**DID Write**

Writing data into NVM

1)Creating a data block in nvm to be written

1) Add parameter in coding sheet

Take care of padding while giving the size of the parameter

Size of the parameter indicate how many byte of data can be written into the nvm block

Here we are creating a API that will write into and read from nvm

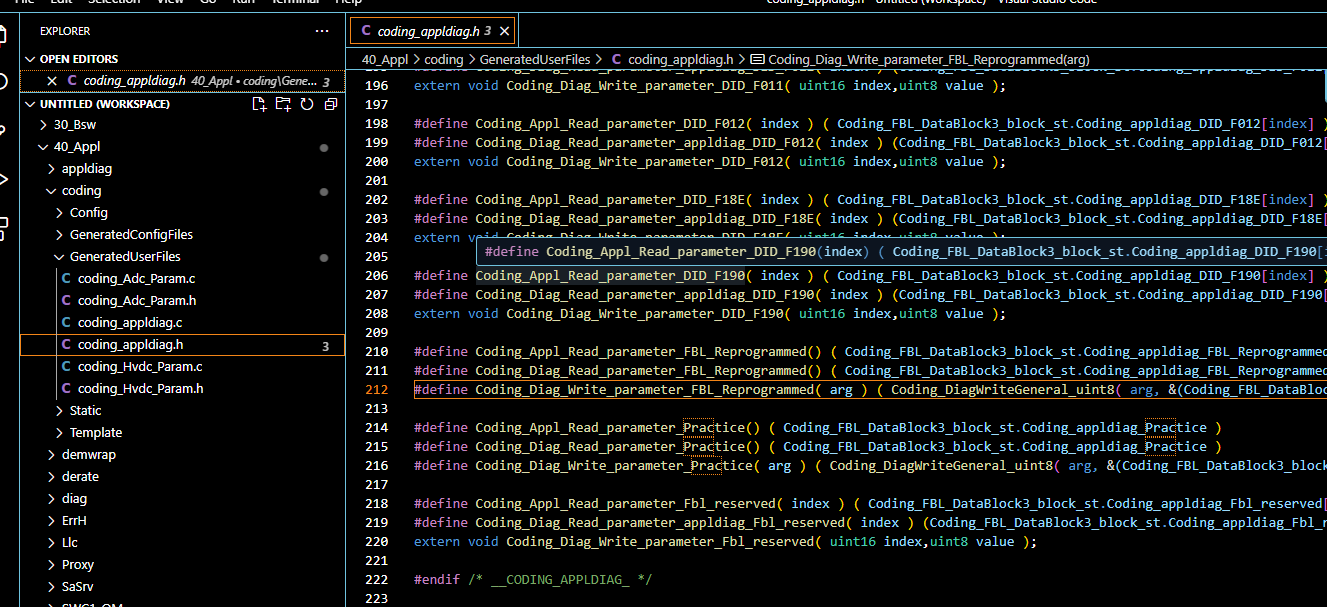
2) Save the excel file and generate

3)Copy to appl

Now new API will be created in coding swc

Since we initialised a block for appldiag in the excel, we need to look in file coding\_appldiag.h

In the header file there will be read and write API created



Coding\_Appl\_Read\_parameter\_Practice()

Coding\_Diag\_Read\_parameter\_Practice()

Coding\_Diag\_Write\_parameter\_Practice( arg )

These API can be used to write data into the nvm block and can be used to read data written by us in the block using read API

2) Giving write access to the DID on creation (here CodingWrite is the DID name with DID :0x1234)

We can create DID with data size (here 1byte) same as size provided in coding sheet

3) On creating DID a client port is created in DCM

4) Create a Server port in appldiag using the port interface created while making client port in DCM

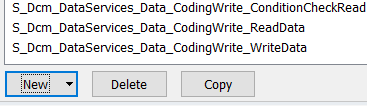
5) Create a server runnable using the port, which will create 3 server runnable

Read, write and condition check

6)Now connect both S and C port in cfg

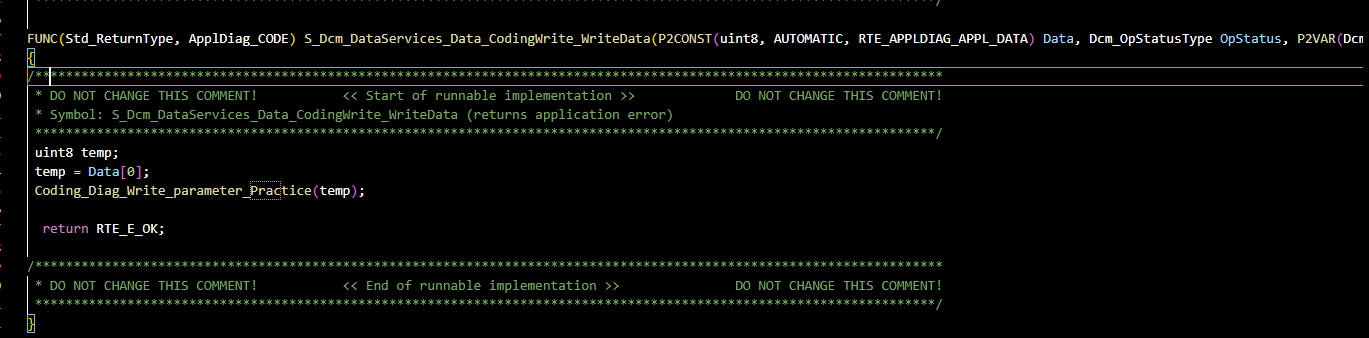
7) Now generate the c files for appldiag

We can see that 3 templates for read, write and condition check functions are created in appldiag.c

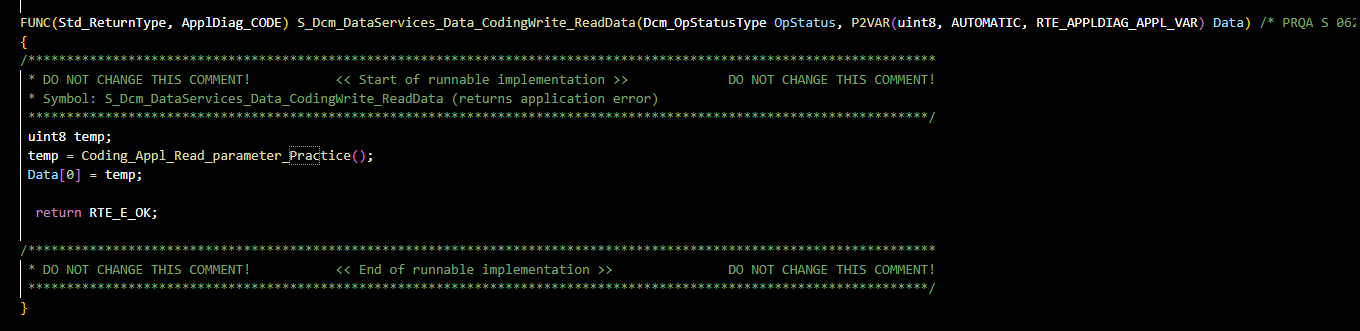


Now we can use the created API in Write and Read function template created in appldiag.c to write the data into nvm block and read that data from the block.

In appldiag.c we can use the write API in the Writedata function



Read API can be used to read the written data



Note: Here we are using coding swc which helps to write data into NVM

SWC Coding is responsible for creating the API which we use in appldiag to achieve the write to and read from NVM

Now compile and test using 2E SID which is used for write data in UDS protocol

Eg: 2E 1234 0A 0B 0C 0D here 1234 is the SID and remaining 4 bytes are data.