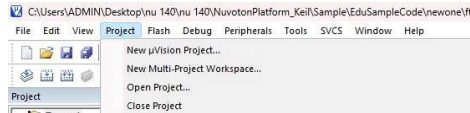


## A) Steps For .asm files

### STEP 1:

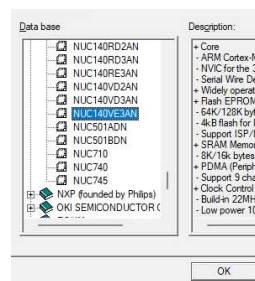
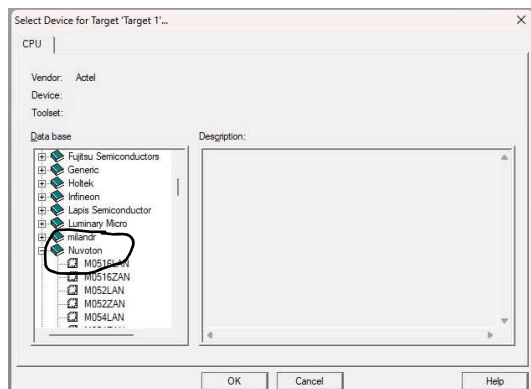
Open Keil Software -> Go to project Tab -> New  $\mu$ Vision Project -> Go to nu folder on Desktop -> Nuvoton Platform -> Sample -> EduSample -> Create a new folder -> create a file(without extension)



### STEP 2:

A dialogue box will appear where we need to select Device for Target.

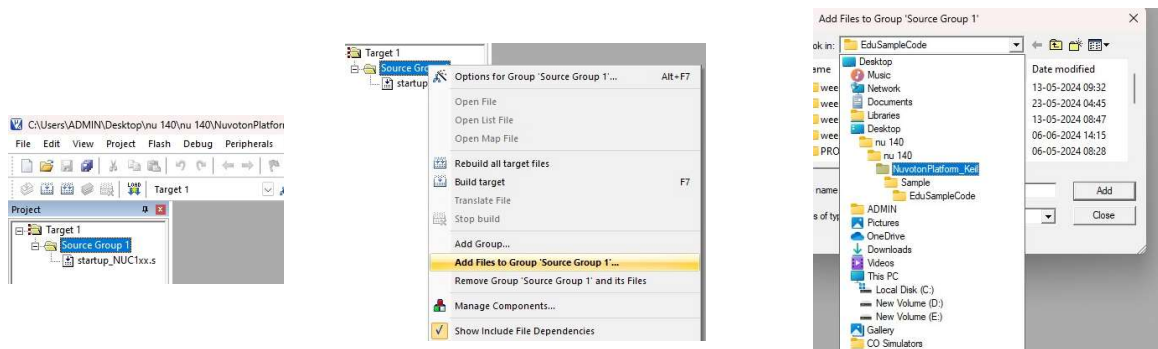
Under Database -> Select Nuvoton -> Select it last Version (NUC140VE3AN) -> click OK



### STEP 3:

click on Target1 -> Source Group1 (right click) -> Add files to group 1 -> Go to nu140 -> CMIS -> CMO -> CoreSupport -> core\_cm0.c

Do the same thing again but this time -> DeviceSupport -> Nuvoton -> NUC1xx -> system\_NUC1xx.c

