AN1389: Running Zigbee Host Applications in a Docker Container

The updated application structure in Zigbee EmberZNet 7.0 and higher no longer supports compiling host applications in MinGW for Windows. This document offers an alternative solution by using a Docker container to run the NCP Host Application.

KEY POINTS

- Example Configuration for NCP Host Application in Docker
- Host configuration solution for Windows users.

1 Introduction

Beginning with Zigbee EmberZNet 7.0, Silicon Labs introduced a new callback framework that automatically supplies *weak* callback functions to the stack, such as global-callback.c and zap-event.c. These *weak* functions ensure that the project builds when the callbacks are not explicitly implemented. This new framework replaces the 'callback-stub.c' file generated in EmberZNet 6.0 by AppBuilder. Overall, the new callback structure simplifies the user experience in application development but has been seen to cause errors in the linking stage in environments that do not support *weak* functions. More information on the new EmberZNet 7.0 callback framework can be found in *UG491: Zigbee Application Framework Developer's Guide for SDK 7.x and Higher.*

Specifically, the *weak* functions in EmberZNet 7.0 cause issues for Windows users when running applications in MinGW on Cygwin. MinGW treats *weak* function as "Null", causing undefined reference errors in the linking stages of the project build. This application note offers an alternative solution for Windows users that uses Docker to configure the host application. This solution is not required for Linux and MacOS operating systems, as Clang and GCC support *weak* functions. However, if you want to follow a similar procedure for Linux and MacOS, configuring a Docker host application will differ slightly from the procedure below. For more information contact Silicon Labs support through the support portal https://www.silabs.com/support.

This application note will walk through the configuration of the host application in a Docker container on a local Windows PC connected to an NCP-configured radio board via USB. To follow the procedure, you should have both Docker Desktop and Silink installed on your local PC. Silink is a Simplicity Studio tool that maps USB Ports to IP addresses, which allows Docker Containers running on Windows PCs to connect to the radio board.

Currently, this configuration does not allow Network Analyzer's network capture to run while the host and NCP applications are running. This is due to conflicts with Silink's port mapping. It is recommended to configure another radio board as a sniffer node and complete the network capture that way.

Requirements:

Docker (Version 20.10.0 or Higher) installed from https://www.docker.com/products/docker-desktop. This will install all dependencies needed for this procedure.

Simplicity Studio 5 installed from https://www.silabs.com/developers/simplicity-studio

Silink installed as part of Simplicity Studio 5 adapter pack tools (see section 2.3 Step 3: Install and Config Silink for instructions)

EmberZNet SDK 7.x or higher, installed as part of the Gecko SDK Suite (GSDK). See the Simplicity Studio v5 User's Guide for more information about installing the GSDK.

Radio Board installed on a development mainboard (BRD4162A is used in this example)

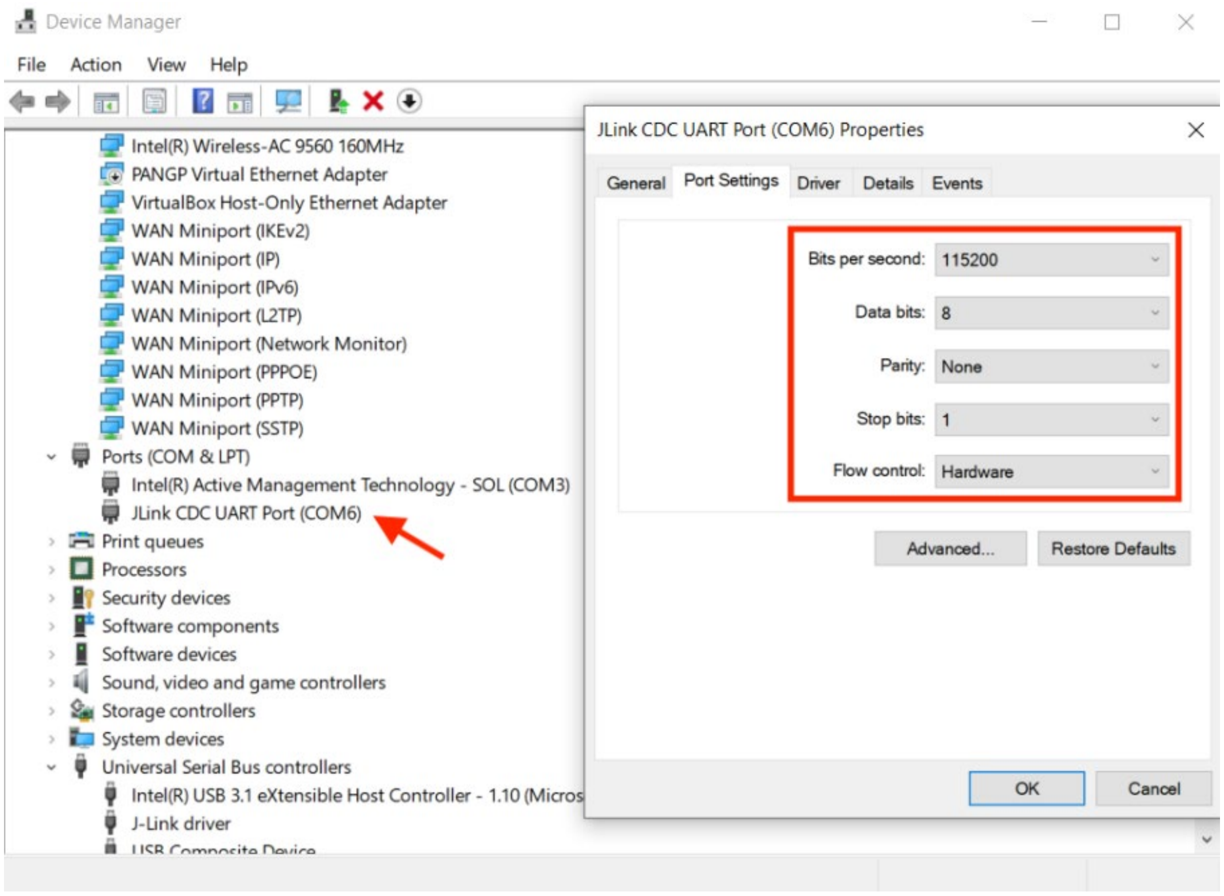
2 Running a Host Application in a Docker Container

