

# Renode Example Guide

## Introduction

This guide introduces how to emulate applications on Renode, the following platform is used

Renode platform: STM32F4\_discovery kit

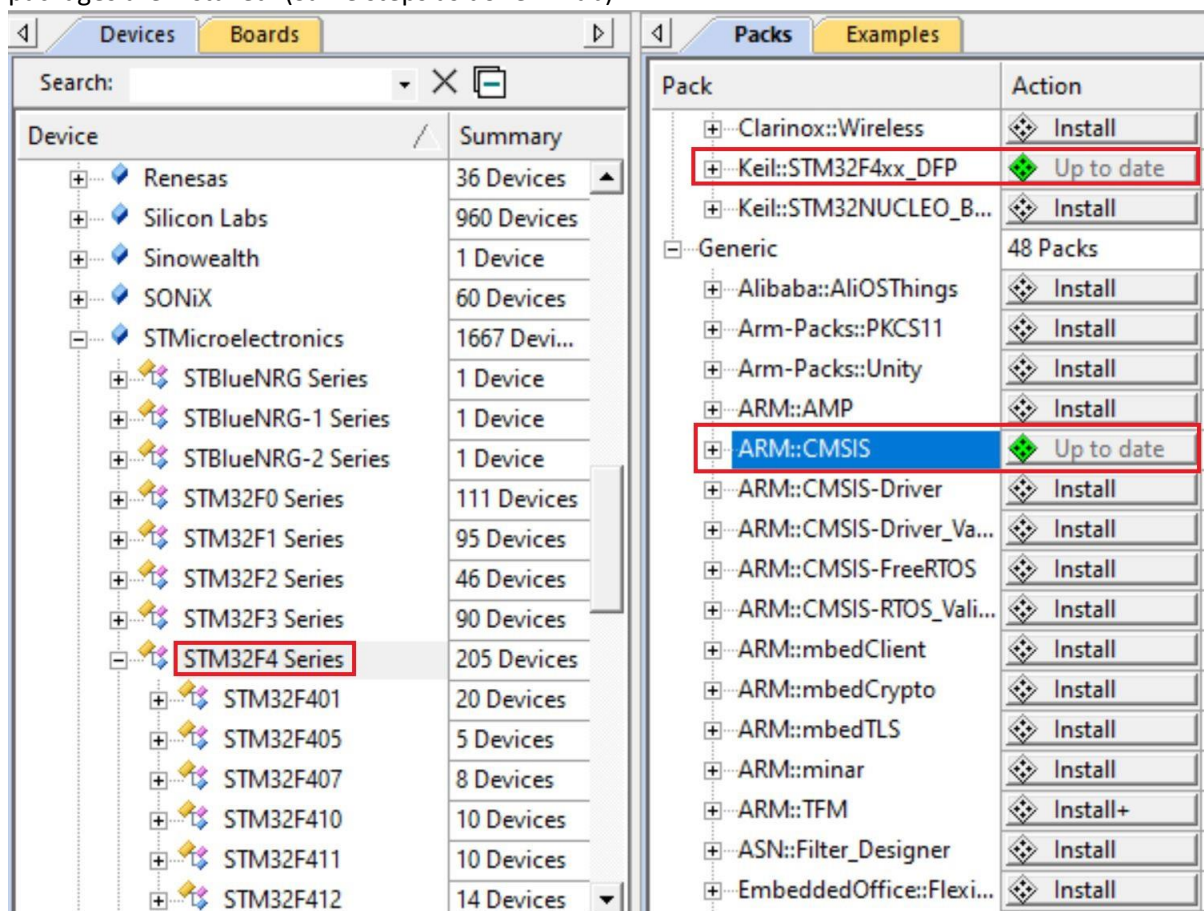
Microcontroller: STM32F407xx

CPU: Cortex® -M4 with FPU core

The guide starts with describing how to setup the development environment of the application on Keil. Then it describes how to run the application over Renode.

