

Keil u vision Software Installation Process.

The following are the steps to be followed to download & initiate the keil software (installation).

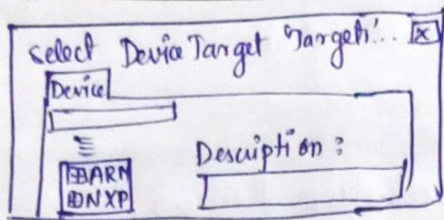
- 1) Google Keil.com
- 2) Go to download → Product Downloads → MDK - ARM
- 3) In MDK - ARM page, fill in details, company: VNRVJIT,
Job Title: Student
Last Column: LPC 1768 (Device name column)
- 4) Submit the details.
- 5) Click on setup file "MDK538A.EXE" (908.615 KMB)
- 6) Once, the .exe file is downloaded double click on the file for setup and run. Follow the instructions prompted during installation.

In software

- ① Package Installer opens.
- ② Select IC LPC 1768 (Nxp)
- ③ In device window, nxp LPC1700 series LPC 17xx ^{goto} → LPC 1768
- ④ ## packages on right get displayed
select keil: LPC1700_dtp package → install.
- ⑤ The package has to display "upto date" then close.
- ⑥ Keil u vision desktop icon gets displayed on desktop.

Project Creation

- ① Close window, Random project appears.
- ② Project folder, right click → new u vision project. (.uvproj, .uvprojx)
↓
extensions
- ③ In nxp → LPC1768 → select OK → select Yes.
- ④ Soon a window pops out for selecting device.



⑤ Expand NXP series.

[-] NXP

[-] LPC 1700 series


[+] LPC 175X

[+] LPC 176X

[+] LPC 177X

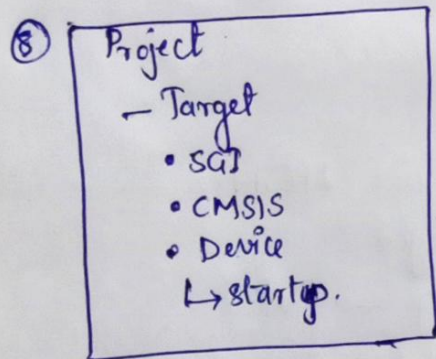
[+] LPC 178X

⑥ Expand LPC176X series >> select LPC1768. upon selection, the description box fills with info. click 'OK'.

⑦ Manage Runtime Environment window opens  ribbon icon.

select CMSIS >> CORE.

select Device >> startup.



Project window appears.

⑧ Double click on startup - lpc17xx.s
startup file opens.

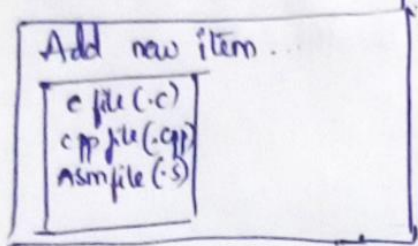
⑩ Upon file opening, comment few lines to enable assembly language editor.

Comment 127, 129, 130, 272, 273 i.e., add ; to lines.

⑪ Ctrl+S to save the file.

⑫ Right click on source Group 1 >> select

"Add new item to Group 'Source Group 1' window pops.



Select (.s) ASM file \rightarrow assembly language
Name the file.

⑬ Verify file name SG1 to confirm creation

⑭ Now start writing the instructions

```
AREA... ALIGN=2
```

```
...
```

⑮ Save program.

⑯ Right click on Target \gg Select options (Alt+F7).
window opens \rightarrow options for Target 'Target 1'

⑰ Now Goto Debug panel \gg select use simulator \gg click Ok.

⑱ Go to projects click on translate option for compilation

⑲ Now link all header files to the source file to create object file \gg select Build project in Project Panel.

⑳ Start/Stop debug then run the program.

Note : To edit a file come out of debug mode, edit, translate and build the project and debug and run.

Write an assembly program to add two numbers,

Program :

; Addition of 2 numbers.

AREA ADDITION, CODE, READONLY, ALIGN = 2

ENTRY

EXPORT --main

--main

MOV R0, #10

MOV R1, #15

ADD R1, R0

STOP B STOP

END.

Op: In register window.

R0 0x0000000A

R1 0x0000000F

After execution.

R1 0x00000015

Write a program to perform arithmetic operations in assembly language on two operands.

Program :

```
AREA ARITHMETIC, CODE, READONLY
```

```
ENTRY
```

```
EXPORT __main
```

```
NUM1 DCD 0x0000000B
```

```
NUM2 DCD 0x0000000E
```

```
-- main
```

```
LDR R0, NUM1
```

```
LDR R1, NUM2
```

```
ADDS R2, R1, R0
```

```
SUBS R3, R1, R0
```

```
MUL R4, R1, R0
```

```
UDIV R5, R1, R0
```

```
STOP B STOP
```

```
END
```

output :

```
R0 0x0000000B
```

```
R1 0x0000000E
```

```
R2 0x00000019
```

```
R3 0x00000003
```

```
R4 0x0000009A
```

```
R5 0x00000001
```

```
-x PSR 0x21 00000
```

```
-N 0
```

```
-Z 0
```

```
-C 1
```

```
-V 0
```

```
-Q 0
```

```
-T 1
```

```
-IT Disabled
```

```
-ISR 0
```

Write a program to see address of register stored in address memory plane.

AREA Add, code, READONLY

ENTRY

EXPORT --main

--main

LDR R0, =123

LDR R1, =456

MUL R2, R1, R0

LDR R4, =PROD

STR R2, [R4]

AREA Store, DATA READWRITE

PROD DCD 0x0

END

Output :

In Memory : plane.

Address : R4

0x1000000 : 0000DB18.

R0 : 0x0000007B

R1 : 0x000001C8

R2 : 0x0000DB18.

R4, R2 have same address spaces. (R2 stored in R4).