

AVT9152 EVB

Demo Software Guide

v1.1 – December 11, 2020

Document Control

Document Version: 1.1
Document Date: 11 Dec 2020
Document Author: Jennifer Sy
Document Classification: Public
Document Distribution: Public

Version History

Version	Date	Comment
1.0	04 Dec 2020	First Release
1.1	11 Dec 2020	Fix hyperlinks in chapter 3.2 and 4.

Contents

1	Introduction	4
2	Directory structure.....	4
3	Required external SDK and library	4
3.1	nRF Connect SDK.....	4
3.2	nRF9160 SUPL client library	5
4	nRF9160 modem firmware	6
5	demo_IoTConnect.....	6
5.1	Application source files.....	7
5.2	NCS libraries used directly by the application	7
5.3	IO and peripheral mappings.....	8
5.3.1	On-board sensors.....	8
5.3.2	Serial console	8
5.3.3	Serial recovery.....	8
5.3.4	Modem trace interface	9
5.4	Application flow	9
5.4.1	Main	9
5.4.2	Normal mode	10
5.4.3	Command mode.....	12
5.5	Command mode.....	13
5.6	Kconfig.....	13
5.7	Cloud parameters.....	14
5.8	Building using SEGGER Embedded Studio for ARM (Nordic Edition).....	14
5.9	Collecting modem trace	16
6	hci_uart	18
6.1	Building using SEGGER Embedded Studio for ARM (Nordic Edition).....	18
7	Pre-compiled image	21
8	OTA test image.....	21
9	Alias defined in AVT9152 EVB Device Tree Source (DTS) files	22

1 Introduction

This guide provides more information about the AVT9152 EVB SDK downloadable from [AVT9152 DEMO repo](#).

2 Directory structure

<your working directory>

```
|-- AVT9152_DEMO
|  |-- boards                - zephyr board files for AVT9152 EVB
|  |  |-- arm
|  |    |-- nrf9160_avt9152  - for on-board nRF9160
|  |    |-- nrf52840_avt9152 - for on-board nRF52840
|  |-- applications
|  |  |-- demo_IoTConnect    - source files of demo software in AVT9152's nRF9160
|  |  |-- hci_uart           - source files of demo software in AVT9152's nRF52840
|  |-- precompiled_image     - pre-compiled demo software hex and MCUboot compatible update
|  |                           binary files of projects included in the applications folder
|  |-- ota_test_image        - pre-compiled update image for over-the-air application
|  |                           through IoTConnect
|  |-- docs                  - documents
```

3 Required external SDK and library

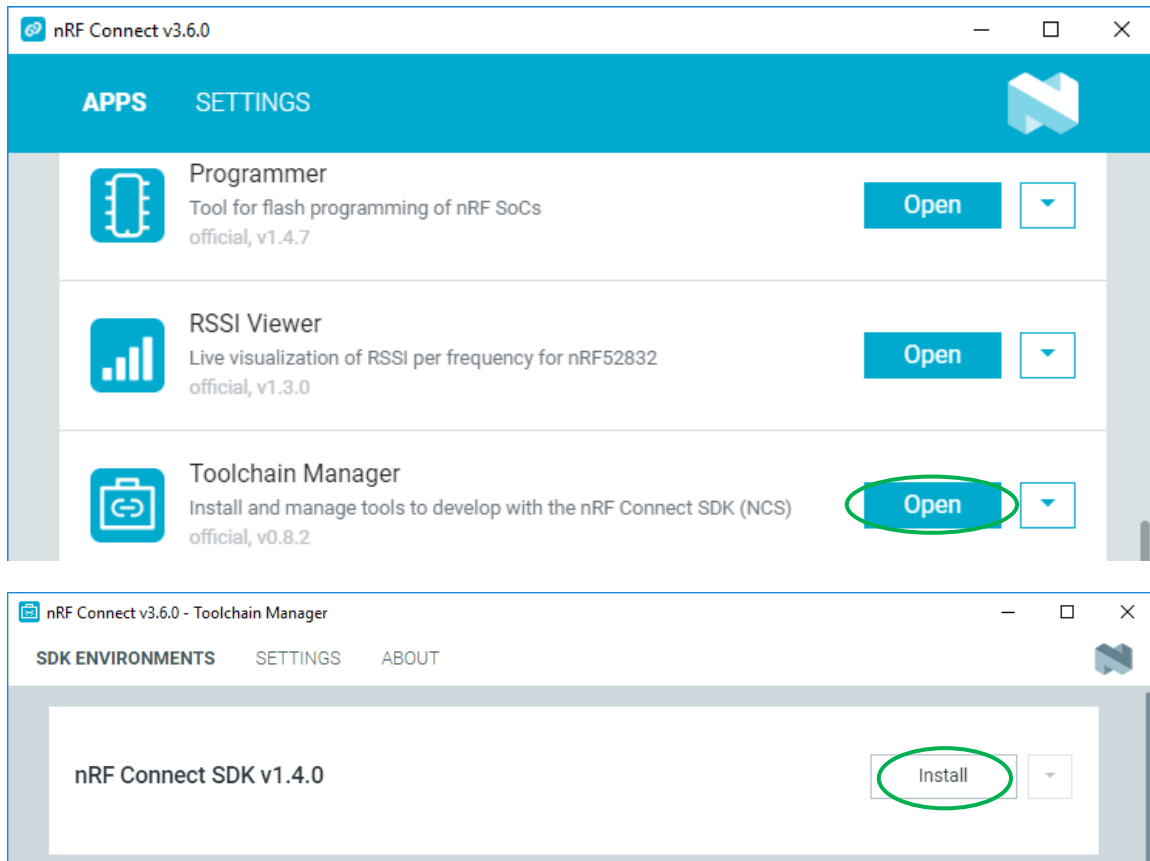
The following SDK and library from Nordic are needed in order to build the application projects provided in AVT9152 EVB SDK.

3.1 nRF Connect SDK

nRF Connect SDK (NCS) version 1.4.0 is the base SDK/code used to develop AVT9152 EVB demo software.

You can get it through nRF Connect for Desktop Toolchain Manager which also helps install and manage the needed tools to work on NCS.

Click the corresponding **Install** button for **nRF Connect SDK v1.4.0**.



By default, it will be downloaded and installed into your `%USERPROFILE%\ncs\v1.4.0\` folder. We will be referring to this folder as your *<NCS folder>*.

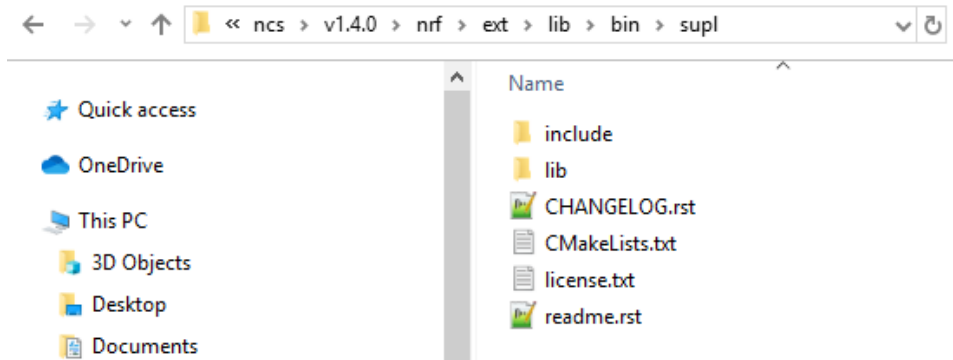
You can find more information on nRF Connect SDK from http://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/index.html.

3.2 nRF9160 SUPL client library

SUPL library is used to receive A-GPS data from a SUPL server using the SUPL protocol.

You can request for the SUPL client library from <https://www.nordicsemi.com/Software-and-tools/Development-Kits/nRF9160-DK/Download#supl-c>.

- 1) Create a "lib\bin" folder in your *<NCS folder>\nrf\ext*.
- 2) Extract the zip file into the created folder while maintaining the folder structure that is used in the zip file.