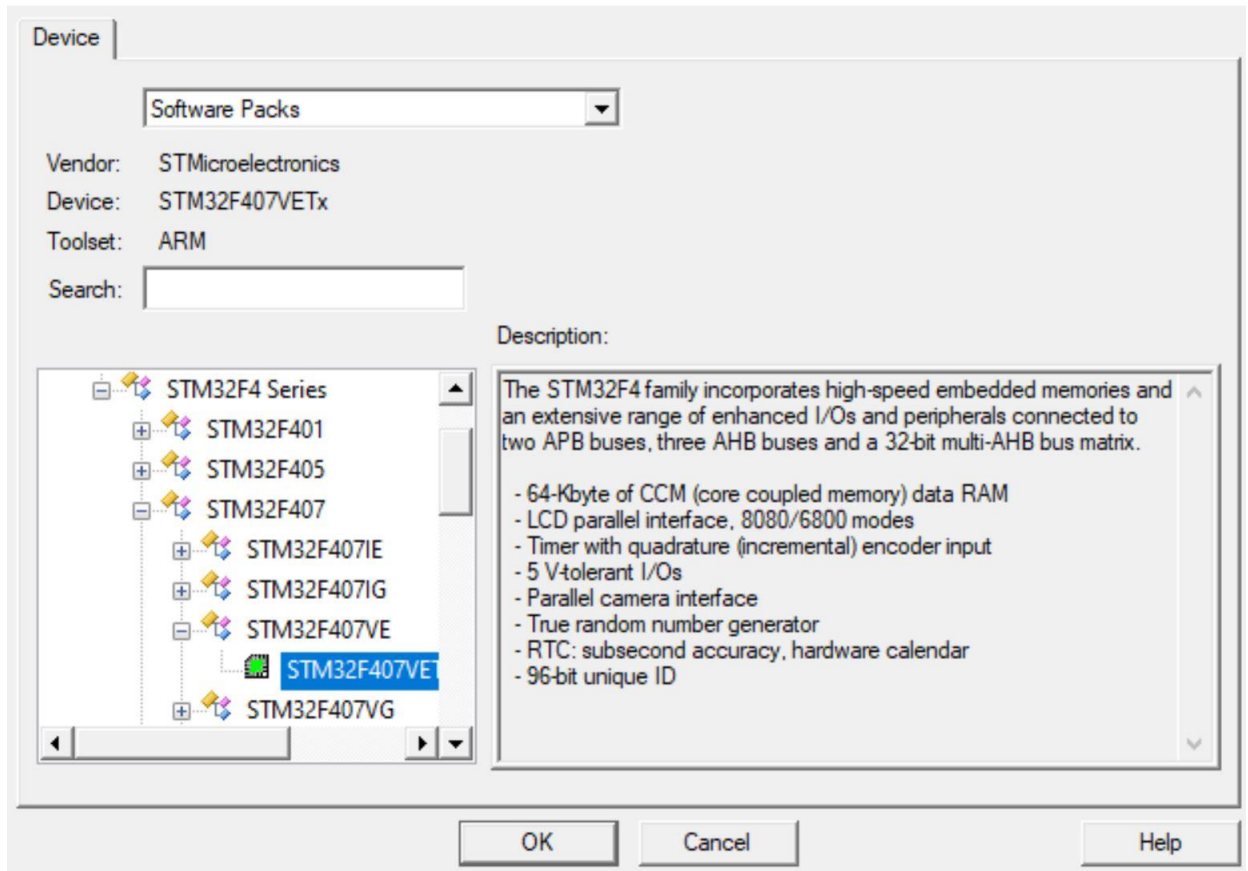
## Development Environment

1. Open Keil, then open package manager and make sure that STM32F4 series and CMSIS core packages are installed. (Same steps as done in Lab)

