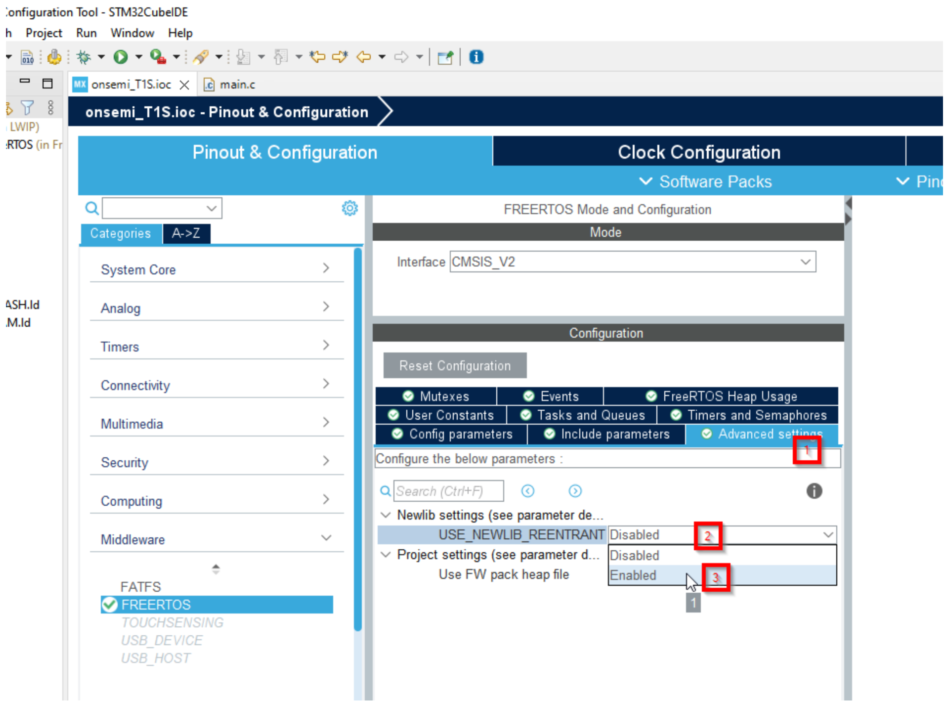


2. In the FreeRTOS settings under "Config parameters", adjust the heap size. We suggest 32K bytes.



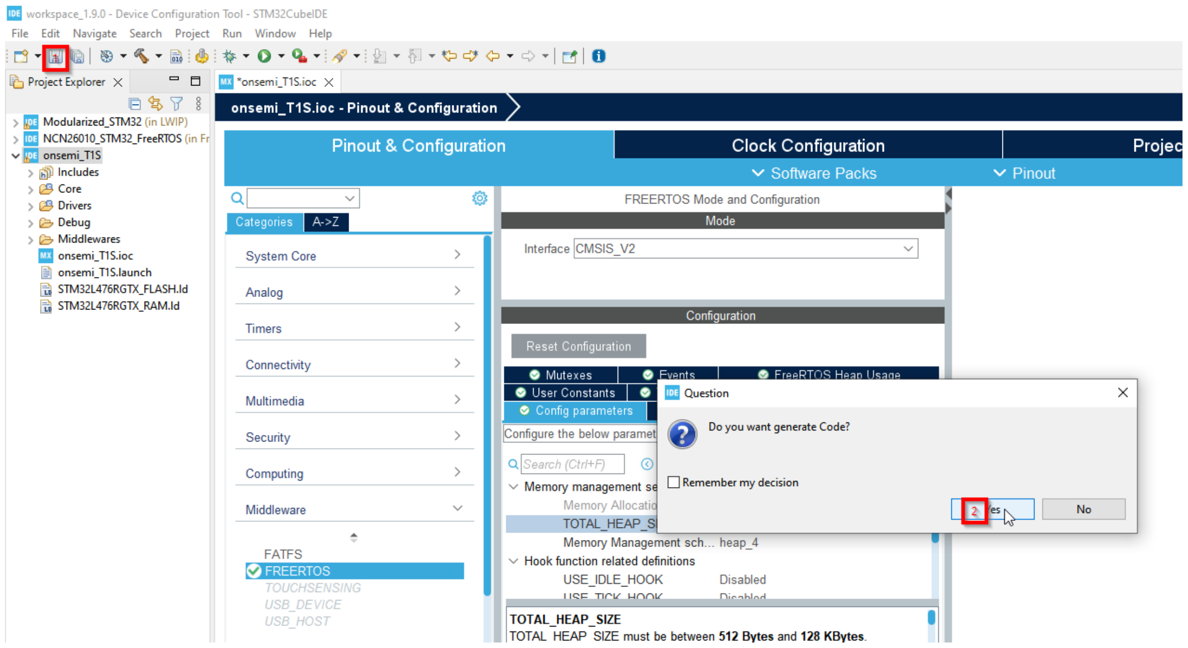
Config parameters
Memory management settings
Memory Allocation Dynamic / Static
TOTAL_HEAP_SIZE 32768 Bytes
Memory Management sch... heap_4