You can find more information on SUPL client and SUPL client OS integration library from https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/include/supl_os_client.html#supl-client.

The pre-flashed AVT9152 demo software is built with SUPL library v0.6.1.

4 nRF9160 modem firmware

The modem firmware provides AT-command and socket interface for the application to support LTE communication.

You can download this from <https://www.nordicsemi.com/Products/Low-power-cellular-IoT/nRF9160/Download#infotabs>.

AVT9152 EVB is pre-flashed with modem firmware v1.2.2.

5 demo_IoTConnect

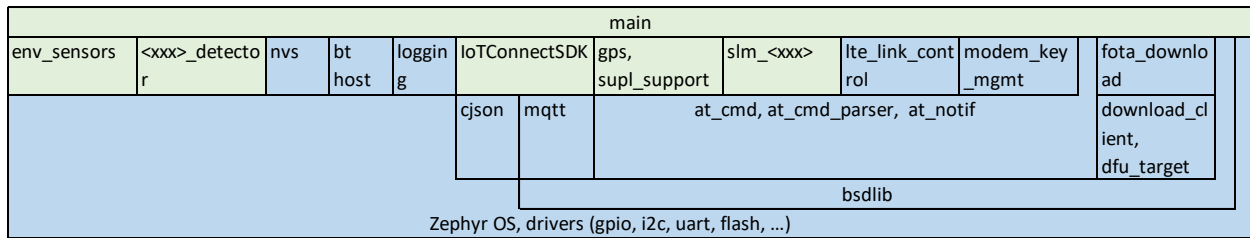
demo_IoTConnect demonstrates how to connect to [AVT9152 demo web portal](#) (that is based on [Avnet IoTConnect](#)) over LTE, transmits sensor and location tracking data as well as receive cloud to device commands. It also shows how to advertise Eddystone URL data using the on-board nRF52840 as the Bluetooth Low Energy controller.

demo_IoTConnect is a [multi-image](#) project with the following child images:

- [MCUboot](#) as the immutable bootloader provides the serial recovery function for firmware update over UART if activated as well as performs the image swapping of upgrade image downloaded over-the-air when present. It then jumps to the Secure Partition Manager which is the first image in the primary area.
- [Secure Partition Manager](#) (SPM) configures resources as non-secure for application in the non-secure region and jumps into it.

Application needs to run in the non-secure area to use the [bsdlb](#) to operate the LTE modem.

Shown below is a diagram of the top-level major components used by the main application. Those in **green** boxes are provided with AVT9152 EVB SDK while those in **blue** boxes are from the NCS.



5.1 Application source files

-- AVT9152_DEMO\applications\demo_IoTConnect	
-- src	
-- main.c	- application entry point
-- env_sensors.c/h	- reads temperature and pressure, humidity and ambient light data regularly from the on-board lps22hbtr and 2314277-1 Ambimate sensor respectively
-- <xxx>_detector.c, <xxx>_def.h	- detects device orientation and double-tap, rotation on z-axis and PIR motion from the on-board lis2dh12, i3g4250d and 2314277-1 Ambimate sensor respectively
-- slm_<xxx>.c/h	- handles proprietary AT commands while in command mode
-- gps.c/h, supl_support.c/h	- checks for GPS location fix data regularly as well as initiates SUPL download and injection as needed
-- IoTConnect_Config.h	- IoTConnect parameters file
- IoTConnect	- provides APIs to communicate with IoTConnect cloud
-- libiotconnect.a	
-- IoTConnect.h	
-- CMakeList.txt	- project file (specify MCUboot, SPM child images config files, source files listing)
-- prj.conf	- main application configuration file
-- Kconfig	- application configuration options
-- child_mcuboot.conf	- MCUboot configuration file
-- child_secure_partition_manager.conf	- SPM configuration file

5.2 NCS libraries used directly by the application

lte_link_control	Used to manage LTE connection.
modem_key_mgmt	Used to check presence of needed keys and certificates to connect to IoTConnect.
fota_download	Used to download and store the upgrade image from an https server.
at_cmd	Used to send at commands.
at_notif	Used to receive at command notifications (response, unsolicited events).
at_cmd_parser	Used to parse at commands, response, unsolicited events.
mqtt	Used to manage connection with and publish/subscribe data to/from IoTConnect MQTT server.

nvs	Used to store default LTE mode in flash.
bt host	Used to enable the BLE modem and start advertising.
logging	Used to print log messages.
cjson	Used to encode and decode json objects.

5.3 IO and peripheral mappings

5.3.1 On-board sensors

The following IOs are used to interface with the different sensors included in the EVB.

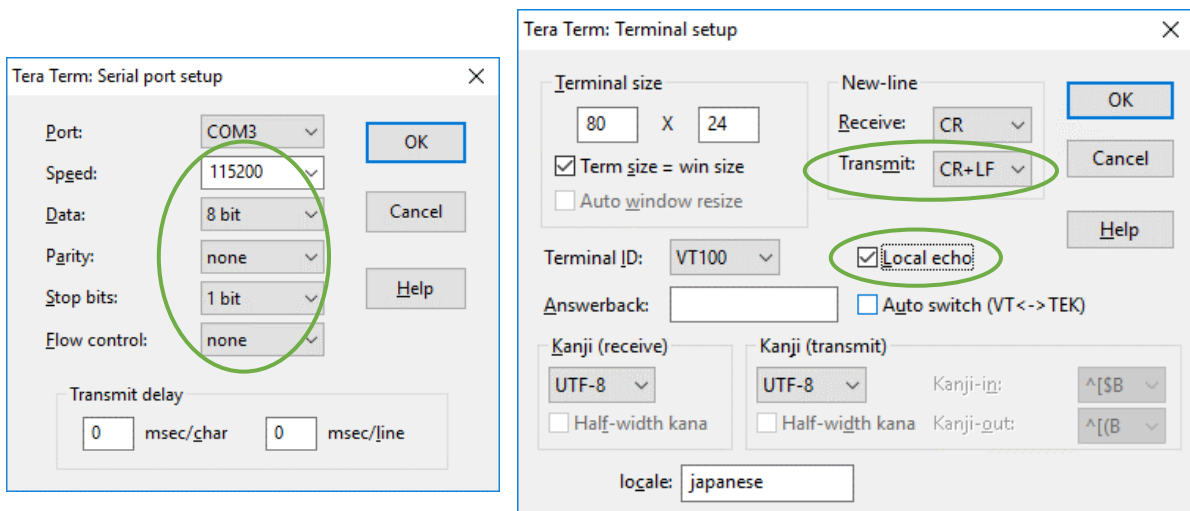
IO Pin	Alias	Connecting to	Usage
P0.22	sens_i2c	sensor I2C SCL	Interface to on-board sensors
P0.23		sensor I2C SDA	
P0.15	ard_d0	2314277-1 EVENT_OUT	PIR motion detection interrupt
P0.16	ard_d1	LIS2DH12 INT1	Orientation change interrupt
P0.20	ard_d6	LIS2DH12 INT2	Tap detection interrupt
P0.13	ard_a2	I3G4250D INT1	Z high event interrupt
P0.14	ard_a3	I3G4250D INT2	Data ready interrupt

5.3.2 Serial console

uart_0 (RX P0.24, TX P0.25) is configured as the serial console that is used by the application to show log messages as well as send/receive AT commands depending on the mode it is in.

This is accessible through AVT9152 EVB P1.

You can use a serial terminal program to connect to it at 115200 baud rate, 8 data bits, 1 stop bit, no parity, no flow control. Configure transmit to end with CR+LF which is the terminating character for AT commands and enable local echo in order to see the commands being entered.



5.3.3 Serial recovery

MCUboot enters serial recovery mode when P0.21 is low on boot up.

Make sure to connect this pin to GND only when intended. Otherwise, leave it floating.