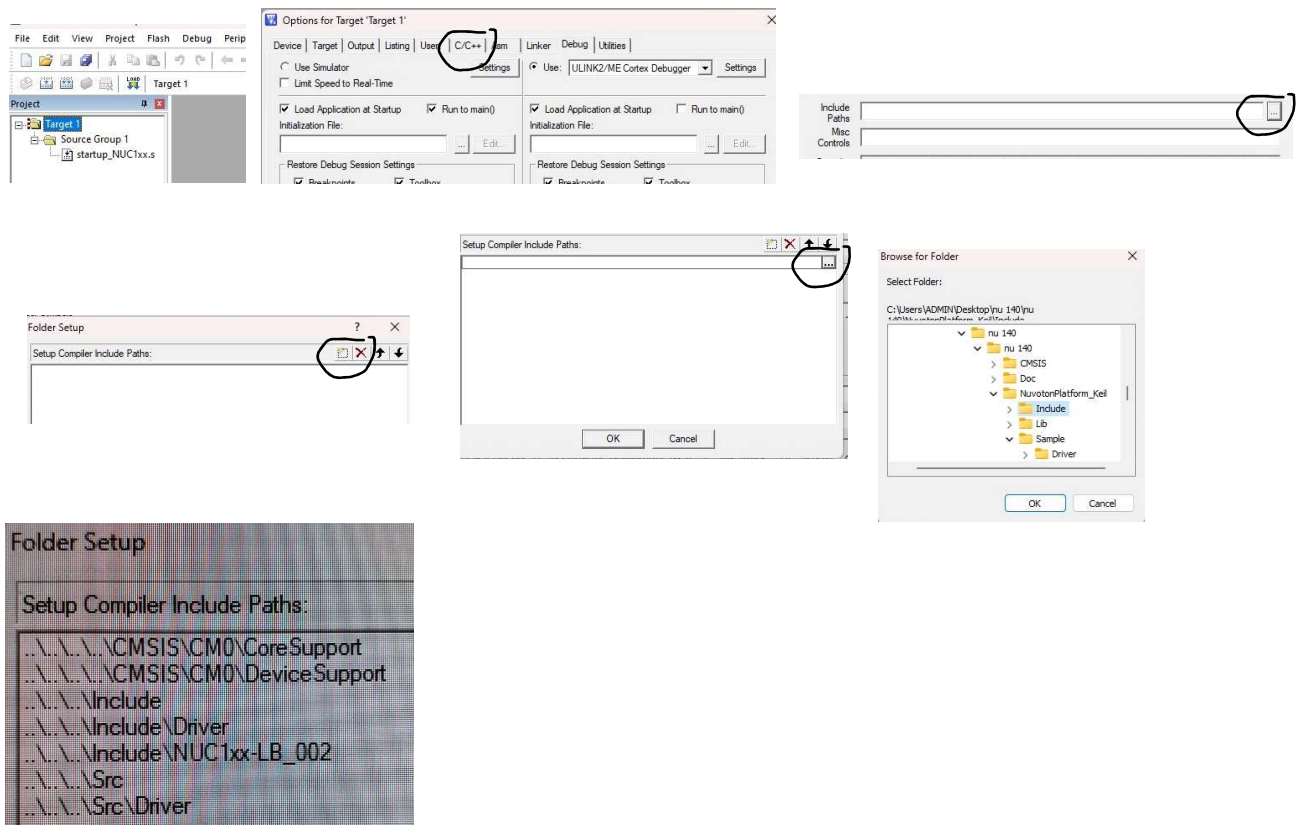


### STEP 4:

Target1(right click) -> options for target 1 -> C/C++ -> Include Paths (click on 3 dots) -> new Path (dotted rectangle) -> Go to nu140 -> CMIS -> CMO -> CoreSupport

Target1(right click) -> options for target 1 -> C/C++ -> Include Paths (click on 3 dots) -> new Path (dotted rectangle) -> Go to nu140 -> CMIS -> CMO -> DeviceSupport

Target1(right click) -> options for target 1 -> C/C++ -> Include Paths (click on 3 dots) -> new Path (dotted rectangle) -> Go to nu140 -> NuvtonPlatform\_keil -> Include



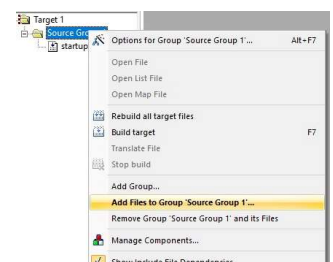
STEP 5:

click on new file -> type your code -> save in same path as in STEP 1

More specifically (to save): Go to nu folder on Desktop -> Nuvoton Platform -> Sample -> EduSample -> your previously created folder -> save with extension (.asm)

STEP 6:

Source Group 1(right click) -> Add files to group 1 -> Choose yourfile.asm -> Add



## STEP 7:

click on build (top left corner) -> if build successful -> click on debug (red color, middle of title bar)



Reference Video: <https://youtu.be/YSIkYs4NAzU?si=RZrJUMQkFCy58SFR>

## B) Steps For Nuvtron codes (Device):

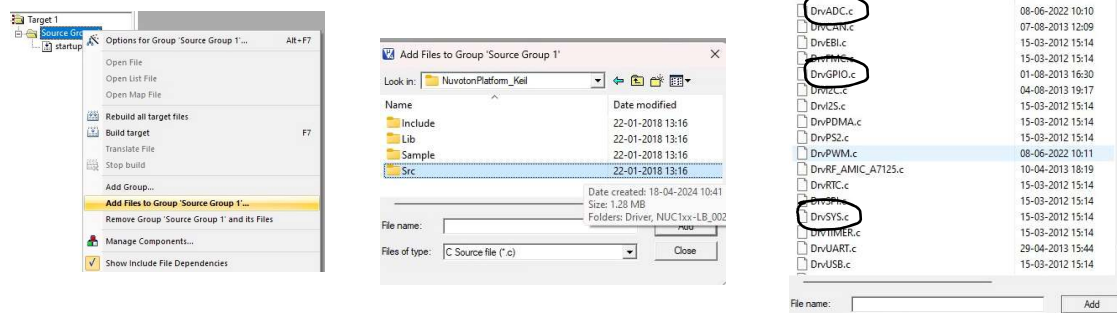
Follow above steps 1,2,3,4,5,6,7

Note: In step 6 keep file extension as ( .c )