3. Under advanced settings Enable the FreeRTOS Reentrant mode.

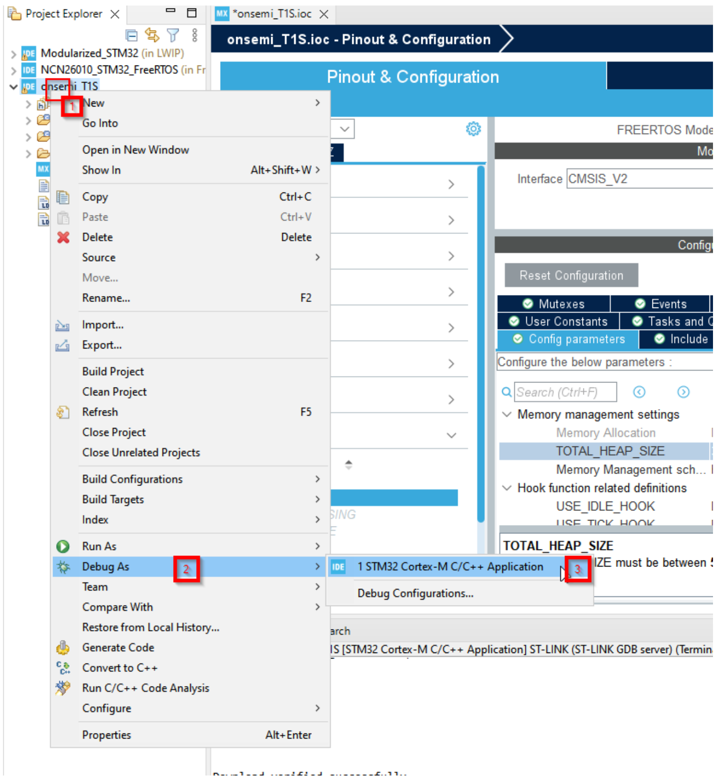


Flash and/or Debug the Example

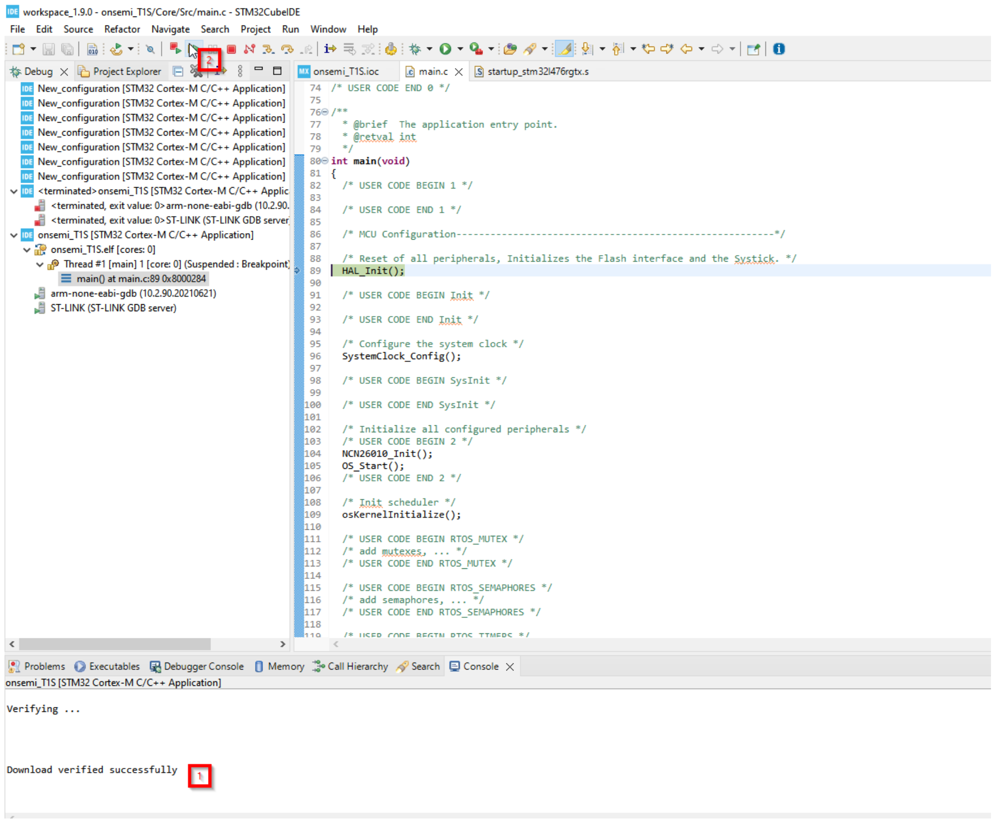
1. Save the Configuration and have the STM32CubeIDE generate code.