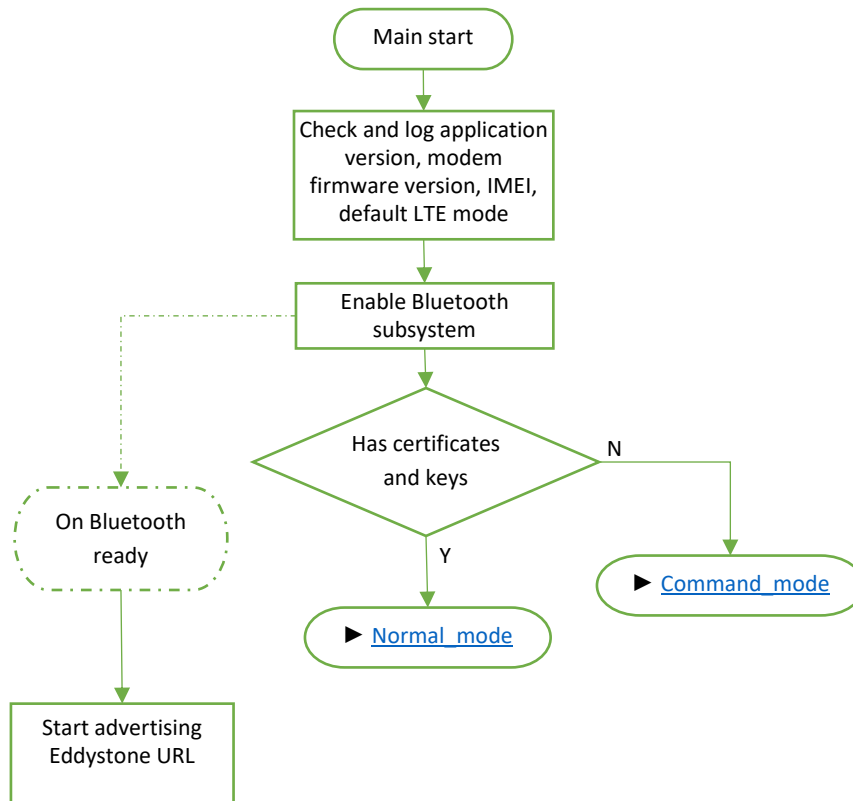
5.3.4 Modem trace interface

Modem trace when enabled is channeled through uart_1 (TX P0.21). This is disabled by default.

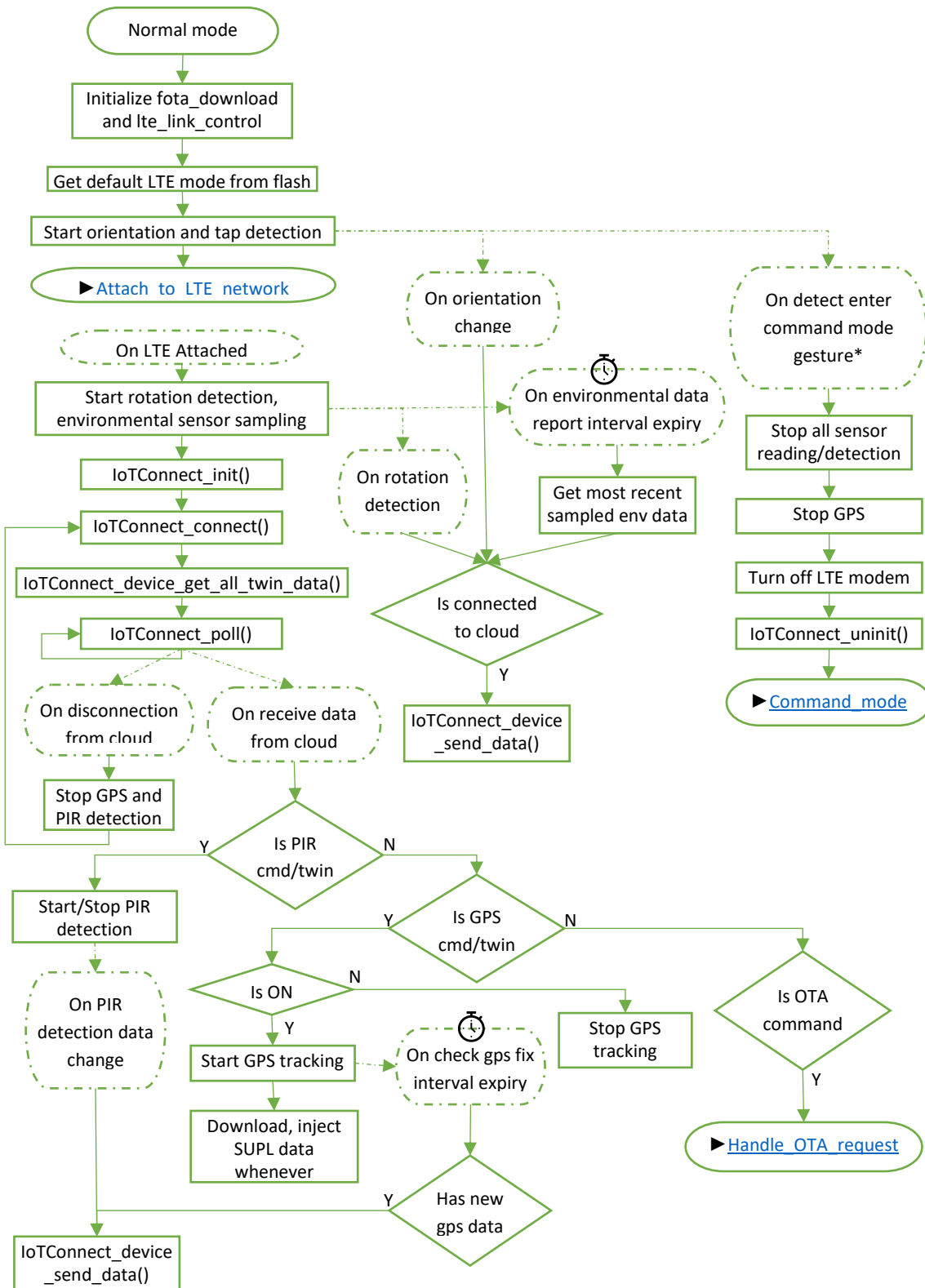
See [Collecting modem trace](#) on how to enable it and start collecting trace data.