2.1 Step 1: Create and Flash NCP-UART-HW Example Project onto BRD4162A

Create a new Simplicity Studio project for the target radio board (BRD4162A). Select the **Zigbee - NCP UartHw** example project. Build and flash the project onto BRD4162A. This will be the NCP-configured device. For more information on how to create projects in Simplicity Studio, refer to *QSG180: Zigbee EmberZNet Quick-Start Guide for SDK v7.0 and Higher*.

2.2 Step 2: Config COM Port

Make sure the radio board (BRD4162A) is connected to the host PC via USB. Open the Windows Device Manager and configure the "Port Settings" to match those used by the NCP application (115200 Baud rate with hardware flow control), as shown in the following figure.

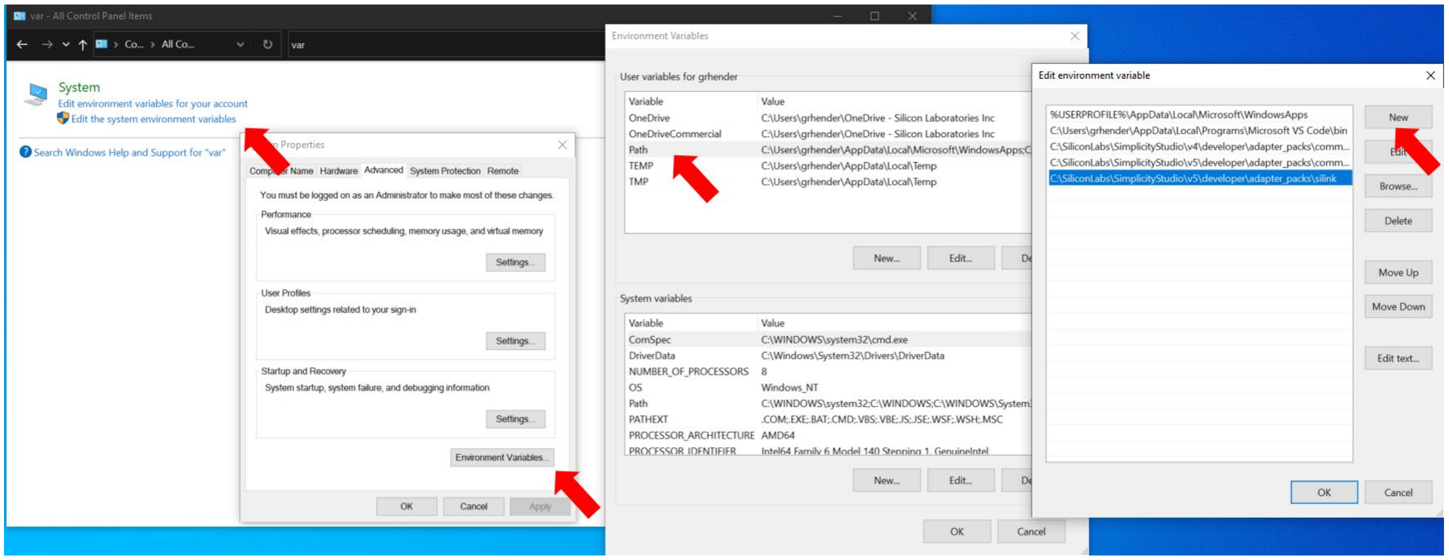


Bits per second: 115200
Data bits: 8
Parity: None
Stop bits: 1
Flow control: Hardware

2.3 Step 3: Install and Config Silink

To find Silink's installation path, navigate through the installation path of Simplicity Studios until you reach the adapter pack directory. If you used the default installation path, the file path should look like: C:/SiliconLabs/SimplicityStudio/v5/developer/adapter_packs/Silink.