workspace_1.9.0 - Device Configuration Tool - STM32CubeIDE
File Edit Navigate Search Project Run Window Help
Project Explorer
Modularized_STM32 (in LWIP)
NCN26010_STM32_FreeRTOS (in Fr
onsemi_T1S
Includes
Core
Drivers
Debug
Middlewares
onsemi_T1S.ioc
onsemi_T1S.launch
STM32L476RGTX_FLASH.ld
STM32L476RGTX_RAM.ld
*onsemi_T1S.ioc
onsemi_T1S.ioc - Pinout & Configuration
Question
Do you want generate Code?
Remember my decision
2
Yes
No
Memory management se
TOTAL_HEAP_S
Memory Management sch... heap_4
Hook function related definitions
USE_IDLE_HOOK Disabled
TOTAL_HEAP_SIZE
TOTAL_HEAP_SIZE must be between 512 Bytes and 128 KBytes.

2. Ensure that your STM32L4xx evaluation board is connected to your computer, then Right click the project, then select debug as, then select STM32 Cortex-M C/C++ Application


Project Explorer
*onsemi_T1S.ioc
onsemi_T1S.ioc - Pinout & Configuration
Pinout & Configuration
Modularized_STM32 (in LWIP)
NCN26010_STM32_FreeRTOS (in Fr
onsemi_T1S
1
New
Go Into
Open in New Window
Show In Alt+Shift+W
Copy Ctrl+C
Paste Ctrl+V
Delete Delete
Source
Move...
Rename... F2
Import...
Export...
Build Project
Clean Project
Refresh F5
Close Project
Close Unrelated Projects
Build Configurations
Build Targets
Index
Run As
Debug As
2
Team
Compare With
Restore from Local History...
Generate Code
Convert to C++
Run C/C++ Code Analysis
Configure
Properties Alt+Enter
1 STM32 Cortex-M C/C++ Application
3
Debug Configurations...
FREERTOS Mode
Interface CMSIS_V2
Config
Reset Configuration
Mutexes
Events
User Constants
Tasks and Q
Config parameters
Include
Configure the below parameters :
Search (Ctrl+F)
Memory management settings
Memory Allocation
TOTAL_HEAP_SIZE
Memory Management sch...
Hook function related definitions
USE_IDLE_HOOK
USE_TICK_HOOK
TOTAL_HEAP_SIZE
IZE must be between
arch
S [STM32 Cortex-M C/C++ Application] ST-LINK (ST-LINK GDB server) (Termin
Download verified successfully

3. Wait for the build to complete and the hex to download to the board, then click play.



workspace_1.9.0 - onsemi_T1S/Core/Src/main.c - STM32CubeIDE

File Edit Source Refactor Navigate Search Project Run Window Help
2
Debug
Project Explorer
onsemi_T1S.ioc
main.c
startup_stm32l476rgtx.s

```
New_configuration [STM32 Cortex-M C/C++ Application]
New_configuration [STM32 Cortex-M C/C++ Application]
New_configuration [STM32 Cortex-M C/C++ Application]
New_configuration [STM32 Cortex-M C/C++ Application]
New_configuration [STM32 Cortex-M C/C++ Application]
New_configuration [STM32 Cortex-M C/C++ Application]
New_configuration [STM32 Cortex-M C/C++ Application]
<terminated> onsemi_T1S [STM32 Cortex-M C/C++ Applic
    <terminated, exit value: 0> arm-none-eabi-gdb (10.2.90.
    <terminated, exit value: 0> ST-LINK (ST-LINK GDB server
onsemi_T1S [STM32 Cortex-M C/C++ Application]
    onsemi_T1S.elf [cores: 0]
        Thread #1 [main] 1 [core: 0] (Suspended : Breakpoint)
            main() at main.c:89 0x8000284
    arm-none-eabi-gdb (10.2.90.20210621)
    ST-LINK (ST-LINK GDB server)
```

```c
/* USER CODE END 0 */

/**
  * @brief  The application entry point.
  * @retval int
  */
int main(void)
{
  /* USER CODE BEGIN 1 */

  /* USER CODE END 1 */

  /* MCU Configuration--------------------------------------------------------*/

  /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
  HAL_Init();

  /* USER CODE BEGIN Init */

  /* USER CODE END Init */

  /* Configure the system clock */
  SystemClock_Config();

  /* USER CODE BEGIN SysInit */

  /* USER CODE END SysInit */

  /* Initialize all configured peripherals */
  /* USER CODE BEGIN 2 */
  NCN26010_Init();
  OS_Start();
  /* USER CODE END 2 */

  /* Init scheduler */
  osKernelInitialize();

  /* USER CODE BEGIN RTOS_MUTEX */
  /* add mutexes, ... */
  /* USER CODE END RTOS_MUTEX */

  /* USER CODE BEGIN RTOS_SEMAPHORES */
  /* add semaphores, ... */
  /* USER CODE END RTOS_SEMAPHORES */

  /* USER CODE BEGIN RTOS_TIMERS */
```