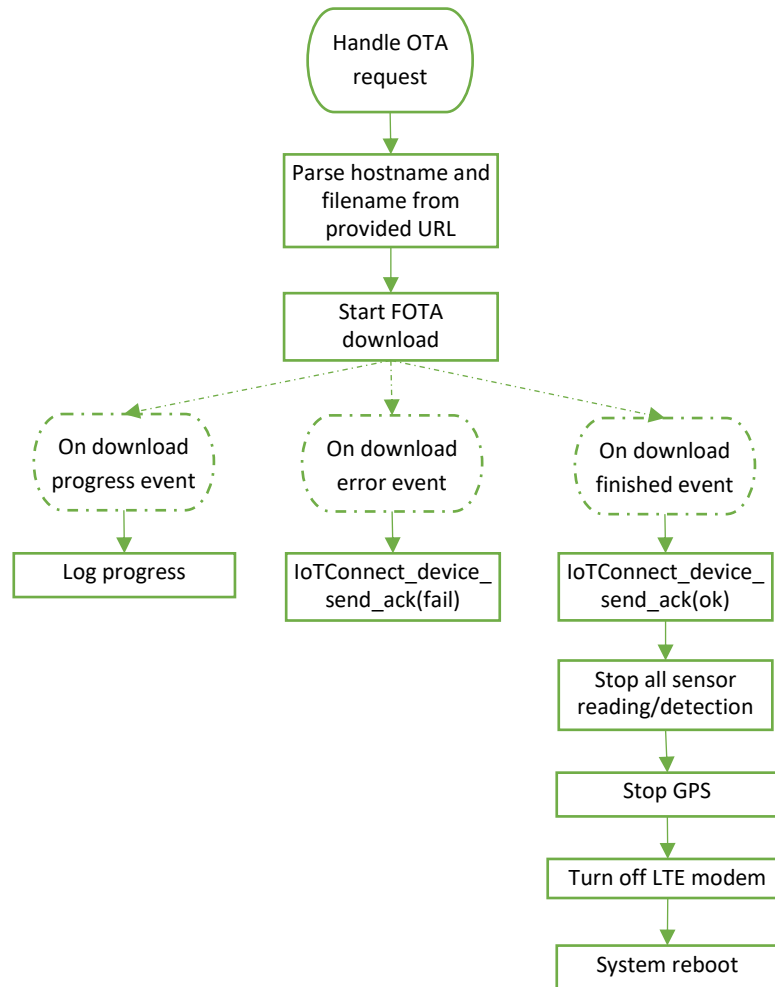
5.4 Application flow

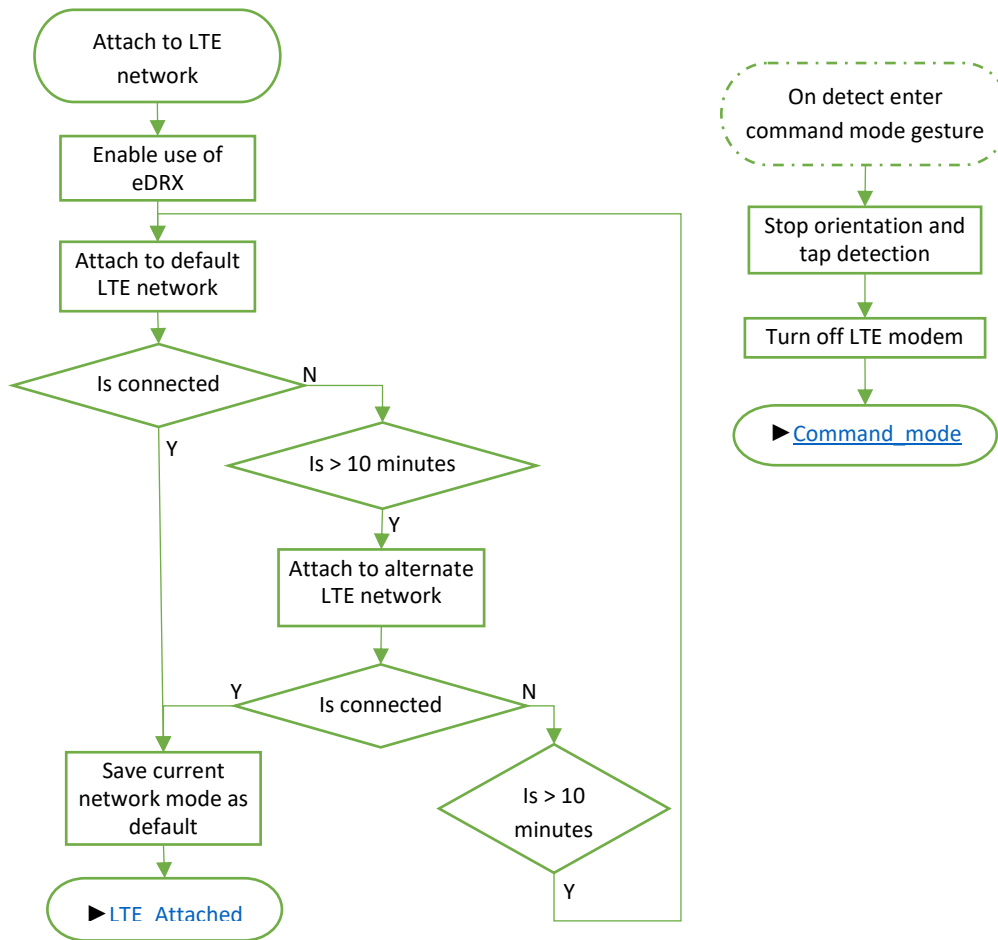
5.4.1 Main



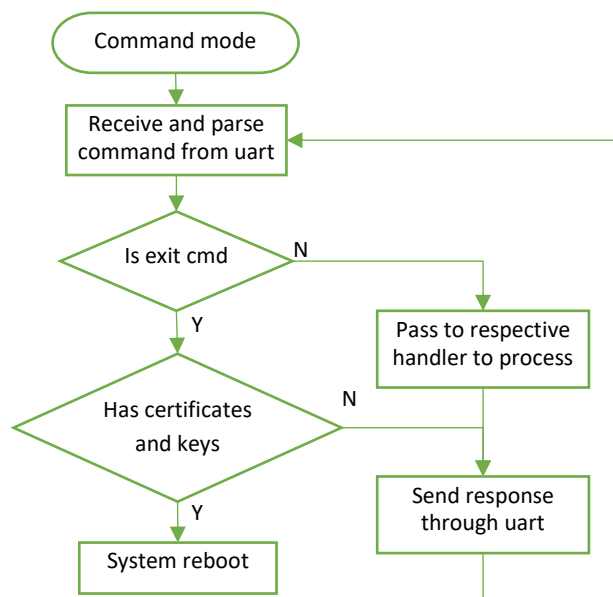
5.4.2 Normal mode







5.4.3 Command mode



5.5 Command mode

A command mode is provided to allow user to configure credentials needed to connect to the cloud as if the target is running the at_client sample from NCS.

You can explicitly switch to this mode by doing a double tap along the middle area where U2 LIS2DH12 is installed while the target is in a face down position.

You can start issuing AT command from the terminal program after seeing the **Ready** response as shown below.

```
[00:00:28.301,940] <inf> app: Device orientation: FD
[00:00:29.618,896] <inf> app: Tap detected with device orientation FD
[00:00:29.625,885] <inf> app: Preparing to switch to command mode...
[00:00:30.086,761] <dbg> app.lte_lc_evt_handler: nw_reg_status 0
[00:00:30.310,394] <inf> lte_lc: Previously registered handler (0x00000000) dere
gistered
[00:00:30.319,030] <inf> app: Inside Command Mode
Ready
```

The following proprietary commands are supporting aside from those listed in

https://infocenter.nordicsemi.com/topic/ref_at_commands/REF/at_commands/intro.html

- AT#XLTEMODE
 - To configure the default LTE network mode to use when in application normal mode
 - Accepts 1 parameter.
 - **0** for LTE CAT-M1
 - **1** for LTE CAT-NB1
- AT#XEXIT
 - To exit Command mode

AT commands ends with <CR><LF>.

5.6 Kconfig

Below is a list of application introduced build-time options.

Config	Description
DEMO_ENVIRONMENT_DATA_SEND_INTERVAL	How often to report environmental sensor data to the demo web portal Default: 60 seconds
DEMO_AT_MAX_PARAM	Max number of AT command parameters to accept in command mode. Default: 8
DEMO_SUPL_SERVER	SUPL server to use when SUPL_CLIENT_LIB is enabled. Default: supl.google.com
DEMO_SUPL_SERVER_PORT	SUPL server port. Default: 7276

You can change the default configuration by adding it to the prj.conf with your desired value.

5.7 Cloud parameters

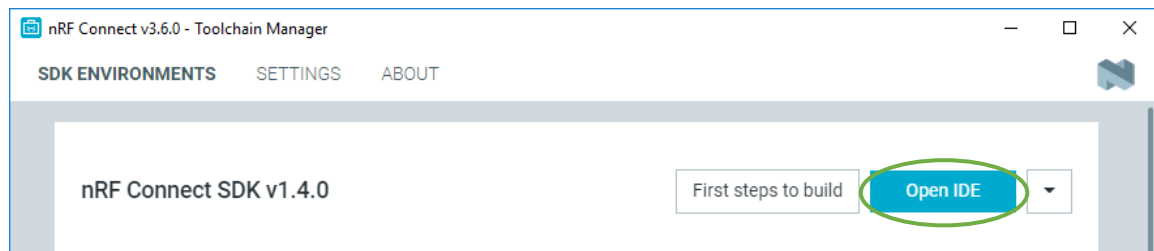
The following values defines the essential parameters specific to the IoTConnect cloud instance prepared for this demo. They must not be changed for the EVB to connect to [AVT9152 demo web portal](#).