Add the Silink file path to the Windows "PATH" environment variables. The following figure shows the steps required.



You must be logged on as an Administrator to make most of these changes.

Visual effects, processor scheduling, memory usage, and virtual memory

Desktop settings related to your sign-in

System startup, system failure, and debugging information

User variables for grhender

Variable	Value
OneDrive	C:\Users\grhender\OneDrive - Silicon Laboratories Inc
OneDriveCommercial	C:\Users\grhender\OneDrive - Silicon Laboratories Inc
Path	C:\Users\grhender\AppData\Local\Microsoft\WindowsApps;C
TEMP	C:\Users\grhender\AppData\Local\Temp
TMP	C:\Users\grhender\AppData\Local\Temp

System variables

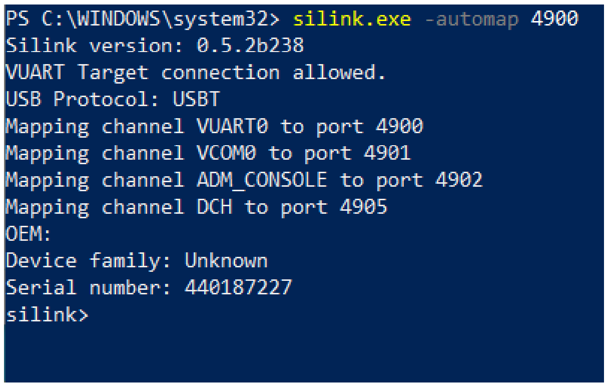
Variable	Value
ComSpec	C:\WINDOWS\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PROCESSORS	8
OS	Windows_NT
Path	C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE	AMD64
PROCESSOR_IDENTIFIER	Intel64 Family 6 Model 140 Stepping 1, GenuineIntel

```
%USERPROFILE%\AppData\Local\Microsoft\WindowsApps
C:\Users\grhender\AppData\Local\Programs\Microsoft VS Code\bin
C:\SiliconLabs\SimplicityStudio\v4\developer\adapter_packs\comm...
C:\SiliconLabs\SimplicityStudio\v5\developer\adapter_packs\comm...
C:\SiliconLabs\SimplicityStudio\v5\developer\adapter_packs\silink
```

Run "Windows PowerShell" as Administrator and execute the following commands to map the USB ports to IP address ports.

```
silink.exe -automap 4900
```

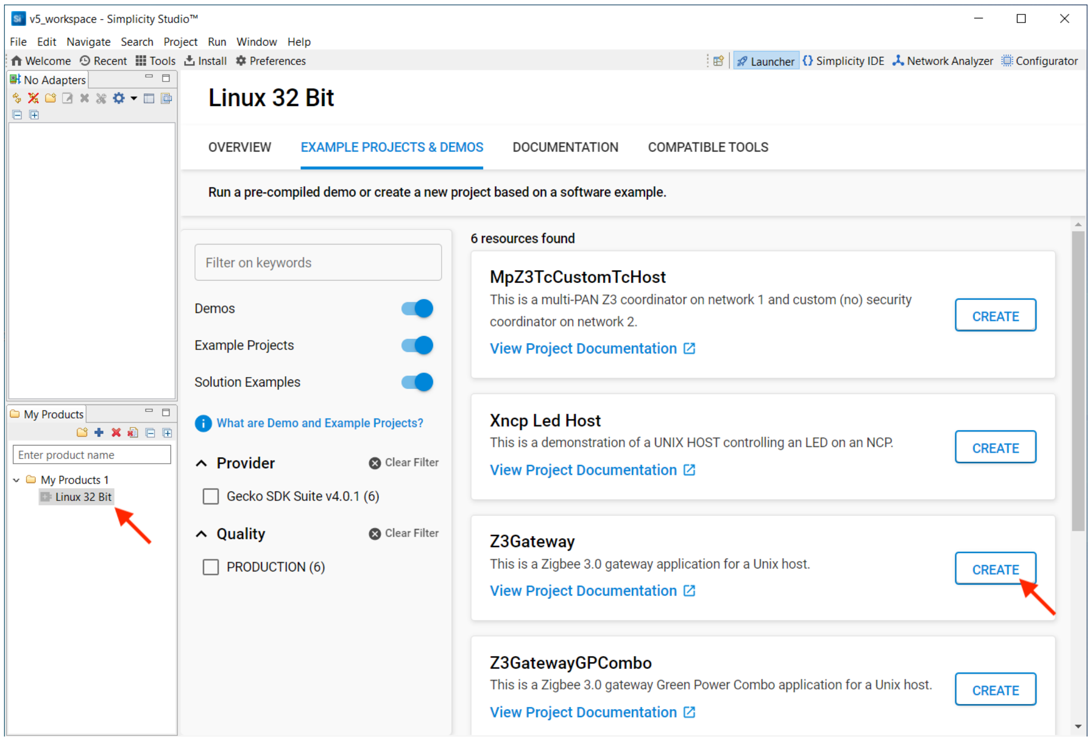
The following figure shows the desired output from the console.



```
PS C:\WINDOWS\system32> silink.exe -automap 4900
Silink version: 0.5.2b238
VUART Target connection allowed.
USB Protocol: USBT
Mapping channel VUART0 to port 4900
Mapping channel VCOM0 to port 4901
Mapping channel ADM_CONSOLE to port 4902
Mapping channel DCH to port 4905
OEM:
Device family: Unknown
Serial number: 440187227
silink>
```

