

ARM7 programming using Keil μ Vision5

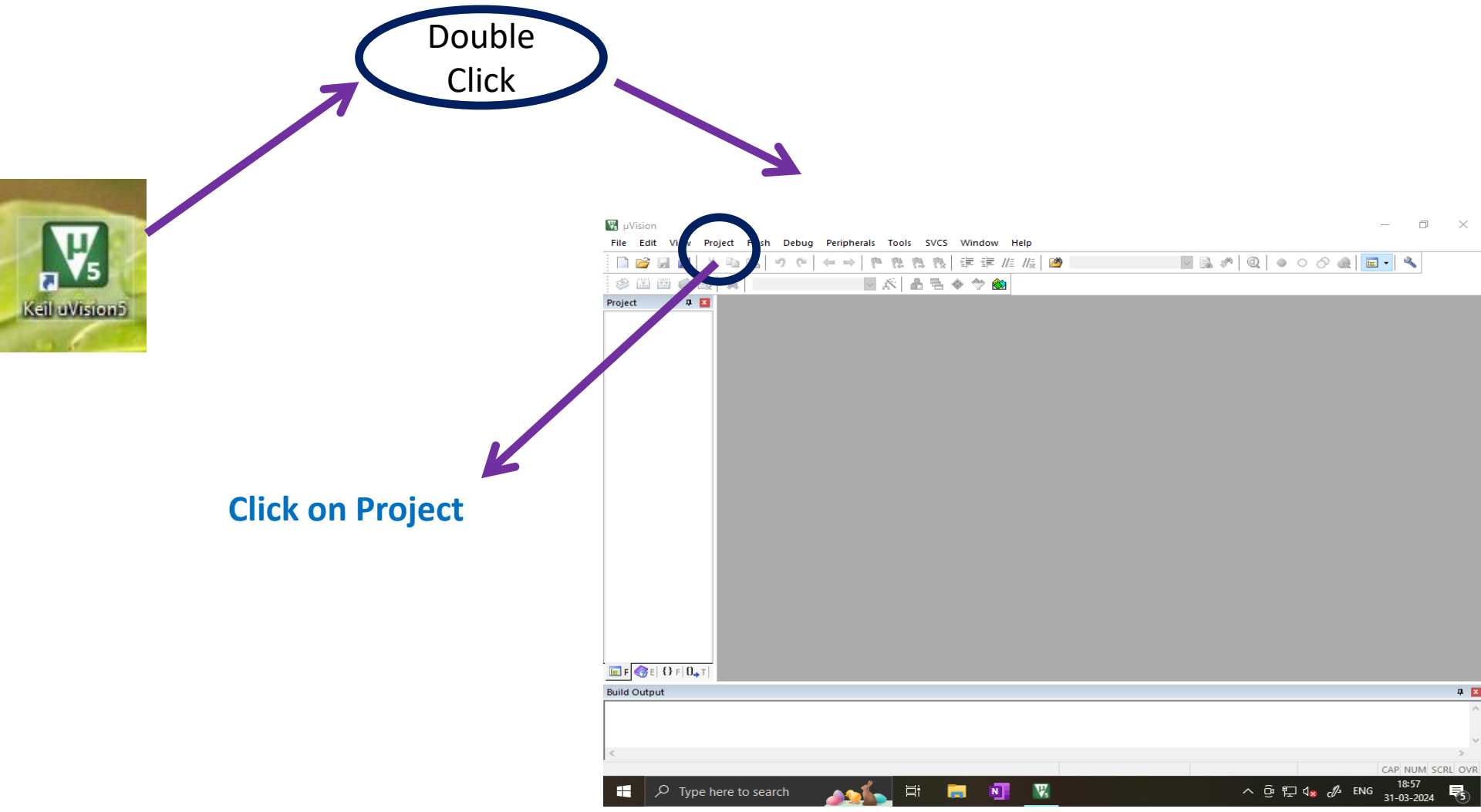
Introduction

- **Keil uVision 5** is a powerful software tool designed for the development of [embedded systems](#), providing users with robust features and optimizations.
- It offers an efficient and streamlined environment for project management and debugging.