Parameter in IoTConnect_Config.h	Value
CONFIG_IOTCONNECT_DEVICE_CP_ID	"149847e8f3ab4f11a90878bfa7da9922"
CONFIG_IOTCONNECT_DEVICE_ENV	"avnet"
CONFIG_IOTCONNECT_DEVICE_UNIQUE_ID	(must not be defined to use device's IMEI)
CONFIG_IOTCONNECT_DISCOVERY_SEC_TAG	623
CONFIG_IOTCONNECT_MQTT_SEC_TAG	624

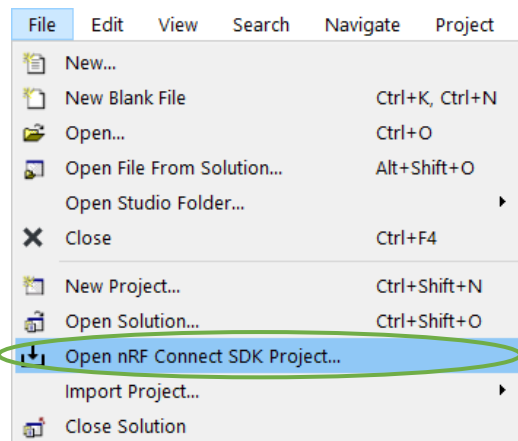
Care must be taken as not to overwrite or erase the pre-installed keys and certificates at the above-mentioned security tag.

5.8 Building using SEGGER Embedded Studio for ARM (Nordic Edition)

- 1) Open SEGGER Embedded Studio for ARM (SES).

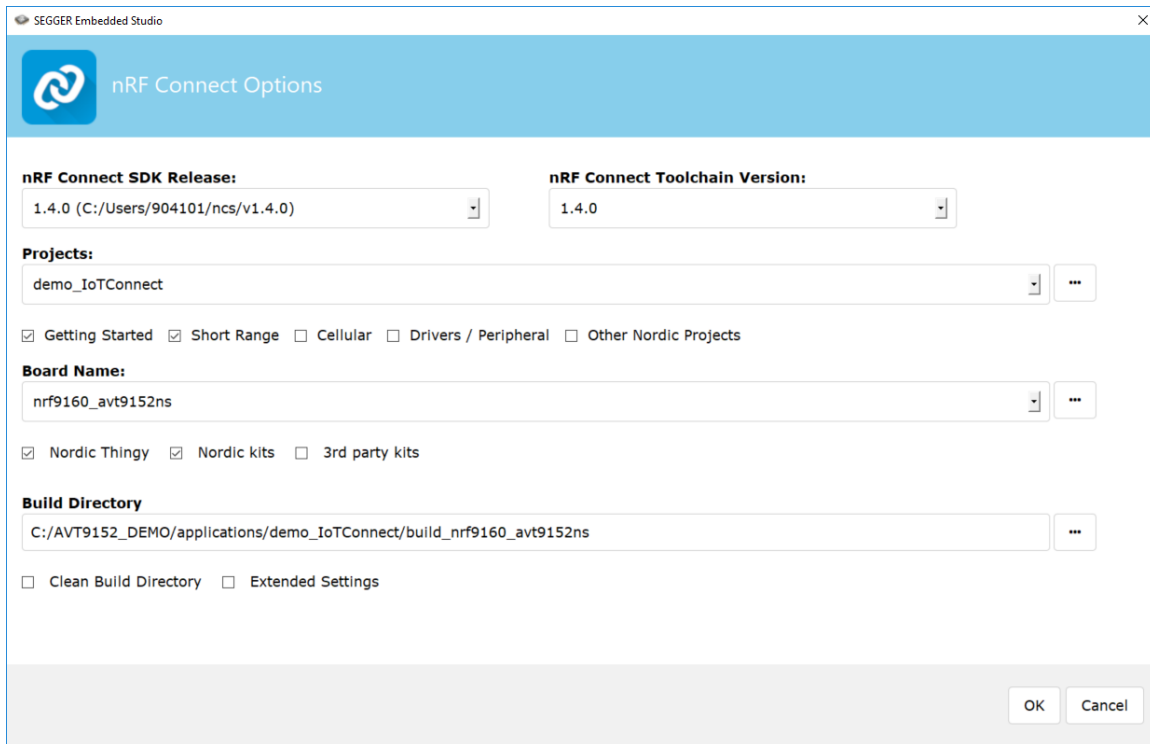


- 2) Select **File -> Open nRF Connect SDK Project...**

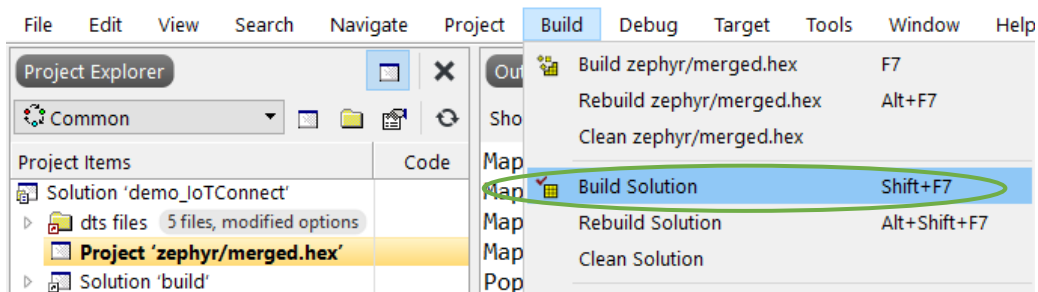


- 3) Fill in the **nRF Connect Options** dialog box accordingly.

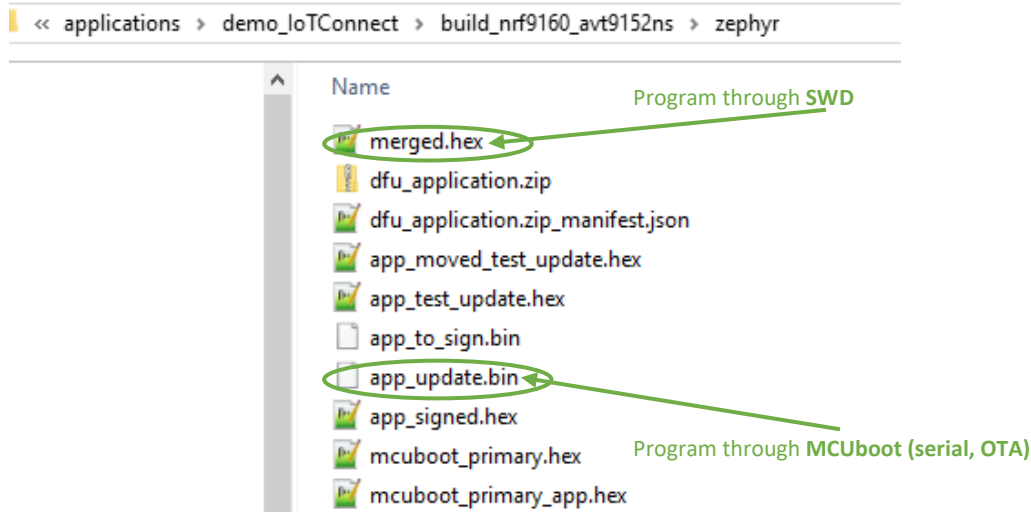
- a. Click the ... button on the right of the dropdown-list for **Projects** and select <your working directory>\AVT9152_DEMO\applications\demo_IoTConnect.
- b. Click the ... button on the right of the dropdown-list for **Board Name** and select <your working directory>\AVT9152_DEMO\boards\arm\nrf9160_avt9152.



- 4) Click **OK** and wait for SES to load the project.
- 5) Select **Build -> Build Solution** or press **Shift+F7** to build the application.



- 6) Once the build completes, you can find the **merged.hex** (containing the application, SPM, MCUboot) and the **app_update.bin** in your build directory <your working directory>\AVT9152_DEMO\applications\demo_IoTConnect\build_nrf9160_avt9152ns\zephyr\.