Problems Executables Debugger Console Memory Call Hierarchy Search Console

onsemi_T1S [STM32 Cortex-M C/C++ Application]

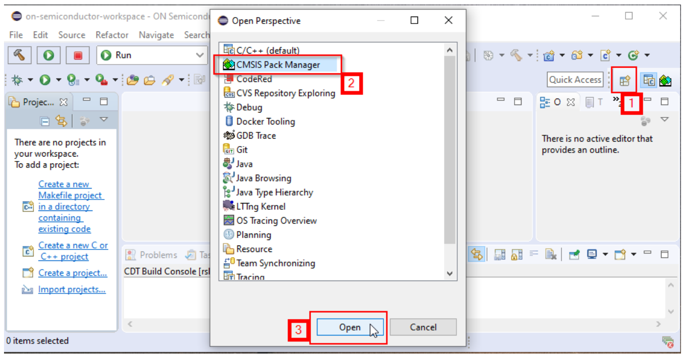
```
Verifying ...


Download verified successfully
```

RSL10

Install Toolchain for Firmware Development

1. Register and Log in to **MyON** account here.
2. Download and install “**ON Semiconductor IDE**”.
3. Download and install base “**RSL10 CMSIS Pack**”.
   a. To install the CMSIS pack open CMSIS Pack Manager. Click on the icon “Open Perspective” as shown below and select “CMSIS Pack Manger”.



There are no projects in your workspace. To add a project:

There is no active editor that provides an outline.

Figure 4. Open CMSIS Pack Manager

   b. Click on the icon “Import Existing Packs…” to import the CMSIS pack downloaded earlier. Select “Open” to install the pack.

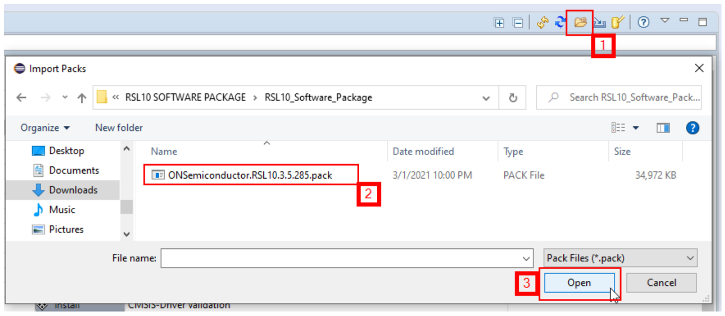
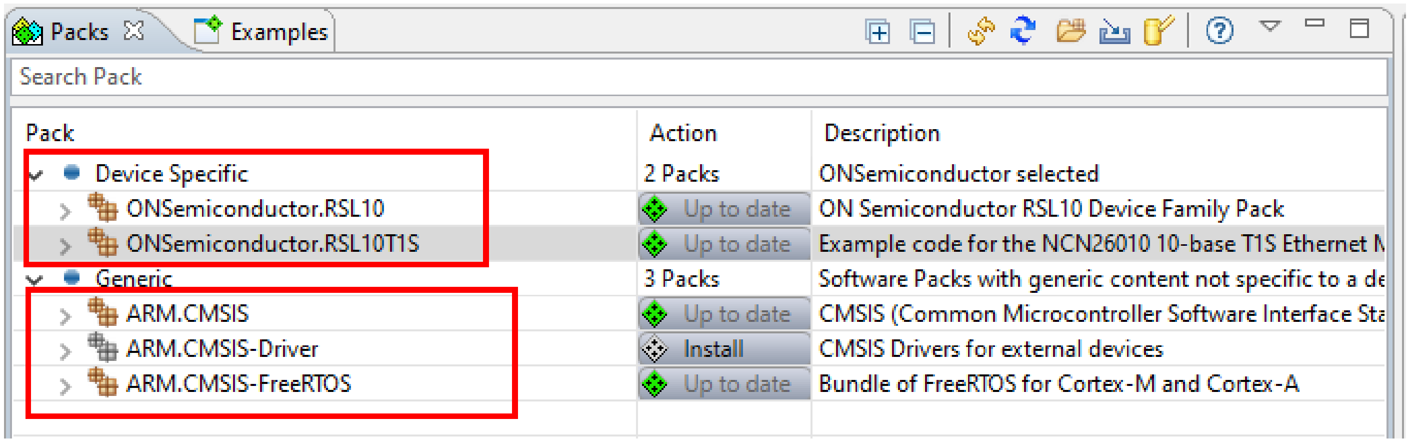