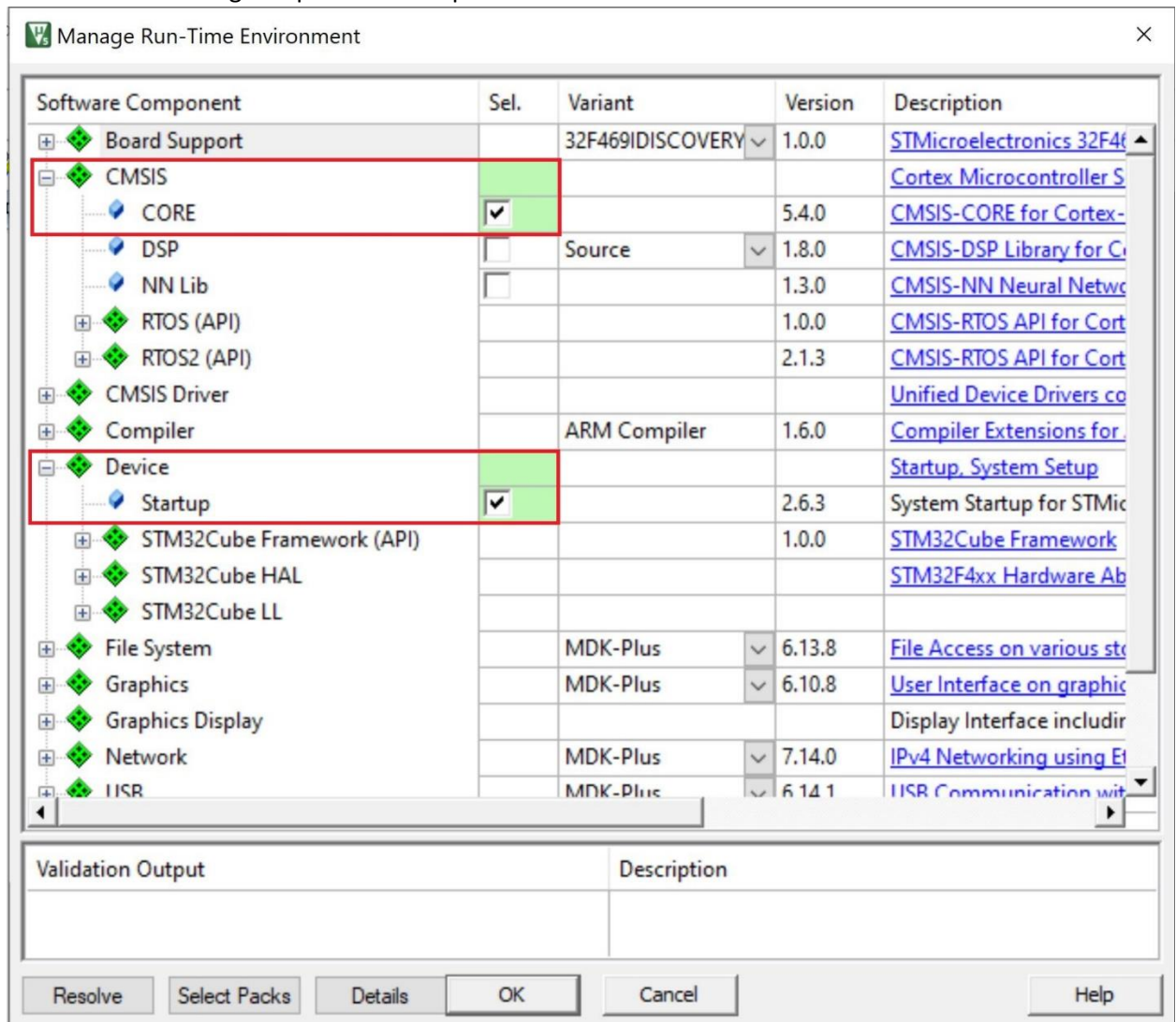


2. Create new project and select device as shown then press OK.

Select Device for Target 'Target 1'...



3. Choose the following components then press OK

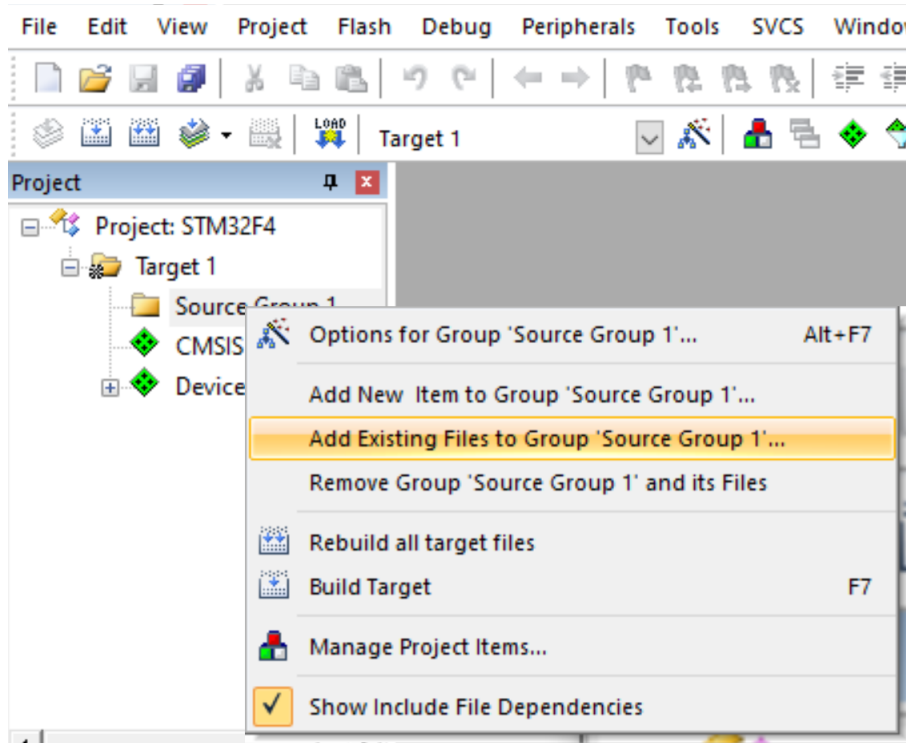
The image shows a 'Manage Run-Time Environment' dialog box. It features a tree view on the left with expandable categories like Board Support, CMSIS, RTOS, CMSIS Driver, Compiler, Device, STM32Cube Framework, File System, Graphics, Graphics Display, Network, and IISIR. The right side is a table with columns: Software Component, Sel., Variant, Version, and Description. Two rows are highlighted with red boxes: 'CMSIS' and 'Device'. In the 'Sel.' column, the 'CMSIS' row has a green square with a checkmark, and the 'Device' row has a green square with a checkmark. The 'Variant' column for 'CMSIS' is empty, while for 'Device' it is 'ARM Compiler'. The 'Version' column shows '5.4.0' for CMSIS and '2.6.3' for Device. The 'Description' column provides links to documentation for each component. At the bottom, there is a 'Validation Output' section with a 'Description' column, and a row of buttons: 'Resolve', 'Select Packs', 'Details', 'OK', 'Cancel', and 'Help'.