- 7) Please refer to [AVT9152 EVB Programming Guide](#) on how to download the new nRF9160 image to your EVB.

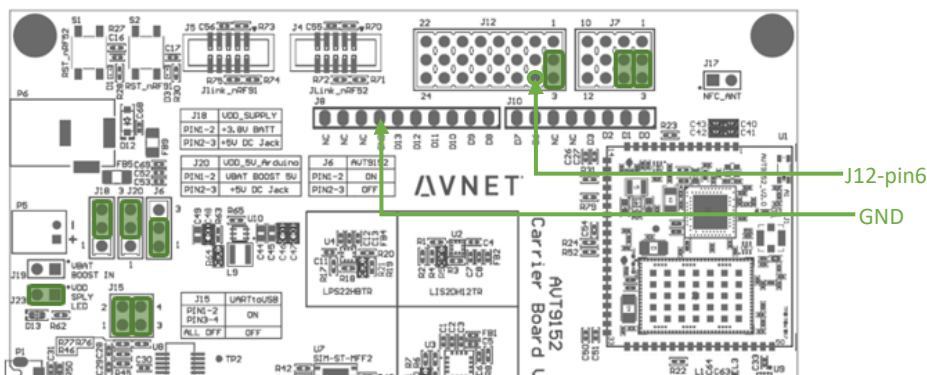
5.9 Collecting modem trace

Modem trace is at times requested by Nordic support team to help debug network related issues.

- 1) Enable CONFIG_BSD_LIBRARY_TRACE_ENABLED in the prj.conf.

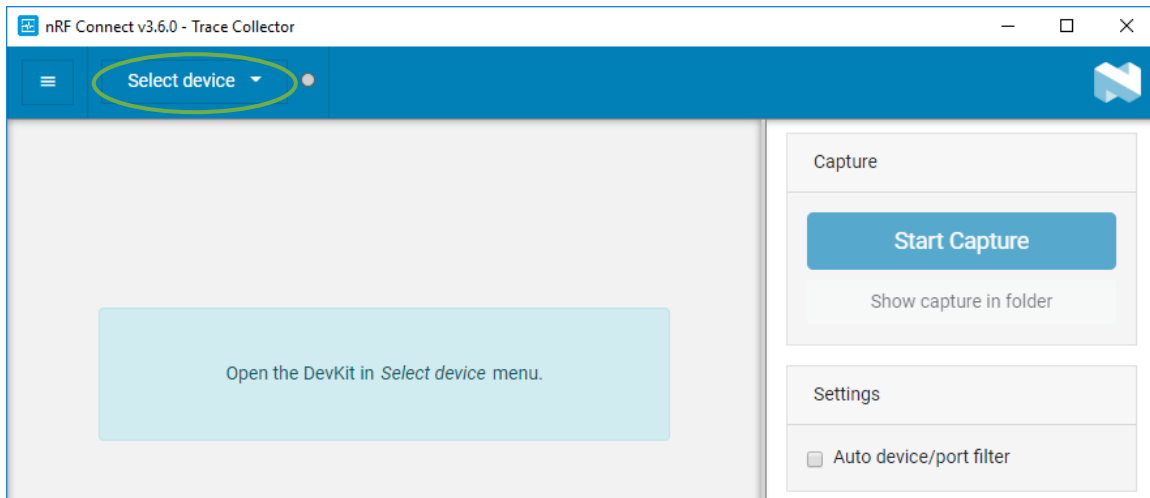
```
#-BSD-Library-for-nrf91
CONFIG_BSD_LIBRARY=y
CONFIG_BSD_LIBRARY_TRACE_ENABLED=y
```

- 2) Reload the project to apply the change in prj.conf if you already have the project opened.
- 3) Rebuild the solution.
- 4) Download the application onto your AVT9152 EVB.
- 5) Connect a USB-to-UART cable RX and GND pins to AVT9152 EVB J12-pin6 and J8 GND respectively.

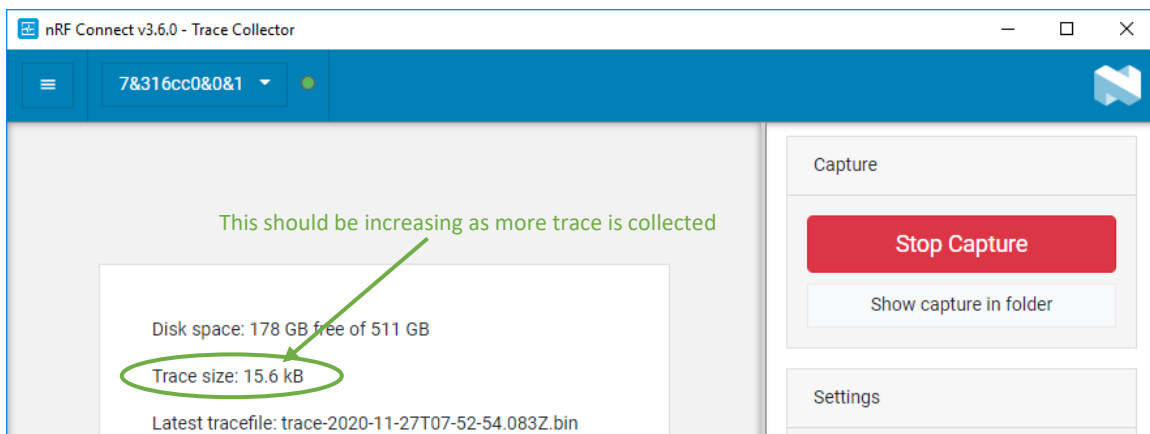
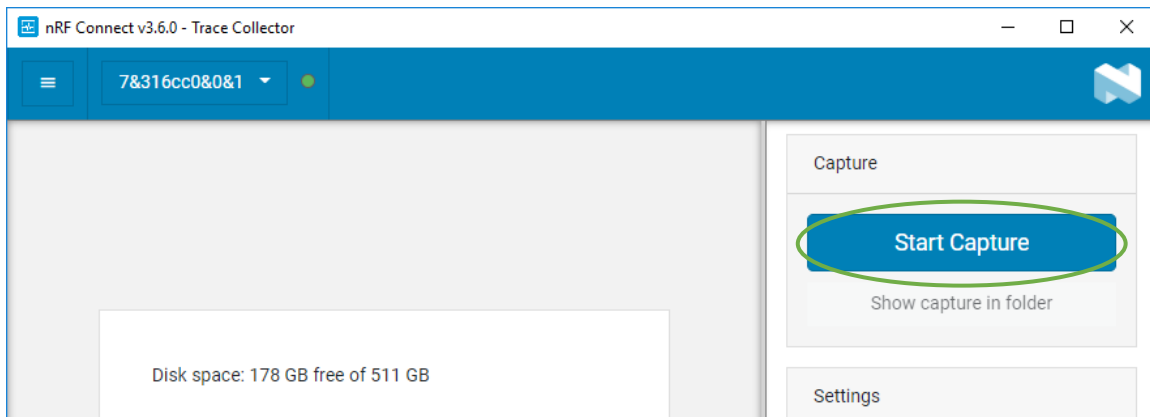


(Note: J12-pin6 is connected to nRF9160's P0.21 where trace data is being outputted)

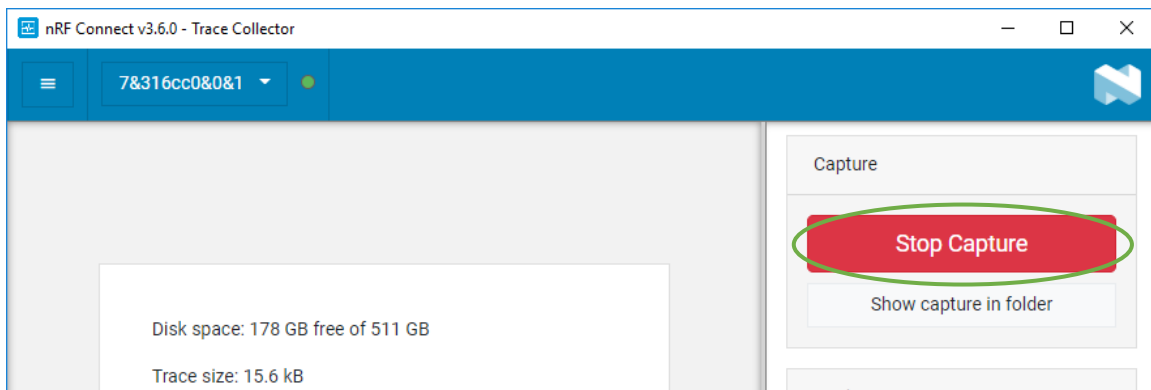
- 6) Open nRF Connect for Desktop Trace Collector and select the Virtual COM port associated to the cable inserted in previous step.