Figure 5. Import Existing CMSIS Pack

4. Download and install latest version of “**ARM CMSIS Pack**” as per step 3.
5. Download and install latest version of “**ARM CMSIS-FreeRTOS Pack**” as per step 3.
6. Download and install latest version of “**onsemi.RSL10T1S Pack**”as per step 3.
7. Once all the packs are installed, this is what CMSIS pack perspective should look like.



Packs | Examples

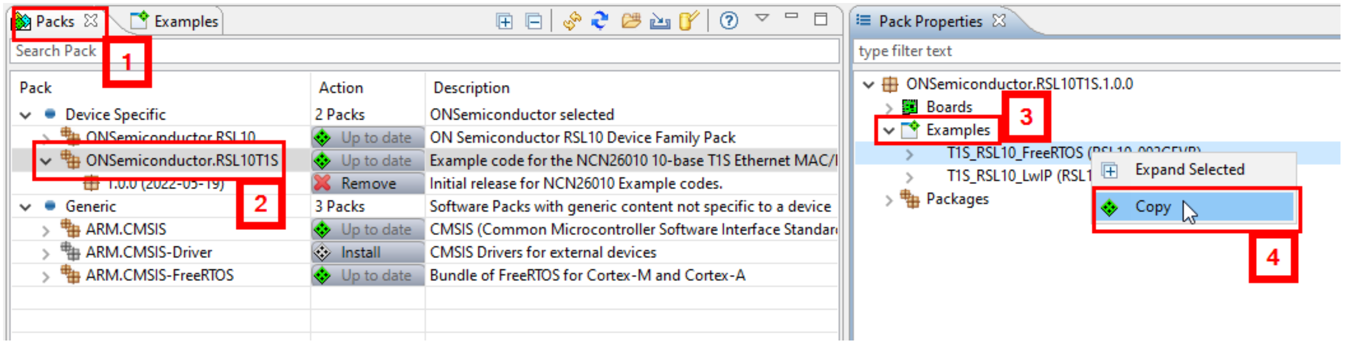
Search Pack

Pack	Action	Description
Device Specific	2 Packs	ONSemiconductor selected
ONSemiconductor.RSL10	Up to date	ON Semiconductor RSL10 Device Family Pack
ONSemiconductor.RSL10T1S	Up to date	Example code for the NCN26010 10-base T1S Ethernet M
Generic	3 Packs	Software Packs with generic content not specific to a de
ARM.CMSIS	Up to date	CMSIS (Common Microcontroller Software Interface Sta
ARM.CMSIS-Driver	Install	CMSIS Drivers for external devices
ARM.CMSIS-FreeRTOS	Up to date	Bundle of FreeRTOS for Cortex-M and Cortex-A

Figure 6. Installed CMSIS Packs

Import and Compile an Example Project from CMSIS Pack

1. Select RSL10T1S pack from CMSIS pack manager. Open the drop-down for “Examples”. Right-click on the preferred example project. Select “Copy”. This will import a copy of that example project into the workspace.



Packs | Examples

Search Pack

Pack	Action	Description
Device Specific	2 Packs	ONSemiconductor selected
ONSemiconductor.RSL10	Up to date	ON Semiconductor RSL10 Device Family Pack
ONSemiconductor.RSL10T1S	Up to date	Example code for the NCN26010 10-base T1S Ethernet MAC/I
1.0.0 (2022-05-19)	Remove	Initial release for NCN26010 Example codes.
Generic	3 Packs	Software Packs with generic content not specific to a device
ARM.CMSIS	Up to date	CMSIS (Common Microcontroller Software Interface Standar
ARM.CMSIS-Driver	Install	CMSIS Drivers for external devices
ARM.CMSIS-FreeRTOS	Up to date	Bundle of FreeRTOS for Cortex-M and Cortex-A

Pack Properties

type filter text

- ONSemiconductor.RSL10T1S.1.0.0
  - Boards
  - Examples
    - T1S_RSL10_FreeRTOS
    - T1S_RSL10_LwIP (RSL1
  - Packages

Figure 7. Copy Example Project

2. Right-click on the project name in Project Explorer and select “Build Configurations” to “Release” or “Debug”. Next, select “Build Project” to compile the project.