2.4 Step 4: Create a Host App for Linux 32 Bit

In the Simplicity Studio My Products view, select **Linux 32 Bit** as the target part and select the Zigbee - Z3Gateway project, as shown in the following figure.


v5_workspace - Simplicity Studio™
File Edit Navigate Search Project Run Window Help
Welcome Recent Tools Install Preferences
Launcher Simplicity IDE Network Analyzer Configurator
No Adapters
My Products
Enter product name
My Products 1
Linux 32 Bit
OVERVIEW
EXAMPLE PROJECTS & DEMOS
DOCUMENTATION
COMPATIBLE TOOLS
Filter on keywords
Demos
Example Projects
Solution Examples
What are Demo and Example Projects?
Provider
Clear Filter
Gecko SDK Suite v4.0.1 (6)
Quality
Clear Filter
PRODUCTION (6)
CREATE
View Project Documentation

Linux 32 Bit

Run a pre-compiled demo or create a new project based on a software example.

6 resources found

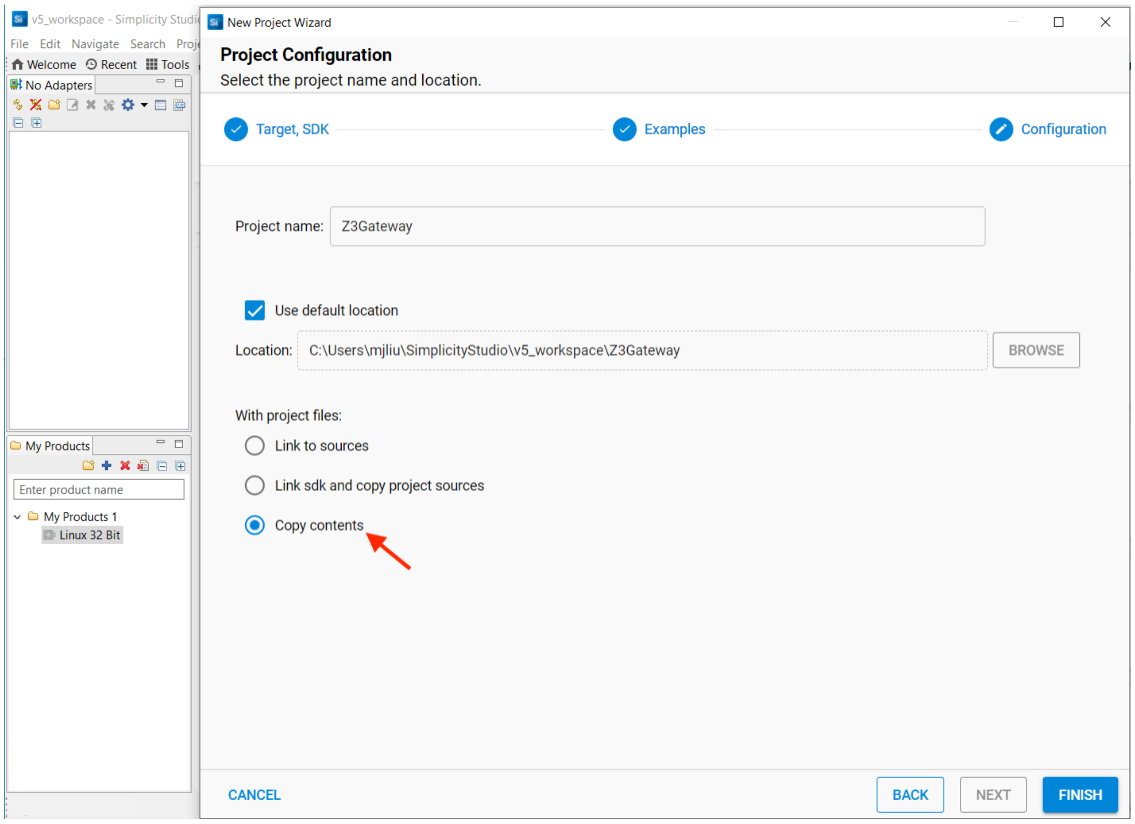
MpZ3TcCustomTcHost
This is a multi-PAN Z3 coordinator on network 1 and custom (no) security coordinator on network 2.