**1st create your own folder with the registration Number under ...
folder (document), and then click on Keil μ Vision5**

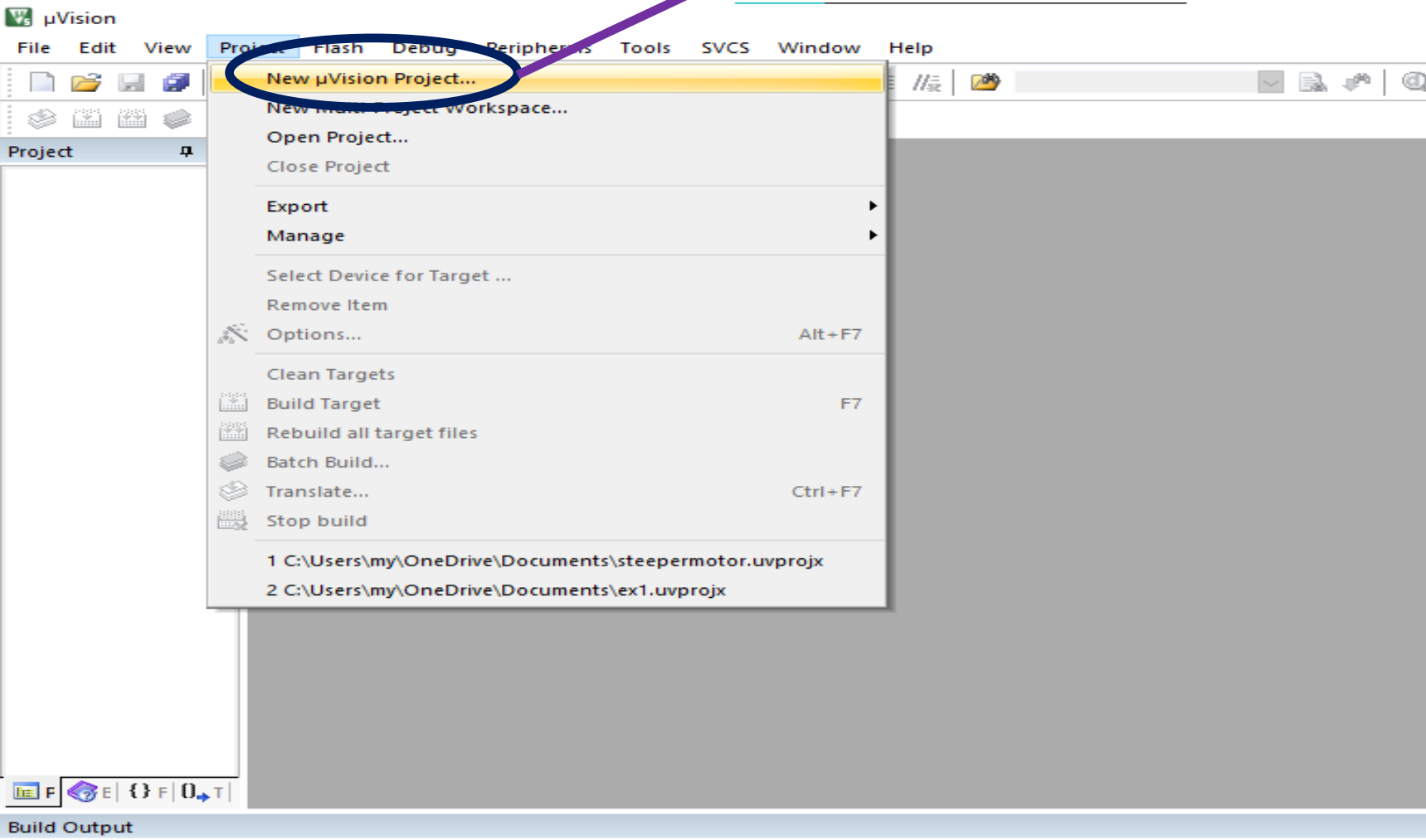


Create a Project



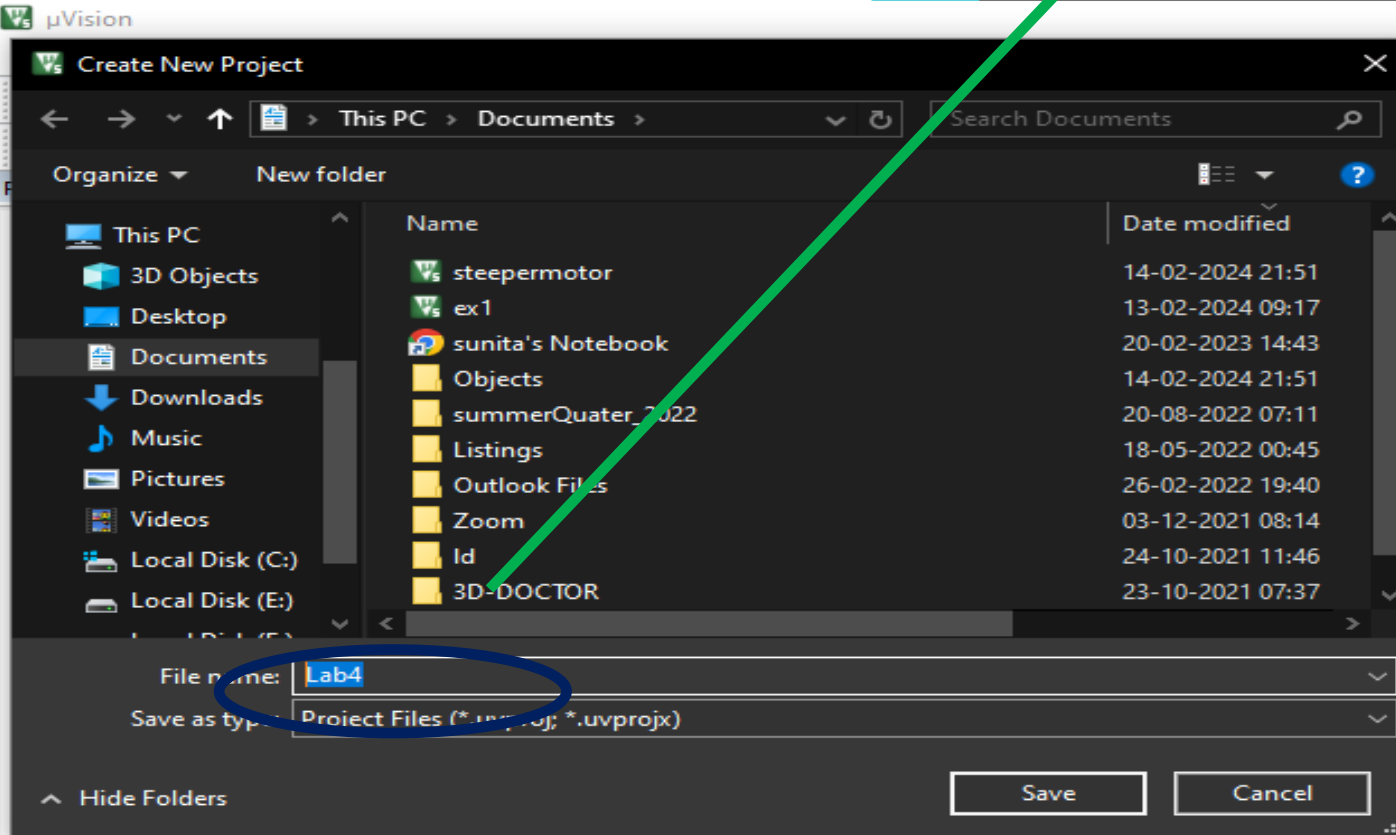
Create a Project

Click on New μ Vision Project