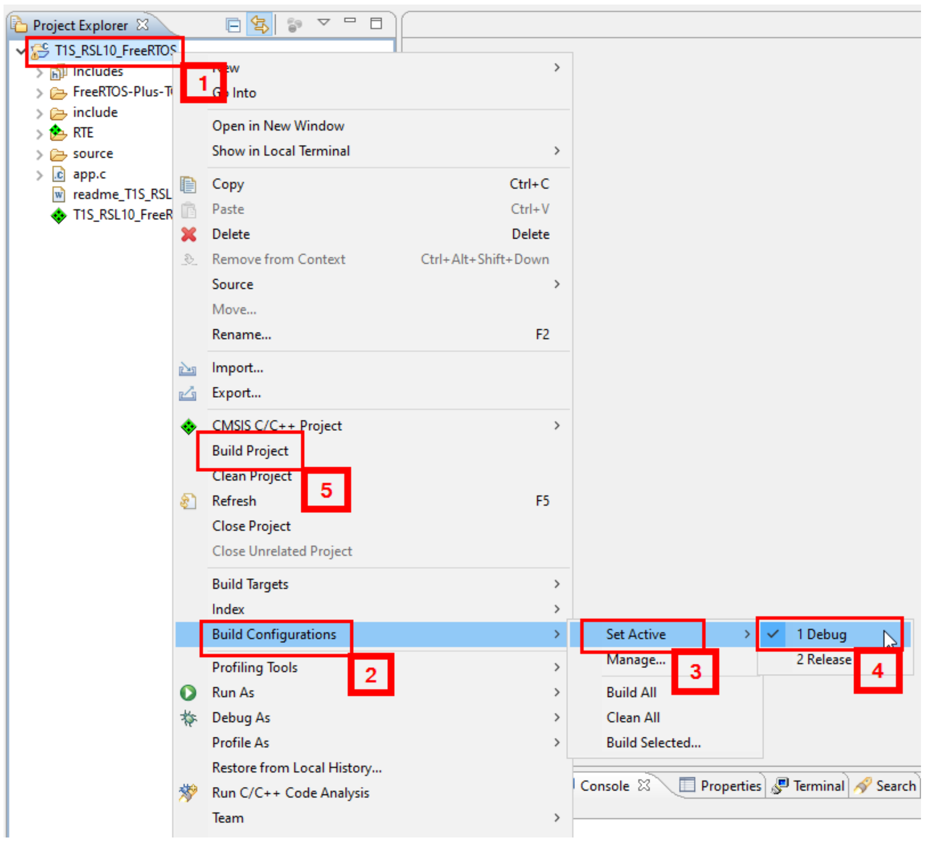


Figure 8. Set Build Configuration for a Project

Flash and/or Debug the Example

1. Create Run/Debug configuration for the project. Select “Run” -> “Run Configurations…”.

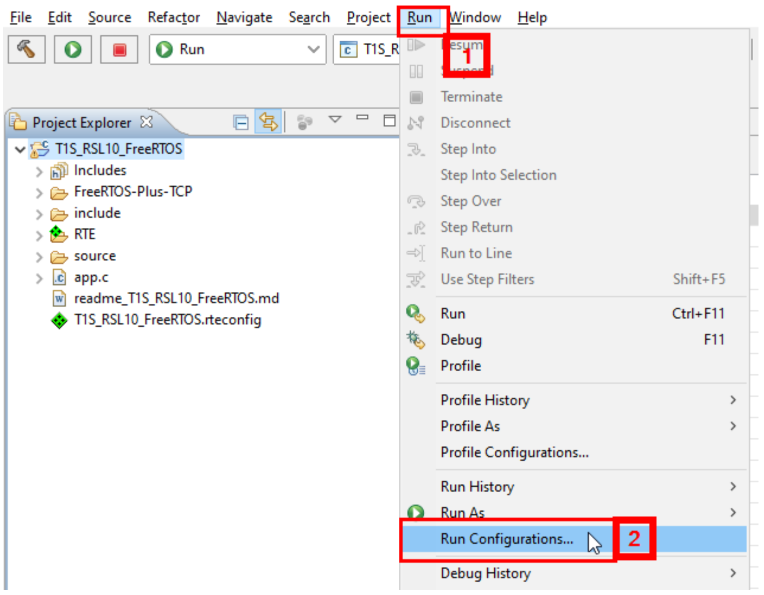
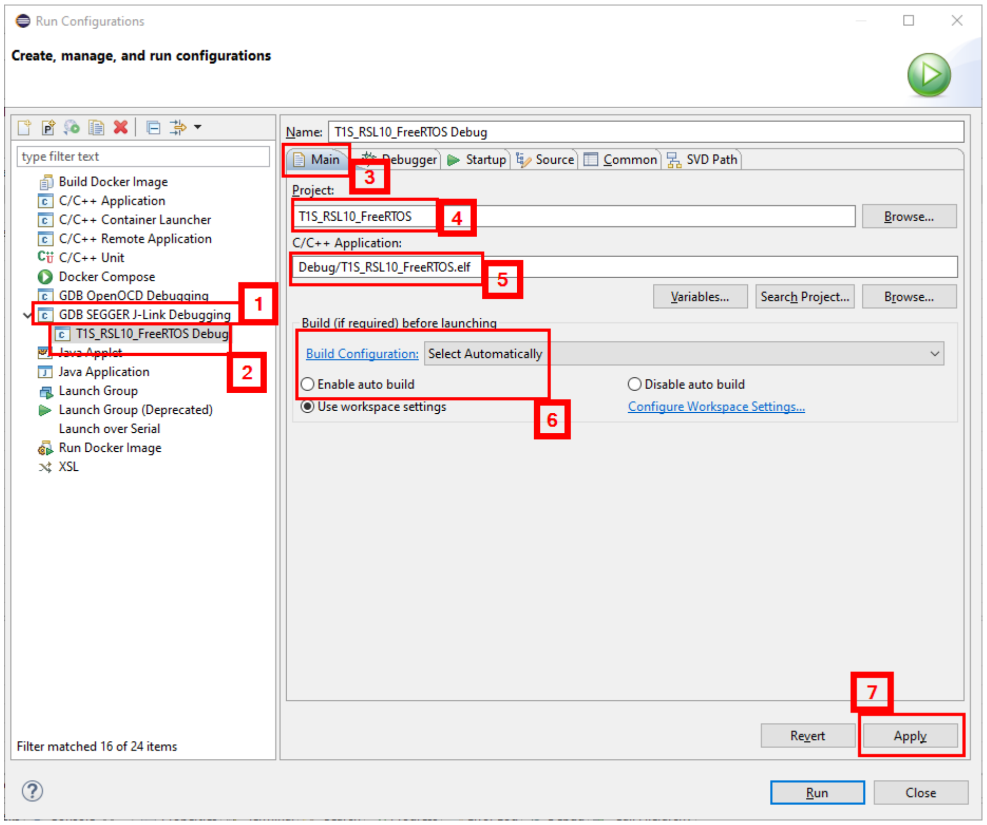


Figure 9. Open Run Configurations

2. Double-click on “GDB SEGGER J-Link Debugging”. This will generate a default configuration. Under “Main” tab, select