Xncp Led Host
This is a demonstration of a UNIX HOST controlling an LED on an NCP.

Z3Gateway
This is a Zigbee 3.0 gateway application for a Unix host.

Z3GatewayGPCombo
This is a Zigbee 3.0 gateway Green Power Combo application for a Unix host.

In the Project Configuration dialog, select **Copy Contents**, as shown in the following figure. This copies necessary files from the GSDK to the project folder, so the Docker Container does not need a complete GSDK.



Project Configuration
Select the project name and location.

Project name: Z3Gateway

[✓] Use default location

Location: C:\Users\mjliu\SimplicityStudio\v5_workspace\Z3Gateway

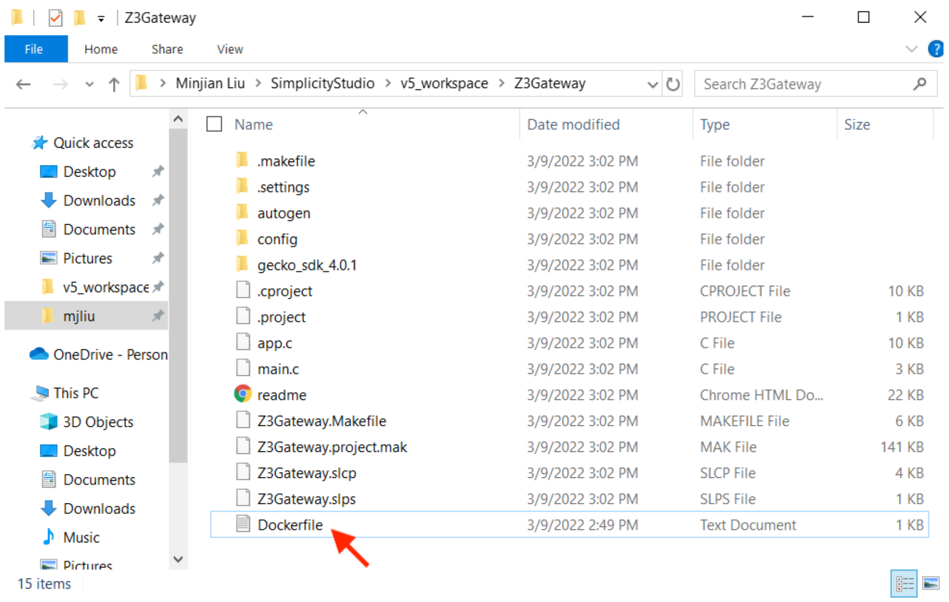
With project files:
Link to sources
Link sdk and copy project sources
[✓ Copy contents]

2.5 Step 5: Create a Dockerfile

Create a new file in the root project directory and name it "Dockerfile" without any file extension or suffix. Add the following commands:

```
FROM gcc
RUN apt-get update && apt-get -y install gcc-multilib socat
RUN echo "socat -d TCP:host.docker.internal:4901 pty,raw,echo=0,link=/dev/ttySilink &" >> ~/.bashrc
COPY . /usr/src/zigbee_z3_gateway
WORKDIR /usr/src/zigbee_z3_gateway
RUN make -f zigbee_z3_gateway.Makefile
```

This file configures the Docker container with a GCC environment and downloads the required packages to run the application. The following figure shows the Dockerfile location in the project structure of zigbee_z3_gateway.