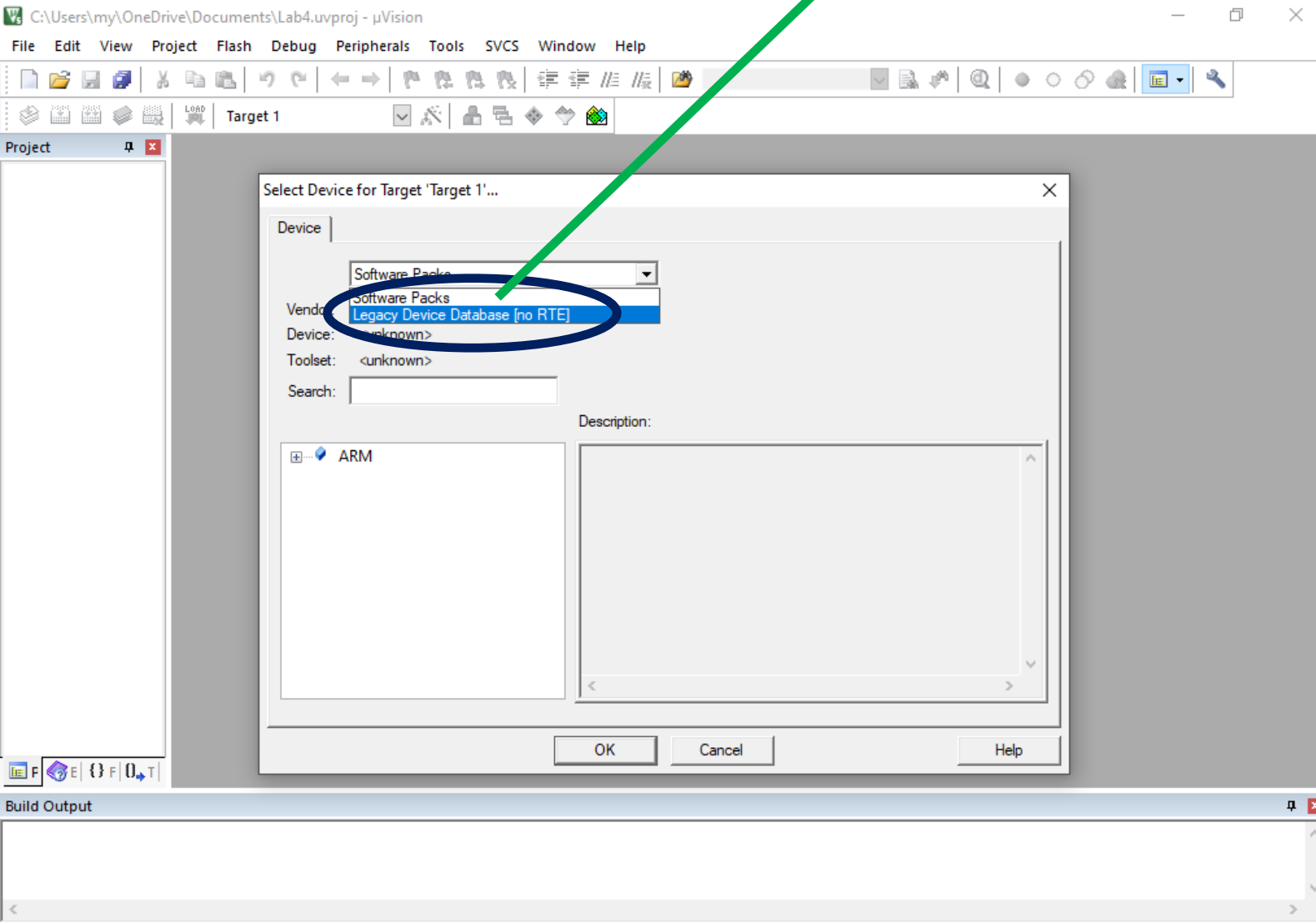
Create a Project

Give your Project name and set the project location to your respective folder