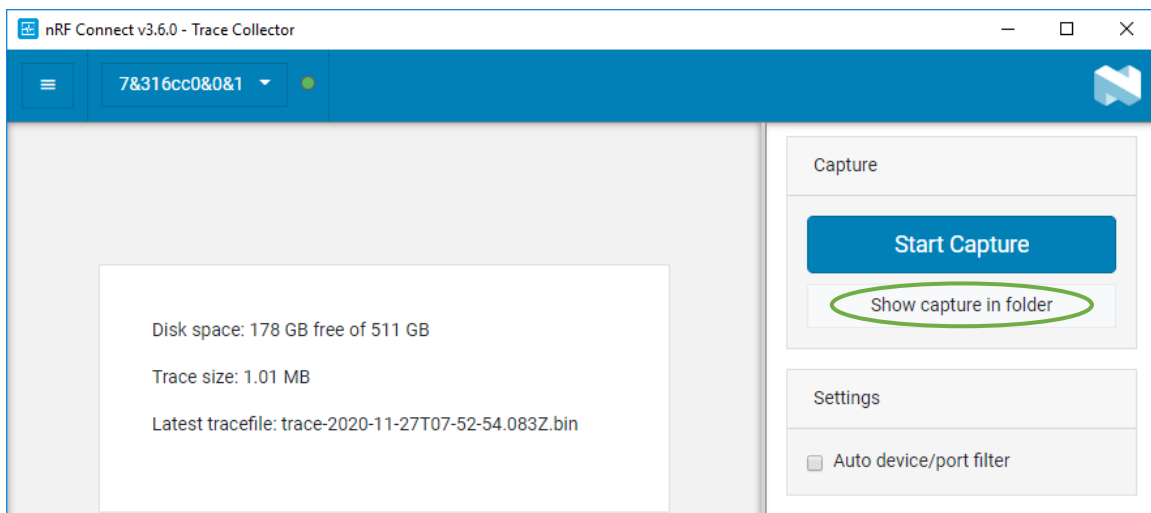
- 7) Click **Start Capture** for the tool to start collecting trace.



- 8) Click **Stop Capture** when done for the tool to close the trace file.



9) Click **Show capture in folder** to open locate the latest trace file.



6 hci_uart

[hci_uart](#) is the Bluetooth Low Energy (BLE) controller over UART application included in the Zephyr OS under <NCS installation folder>\zephyr\samples\bluetooth\hci_uart\.

This allows the use of on-board nRF52840 as the BLE modem to the nRF9160 BLE host enabled application like demo_IoTConnect.

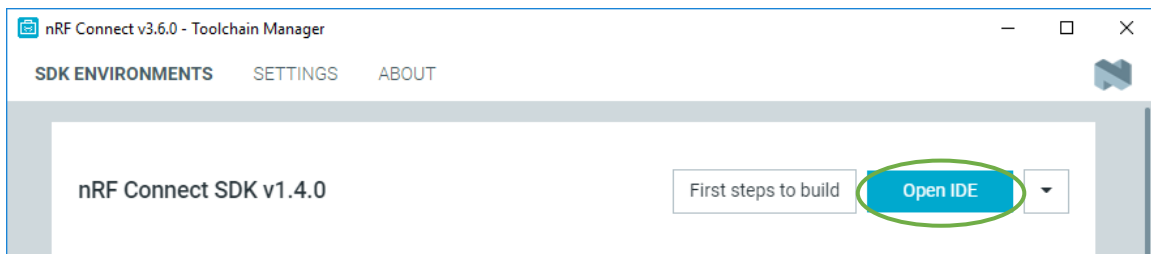
The copy provided in AVT9152_DEMO repository is modified to enable MCUboot serial recovery that allows firmware update over a virtual COM port exposed through P2 of AVT9152 EVB.

MCUboot enters serial recovery mode when P0.21 is low on boot up.

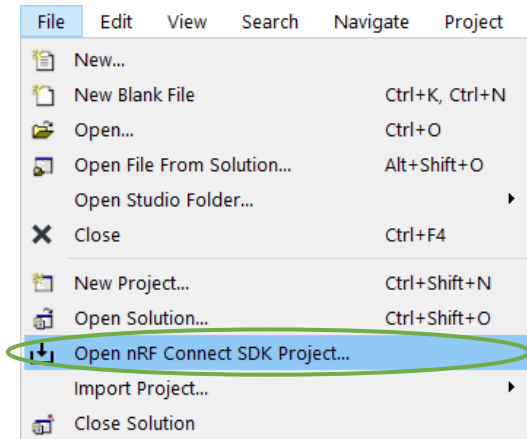
Make sure to connect this pin to GND only when intended. Otherwise, leave it floating.

6.1 Building using SEGGER Embedded Studio for ARM (Nordic Edition)

- 1) Open SEGGER Embedded Studio for ARM (SES).