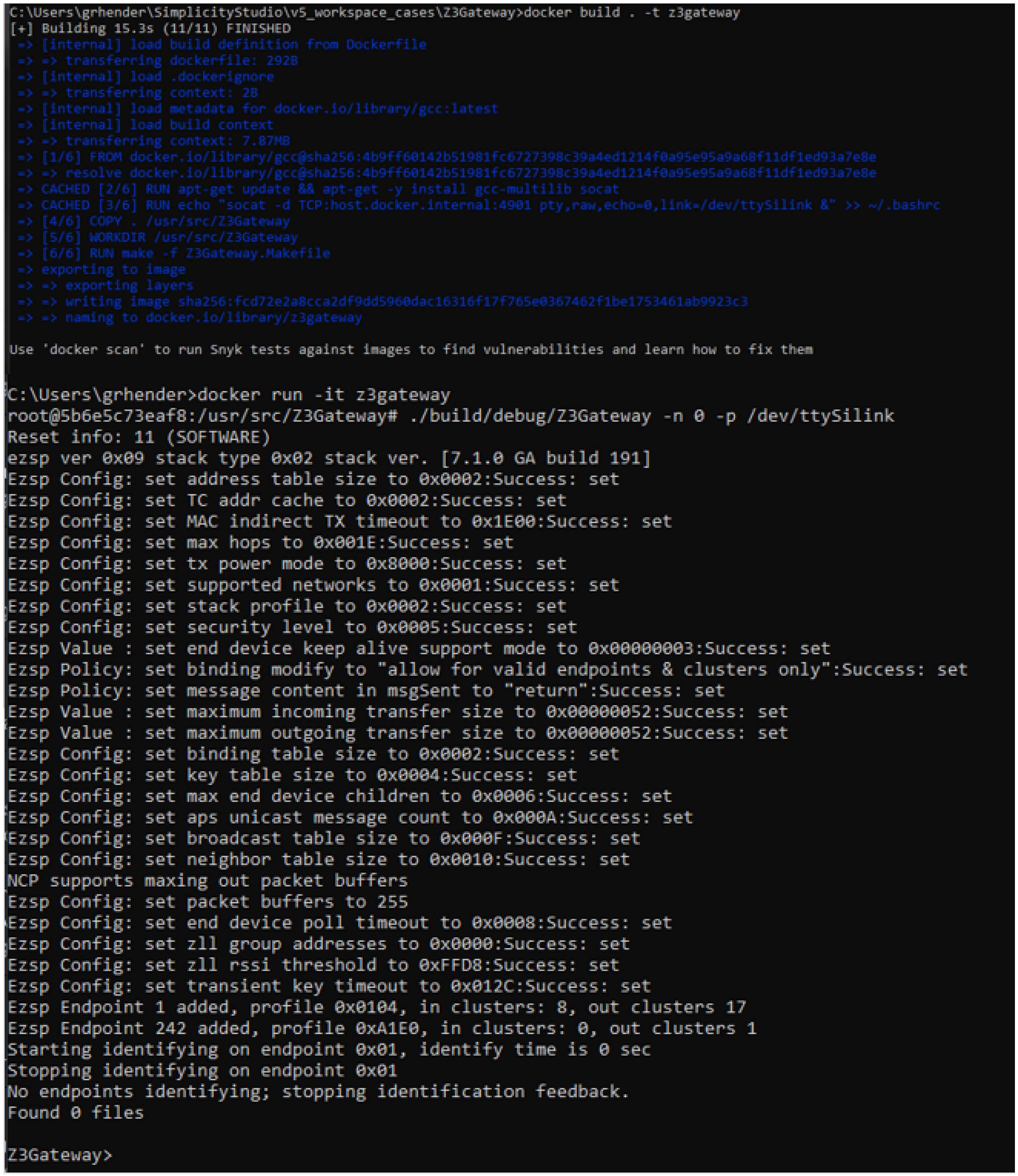
Name	Date modified	Type	Size
.makefile	3/9/2022 3:02 PM	File folder	
.settings	3/9/2022 3:02 PM	File folder	
autogen	3/9/2022 3:02 PM	File folder	
config	3/9/2022 3:02 PM	File folder	
gecko_sdk_4.0.1	3/9/2022 3:02 PM	File folder	
.cproject	3/9/2022 3:02 PM	CPROJECT File	10 KB
.project	3/9/2022 3:02 PM	PROJECT File	1 KB
app.c	3/9/2022 3:02 PM	C File	10 KB
main.c	3/9/2022 3:02 PM	C File	3 KB
readme	3/9/2022 3:02 PM	Chrome HTML Do...	22 KB
Z3Gateway.Makefile	3/9/2022 3:02 PM	MAKEFILE File	6 KB
Z3Gateway.project.mak	3/9/2022 3:02 PM	MAK File	141 KB
Z3Gateway.slcp	3/9/2022 3:02 PM	SLCP File	4 KB
Z3Gateway.slps	3/9/2022 3:02 PM	SLPS File	1 KB
Dockerfile	3/9/2022 2:49 PM	Text Document	1 KB

2.6 Step 6: Build and Run the Host App in the Container

Run the following commands on the console inside the project folder to build and start the container.

```
docker build . -t zigbee_z3_gateway
docker run -it z3gateway
./build/debug/zigbee_z3_gateway -n 0 -p /dev/ttySilink
```

The following figure shows the desired output of these commands.