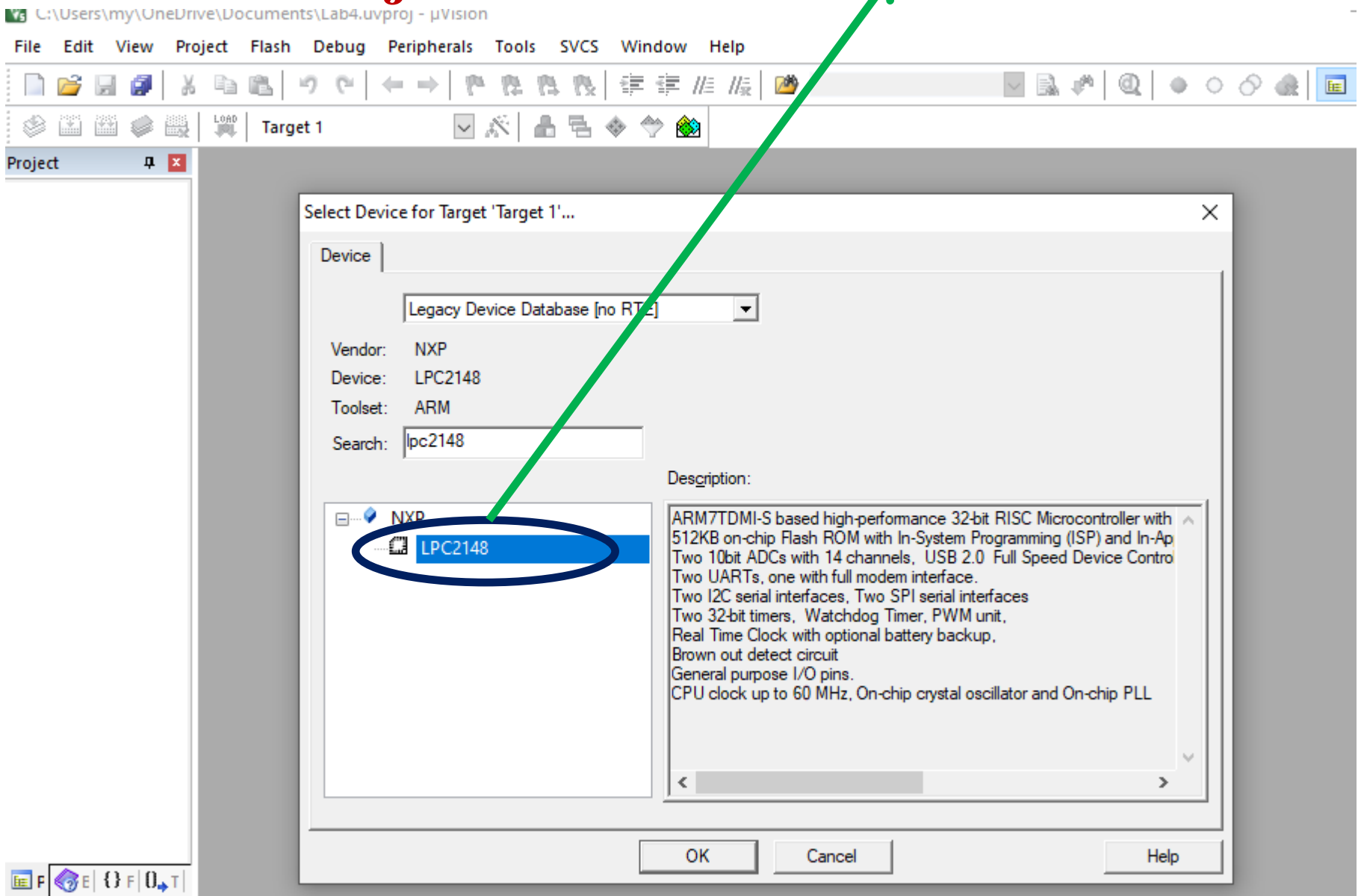
Create a Project

For ARM7, Select the Legacy Device Database for RTE from drop down arrow