Software Component	Sel.	Variant	Version	Description
Board Support		32F469IDISCOVERY	1.0.0	<a href="#">STMicroelectronics 32F469</a>
CMSIS	<input checked="" type="checkbox"/>		5.4.0	<a href="#">Cortex Microcontroller S</a>
CORE	<input checked="" type="checkbox"/>		5.4.0	<a href="#">CMSIS-CORE for Cortex-</a>
DSP	<input type="checkbox"/>	Source	1.8.0	<a href="#">CMSIS-DSP Library for C</a>
NN Lib	<input type="checkbox"/>		1.3.0	<a href="#">CMSIS-NN Neural Netw</a>
RTOS (API)			1.0.0	<a href="#">CMSIS-RTOS API for Cort</a>
RTOS2 (API)			2.1.3	<a href="#">CMSIS-RTOS API for Cort</a>
CMSIS Driver				<a href="#">Unified Device Drivers co</a>
Compiler		ARM Compiler	1.6.0	<a href="#">Compiler Extensions for</a>
Device	<input checked="" type="checkbox"/>			<a href="#">Startup, System Setup</a>
Startup	<input checked="" type="checkbox"/>		2.6.3	System Startup for STMic
STM32Cube Framework (API)			1.0.0	<a href="#">STM32Cube Framework</a>
STM32Cube HAL				<a href="#">STM32F4xx Hardware Ab</a>
STM32Cube LL				
File System		MDK-Plus	6.13.8	<a href="#">File Access on various st</a>
Graphics		MDK-Plus	6.10.8	<a href="#">User Interface on graphic</a>
Graphics Display				Display Interface includir
Network		MDK-Plus	7.14.0	<a href="#">IPv4 Networking using Et</a>
IISIR		MDK-Plus	6.14.1	<a href="#">IISIR Communication wit</a>

Validation Output	Description

Resolve Select Packs Details OK Cancel Help

4. Choose to add existing item to project as below and add main.c file



5. Build target. No errors should be shown and application binary < Keil\_Project\_Name >.axf should be generated in < Keil\_Project\_Path >\Objects folder

Now the development environment is ready, next step is to run the application using Renode.

## Running Applications on Renode

1. After installing Renode, open single node scripts folder which exists in  
<Renode\_Installation\_Path>\scripts\single-node
2. Open stm32f4\_discovery.resc using any text editor
3. Change the value of \$bin to the path of the application binary as shown below (the path of the application binary is <Keil\_Project\_Path>\Objects\<Keil\_Project\_Name>.axf)
4. Make sure showAnalyzer command uses sysbus.uart2

```
:name: STM32F4_Discovery
:description: This script runs Contiki on STM32F4_Discovery.

using sysbus
$name?="STM32F4_Discovery"
mach create $name
machine LoadPlatformDescription @platforms/boards/stm32f4_discovery-kit.repl

cpu PerformanceInMips 125

$bin?=@C:/STM32F4/Objects/STM32F4.axf

showAnalyzer sysbus.uart2

### Set random board UNIQUE ID ###

python "import _random"
python "rand = _random.Random()"

$!d1=`python "print rand.getrandbits(32)"`
$!d2=`python "print rand.getrandbits(32)"`
$!d3=`python "print rand.getrandbits(32)"`
macro reset
"""
... sysbus LoadELF $bin

... sysbus WriteDoubleWord 0x1FFF7A10 $!d1
... sysbus WriteDoubleWord 0x1FFF7A14 $!d2
... sysbus WriteDoubleWord 0x1FFF7A18 $!d3
"""

runMacro $reset
```

5. Open Renode.exe

6. In Renode monitor shell execute stm32f4\_discovery script by typing the command  
**s @scripts/single-node/stm32f4\_discovery.resc**