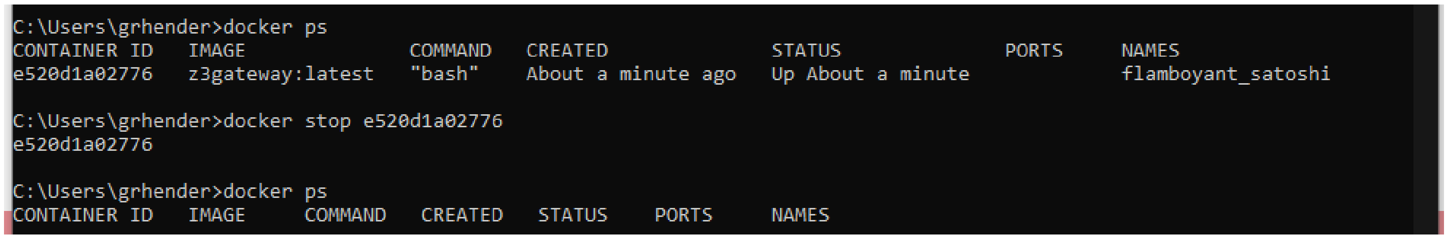


```
C:\Users\grhender\SimplicityStudio\v5_workspace_cases\Z3Gateway>docker build . -t z3gateway
[+] Building 15.3s (11/11) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 292B
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [internal] load metadata for docker.io/library/gcc:latest
 => [internal] load build context
 => => transferring context: 7.87MB
 => [1/6] FROM docker.io/library/gcc@sha256:4b9ff60142b51981fc6727398c39a4ed1214f0a95e95a9a68f11df1ed93a7e8e
 => => resolve docker.io/library/gcc@sha256:4b9ff60142b51981fc6727398c39a4ed1214f0a95e95a9a68f11df1ed93a7e8e
 => CACHED [2/6] RUN apt-get update && apt-get -y install gcc-multilib socat
 => CACHED [3/6] RUN echo "socat -d TCP:host.docker.internal:4901 pty,raw,echo=0,link=/dev/ttySilink &" >> ~/.bashrc
 => [4/6] COPY . /usr/src/Z3Gateway
 => [5/6] WORKDIR /usr/src/Z3Gateway
 => [6/6] RUN make -f Z3Gateway.Makefile
 => exporting to image
 => => exporting layers
 => => writing image sha256:fcd72e2a8cca2df9dd5960dac16316f17f765e0367462f1be1753461ab9923c3
 => => naming to docker.io/library/z3gateway

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them

C:\Users\grhender>docker run -it z3gateway
root@5b6e5c73eaf8:/usr/src/Z3Gateway# ./build/debug/Z3Gateway -n 0 -p /dev/ttySilink
Reset info: 11 (SOFTWARE)
ezsp ver 0x09 stack type 0x02 stack ver. [7.1.0 GA build 191]
Ezsp Config: set address table size to 0x0002:Success: set
Ezsp Config: set TC addr cache to 0x0002:Success: set
Ezsp Config: set MAC indirect TX timeout to 0x1E00:Success: set
Ezsp Config: set max hops to 0x001E:Success: set
Ezsp Config: set tx power mode to 0x8000:Success: set
Ezsp Config: set supported networks to 0x0001:Success: set
Ezsp Config: set stack profile to 0x0002:Success: set
Ezsp Config: set security level to 0x0005:Success: set
Ezsp Value : set end device keep alive support mode to 0x00000003:Success: set
Ezsp Policy: set binding modify to "allow for valid endpoints & clusters only":Success: set
Ezsp Policy: set message content in msgSent to "return":Success: set
Ezsp Value : set maximum incoming transfer size to 0x00000052:Success: set
Ezsp Value : set maximum outgoing transfer size to 0x00000052:Success: set
Ezsp Config: set binding table size to 0x0002:Success: set
Ezsp Config: set key table size to 0x0004:Success: set
Ezsp Config: set max end device children to 0x0006:Success: set
Ezsp Config: set aps unicast message count to 0x000A:Success: set
Ezsp Config: set broadcast table size to 0x000F:Success: set
Ezsp Config: set neighbor table size to 0x0010:Success: set
NCP supports maxing out packet buffers
Ezsp Config: set packet buffers to 255
Ezsp Config: set end device poll timeout to 0x0008:Success: set
Ezsp Config: set zll group addresses to 0x0000:Success: set
Ezsp Config: set zll rssi threshold to 0xFFD8:Success: set
Ezsp Config: set transient key timeout to 0x012C:Success: set
Ezsp Endpoint 1 added, profile 0x0104, in clusters: 8, out clusters 17
Ezsp Endpoint 242 added, profile 0xA1E0, in clusters: 0, out clusters 1
Starting identifying on endpoint 0x01, identify time is 0 sec
Stopping identifying on endpoint 0x01
No endpoints identifying; stopping identification feedback.
Found 0 files

Z3Gateway>
```

After this step, the host application is up and running.