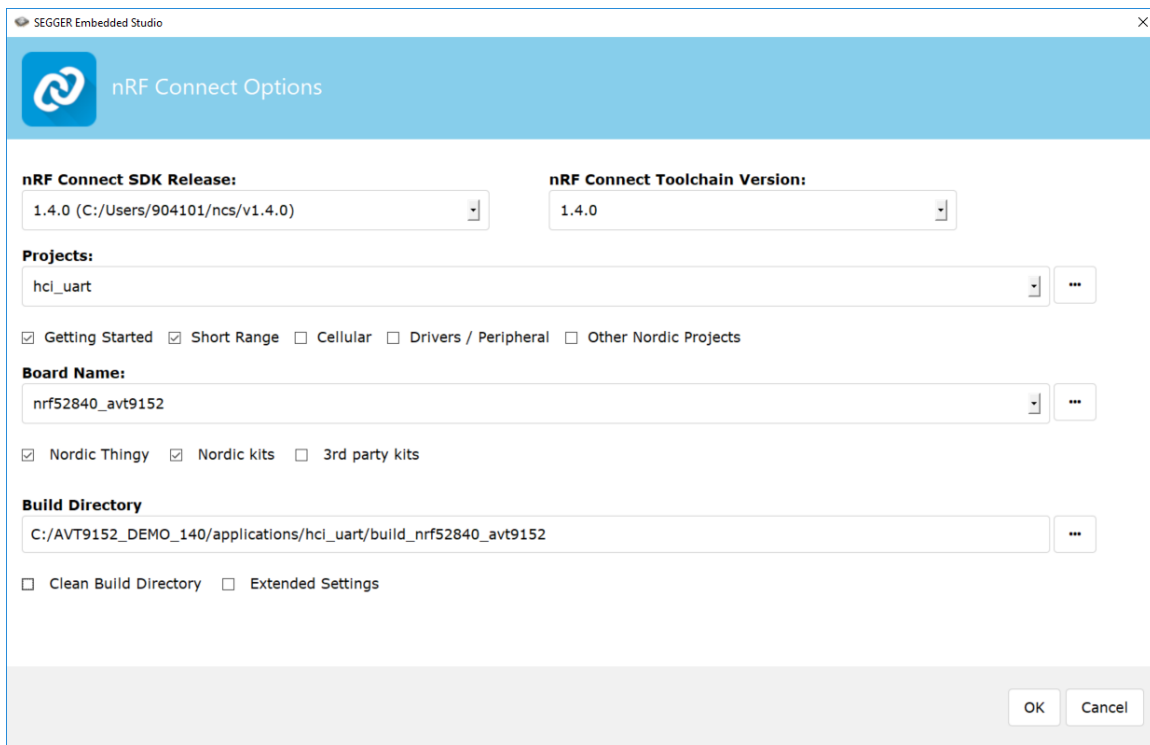


2) Select **File** -> **Open nRF Connect SDK Project...**

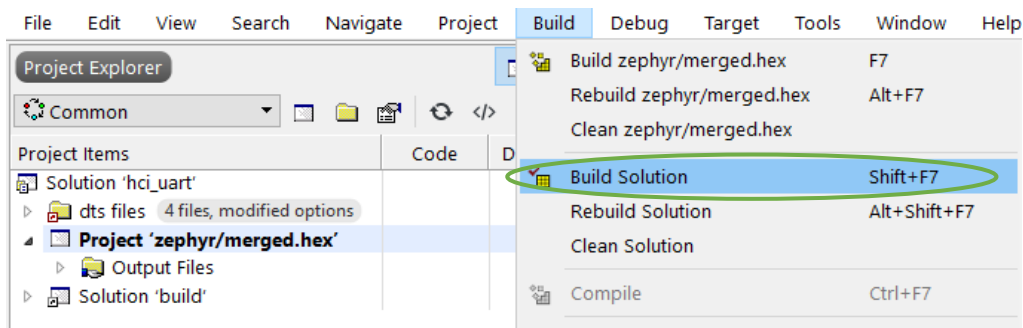


3) Fill in the **nRF Connect Options** dialog box accordingly.

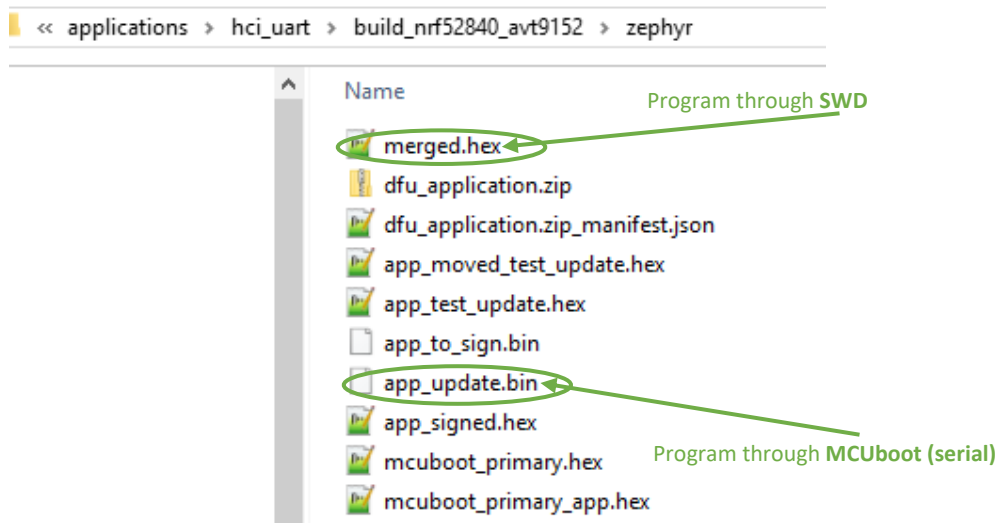
- a. Click the ... button on the right of the dropdown-list for **Projects** and select <your working directory>\AVT9152_DEMO\applications\hci_uart.
- b. Click the ... button on the right of the dropdown-list for **Board Name** and select <your working directory>\AVT9152_DEMO\boards\arm\nrf52840_avt9152.



- 4) Click **OK** and wait for SES to load the project.
- 5) Select **Build -> Build Solution** or press **Shift+F7** to build the application.

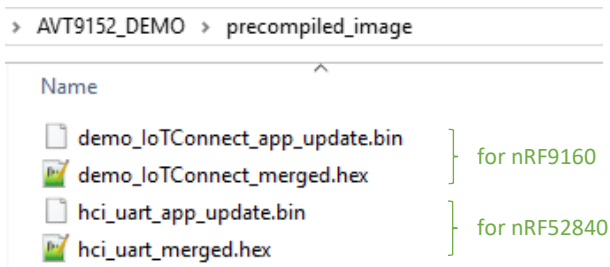


- 6) Once the build completes, you can find the merged.hex (containing the application, SPM, MCUboot) and the app_update.bin in your build directory <your working directory>\AVT9152_DEMO\applications\ hci_uart\build_nrf52840_avt9152\zephyr\.



- 7) Please refer to [AVT9152 EVB Programming Guide](#) on how to download the new nRF52840 image to your EVB.

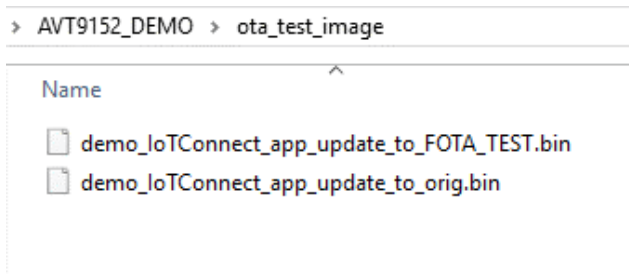
7 Pre-compiled image



This folder contains copy of pre-compiled images of each project under the applications folder.

- <project name>_merged.hex - used when programming image through SWD using nRF Connect Programmer
- <project name>_app_update.bin - used when programming through MCUboot using mcumgr

8 OTA test image



The **FOTA_TEST** image is practically the same as the **orig** image except for having a higher firmware version number.

The **orig** image variant is the same as the demo_IoTConnect_app_update in precompiled_image folder.

9 Alias defined in AVT9152 EVB Device Tree Source (DTS) files

Below is the list of aliases defined in

<your work folder>\AVT9152_DEMO\boards\arm\nrf9160_avt9152\nrf9160_avt9152_common.dts and
 <your work folder>\AVT9152_DEMO\boards\arm\nrf52840_avt9152\nrf52840_avt9152.dts.

Alias	Define for (chip)	Description
ard_a2	Both	Arduino A2 pin
ard_a3	Both	Arduino A3 pin
ard_a4	Both	Arduino A4 pin
ard_a5	Both	Arduino A5 pin
ard_d0	Both	Arduino D0 pin
ard_d1	Both	Arduino D1 pin
ard_d2	Both	Arduino D2 pin
ard_d3	Both	Arduino D3 pin
ard_d6	Both	Arduino D6 pin
ard_d7	Both	Arduino D7 pin
ard_d8	Both	Arduino D8 pin
ard_d9	Both	Arduino D9 pin
ard_d10	Both	Arduino D10 pin
ard_d11	Both	Arduino D11 pin
ard_d12	Both	Arduino D12 pin
ard_d13	Both	Arduino D13 pin
interconn_c0	Both	Inter-connection C0 pin
interconn_c1	Both	Inter-connection C1 pin
interconn_c2	Both	Inter-connection C2 pin
interconn_c3	Both	Inter-connection C3 pin
interconn_c4	Both	Inter-connection C4 pin
interconn_c5	Both	Inter-connection C5 pin
sens_i2c	Both	I2C master peripheral to communicate with all built-in sensors.
nrf52_reset	nRF9160	nRF52840 reset control pin
debug_uart	nRF9160	UART peripheral for debug print via USB port (P1)
nrf91_coex0	nRF52840	nRF52 pin connected to nRF91 COEX0 pin
nrf91_coex1	nRF52840	nRF52 pin connected to nRF91 COEX1 pin
nrf91_coex2	nRF52840	nRF52 pin connected to nRF91 COEX2 pin
nrf91_reset	nRF52840	nRF9160 reset control pin
pwr_good	nRF52840	PG status from DC-DC

You can refer to the env_sensor.c or <xxx>_detector.c files from the demo_IoTConnect application on how to use alias labels configure and use peripheral drivers.