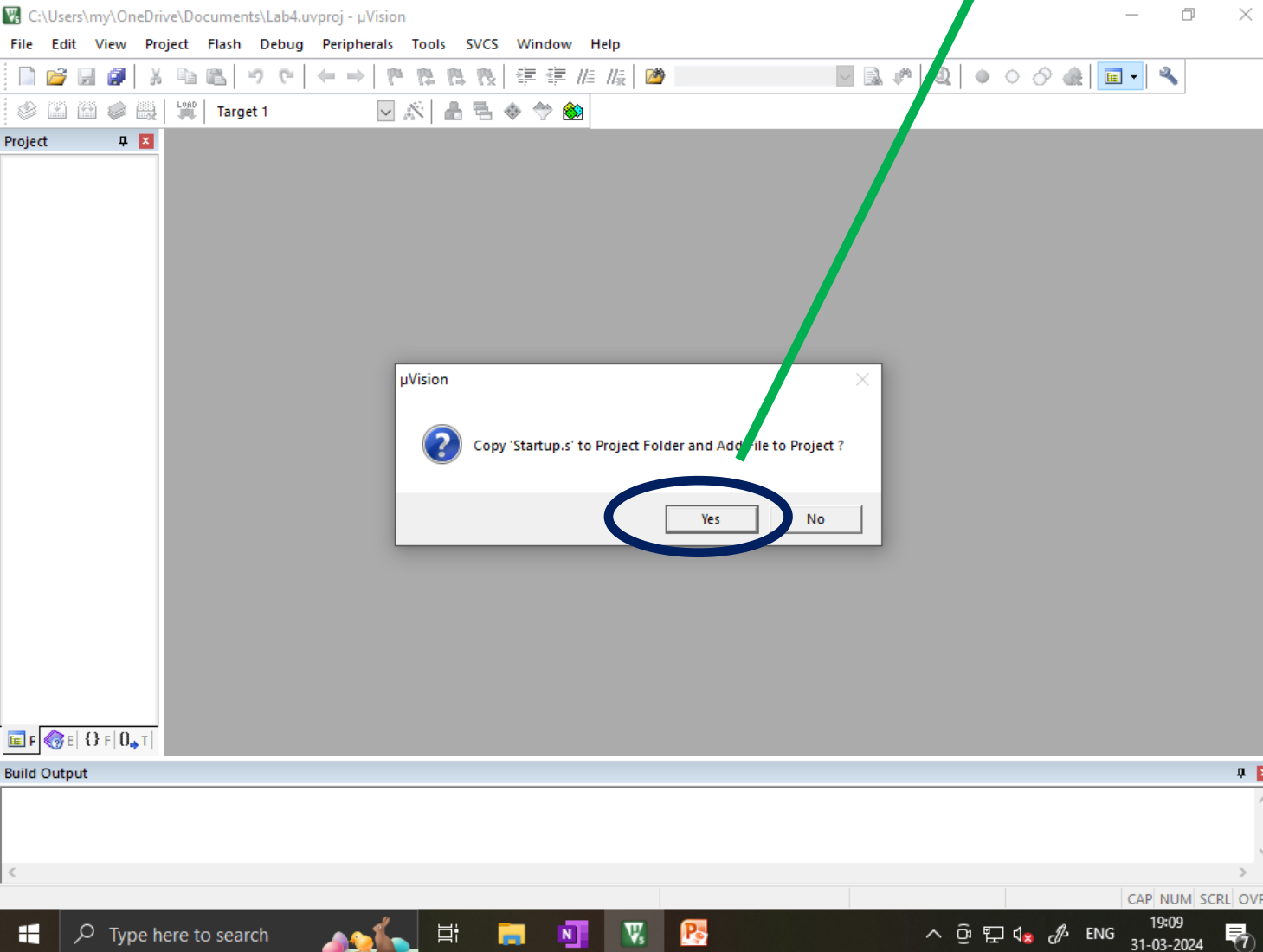
Create a Project

Search the Device **LPC2148** and
Select under Legacy Device Database