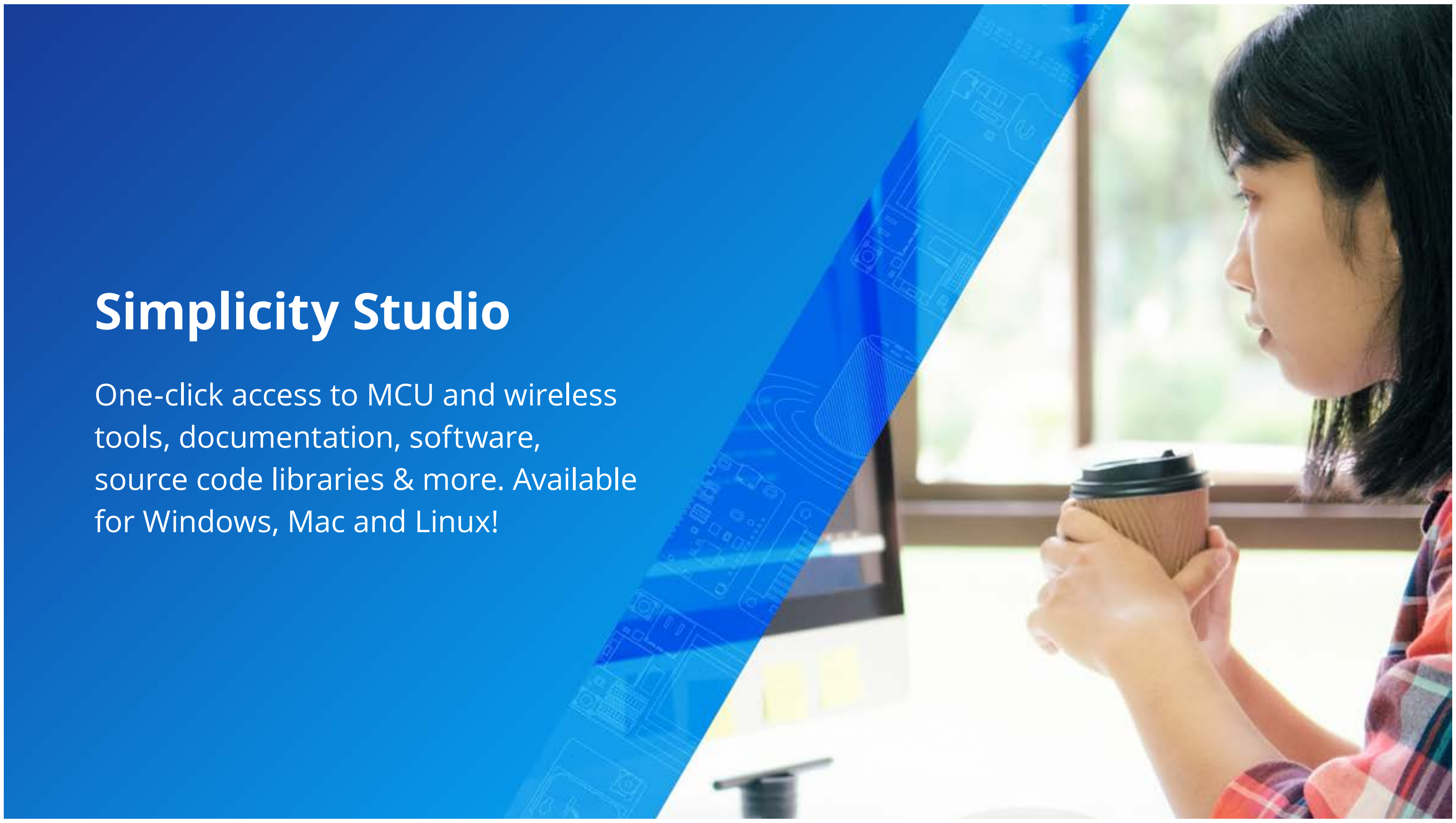
In order to stop the Docker container, enter the following commands into a new console. `Docker ps` lists all running containers on the local PC. `Docker stop` then stops whatever container is explicitly specified.

```
Docker ps
Docker stop <container_id>
```



```
C:\Users\grhender>docker ps
CONTAINER ID   IMAGE              COMMAND   CREATED              STATUS              PORTS     NAMES
e520d1a02776   z3gateway:latest   "bash"    About a minute ago   Up About a minute             flamboyant_satoshi

C:\Users\grhender>docker stop e520d1a02776
e520d1a02776

C:\Users\grhender>docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
```

Alternatively, the Docker desktop application can run, stop, and pause the Docker containers on the local machine.


Simplicity Studio
One-click access to MCU and wireless tools, documentation, software, source code libraries & more. Available for Windows, Mac and Linux!

IoT Portfolio	**SW/HW**	**Quality**	**Support & Community**
www.silabs.com/IoT	www.silabs.com/simplicity	www.silabs.com/quality	www.silabs.com/community

SILICON LABS

Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
USA

www.silabs.com