## Step 8:

In Step 3 again:

Source Group1 (right click) -> Add files to Source Group 1 -> Go to nu140 -> SRC file -> Driver file -> add (DrvSYS.c , DrvGPIO.c , DrvADC.c) \*required files

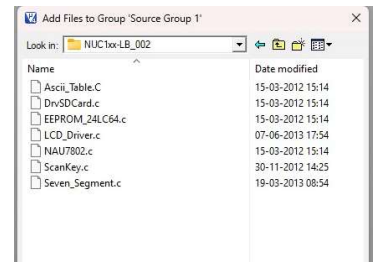
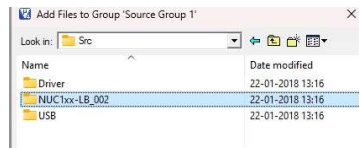
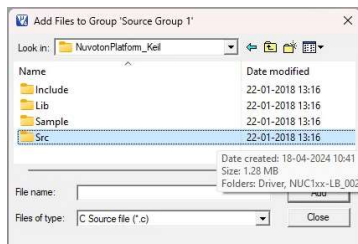


Similarly , according to the programs add the files like:

For Lcd Display :

Source Group1 (right click) -> Add files to group 1 -> Go to nu140 -> SRC file -> file -> led.c

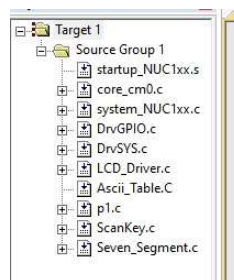
For 7 segment display: add seven\_segment.c



Note: Check the include division of the code you want to perform and add files according to it.

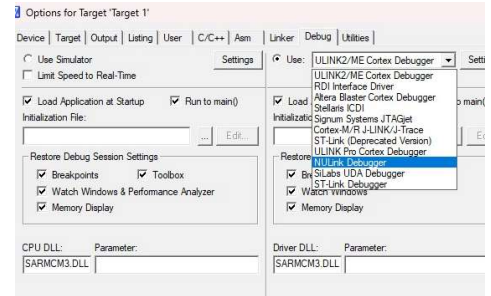
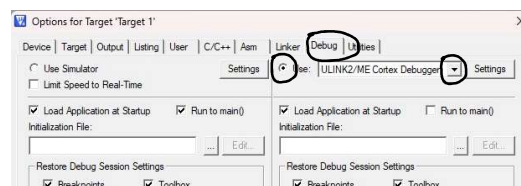
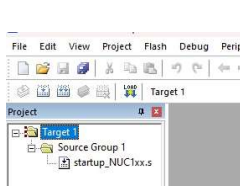
```
#include <stdio.h>
#include "NUC1xx.h"
#include "Driver\DrvSYS.h"
#include "Seven_Segment.h"
#include "DrvGPIO.h"

// display an integer on four 7-segment LEDs
void seg_display(int16_t value)
```

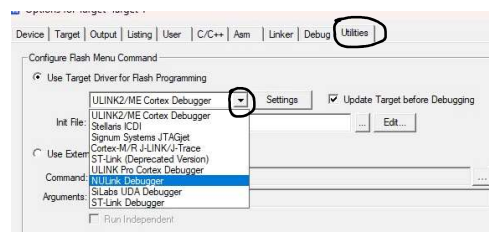


Step 9:

Target1(right click) -> options for target 1 -> Debug -> click on " O use " (right side of window) before debugger selection -> select " Nuvlinkere Debugger " as debugger in drop-down menu.



-> options for target 1 -> Utilities -> again select debugger as " Nuvlinker Debugger " in drop-down menu -> click ok -> save changes.



Step 10:

Save the file -> Attach the device with correct port -> Click on build -> Click on load button ( near build ) OR click on debug icon on right side of window.



Notes->

->notes link for 4<sup>th</sup> Sem MC-IOT lab and class: <https://drive.google.com/drive/folders/1ti3FtDDcJ-XAvLONniCURXklqfrMODhx>

Refrence video for Raspberry PI:

<https://drive.google.com/drive/folders/1ViEsMuA1DdNg8qpIpgyCqyS9A0hH6w-p>