the Project. Provide the .elf file path at “C/C++ Application”. Depending on the build

configuration it can be “Release/<project_name.elf>” or “Debug/<project_name.elf>”. Or add a variable to select automatically, “${config_name:${project_name}}/<project_name.elf>”. Select “Apply”.



Name: T1S_RSL10_FreeRTOS Debug

Project: T1S_RSL10_FreeRTOS

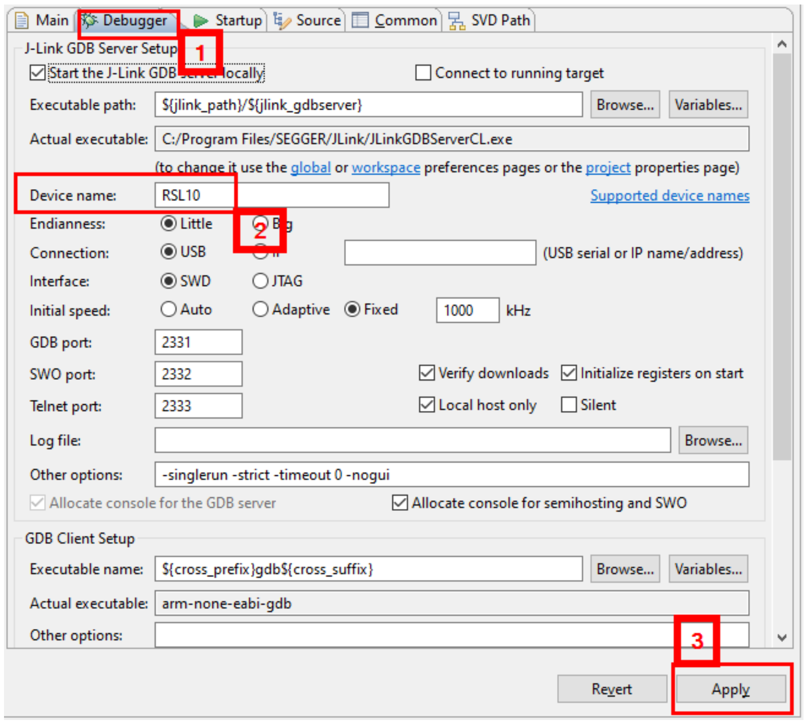
C/C++ Application: Debug/T1S_RSL10_FreeRTOS.elf

Build Configuration: Select Automatically

Enable auto build [✓ Use workspace settings] Disable auto build

Figure 10. Create Run Configuration

3. Under “Debugger” tab, Update “Device Name” to RSL10. Select “Apply” and “Close”.



J-Link GDB Server Setup

[✓] Start the J-Link GDB server locally [ ] Connect to running target

Executable path: ${jlink_path}/${jlink_gdbserver}

Actual executable: C:/Program Files/SEGGER/JLink/JLinkGDBServerCL.exe

(to change it use the global or workspace preferences pages or the project properties page)

Device name: RSL10

Endianness: [✓ Little] Big

Connection: [✓ USB] IP (USB serial or IP name/address)

Interface: [✓ SWD] JTAG

Initial speed: Auto Adaptive [✓ Fixed] 1000 kHz

GDB port: 2331

SWO port: 2332 [✓] Verify downloads [✓] Initialize registers on start

Telnet port: 2333 [✓] Local host only [ ] Silent

Log file:

Other options: -singlerun -strict -timeout 0 -nogui

[✓] Allocate console for the GDB server [✓] Allocate console for semihosting and SWO

GDB Client Setup

Executable name: ${cross_prefix}gdb${cross_suffix}

Actual executable: arm-none-eabi-gdb

Other options:

Figure 11. Add Device name

4. To flash the board, click green play button looking icon in "Run" mode.

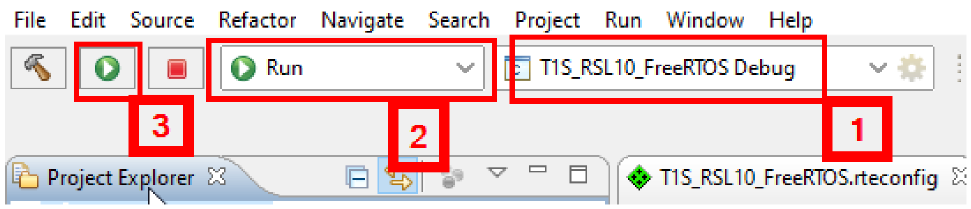


Figure 12. Run the configuration to Flash

5. To debug, click green bug looking icon in "Debug" mode.

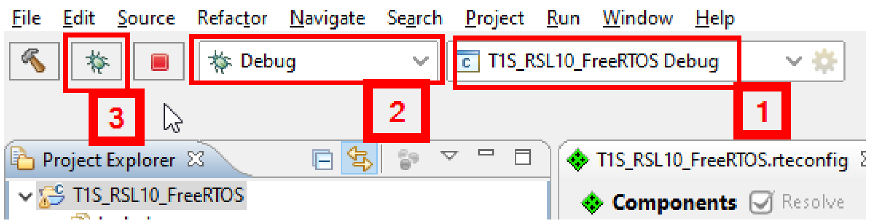


Figure 13. Start Debug configuration

6. Detailed instructions about using the IDE can be found in RSL10 Getting Started Guide.

Unimplemented Features

This example software is provided as a basic example to support designers to use the NCN26010 in their system. There are several optimizations and additional features that onsemi has identified as potential improvements to this software. onsemi may implement some of these features and will use feedback from customers to determine how to prioritize features.

Zero Copy Data Transfer

As currently implemented, the example implementations copy data to and from buffers when transferring from the NCN26010 over the SPI. With some pointer manipulation we could eliminate these copies which saves memory and eliminates the CPU time required to copy the data. Implementing this optimization may reduce system power requirements and may allow for a less expensive MCU to be used.

Runtime Configuration

As currently implemented, all the configuration options are selected at compile time by adjusting macro constants in the ncn26010.h header file. If the configuration needs to be changed after compile time this must be done using the remote configuration options described in ncn26010.h. An option to read a file or some piece of memory to decide how to configure the NCN26010 when initializing would simplify software deployment on multiple nodes.

Serial Communication for Command Line Interface

As currently implemented, there are empty macros defined for printing debug and other information to a console. Filling in these macros with functions that send this console information over a serial interface to a host computer would help simplify debugging. It would also allow for a command line interface to configure and communicate with the MCU.

Software Timestamping

The NCN26010 does not provide timestamps for incoming frames. Certain time sensitive messages require an accurate understanding of when the message arrived. The NCN26010 can be configured to signal on its DIO pins when it detects a start of frame delimiter. Triggering an interrupt on this signal and recording a timestamp would allow a software algorithm to pair together messages with their associated timestamp.

Documentation

The software package is documented in the source using Doxygen style comments. The generated documentation is distributed as part of the software package.