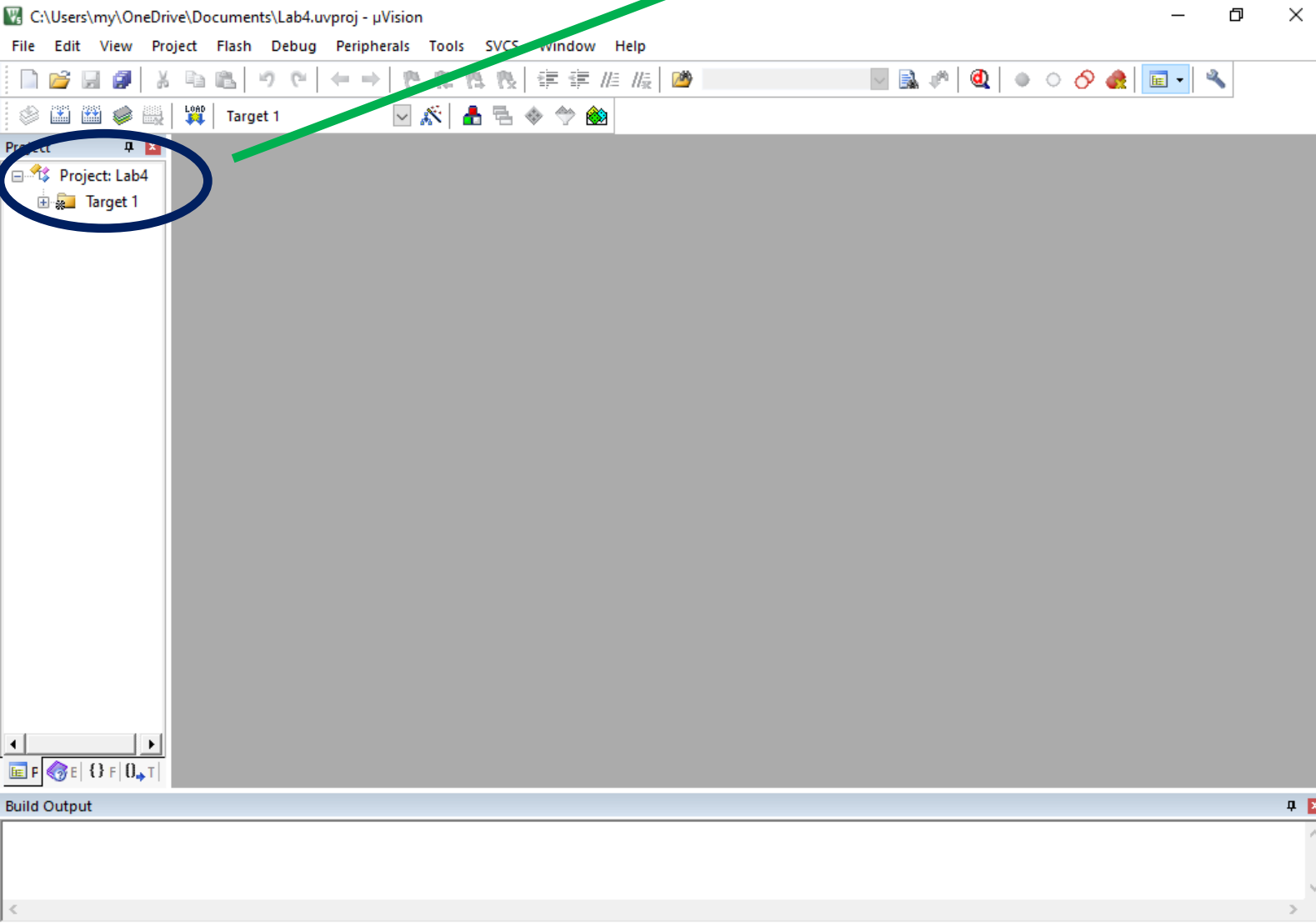
Create a Project

Click on **Yes** to Copy Startup.s file in the project Target



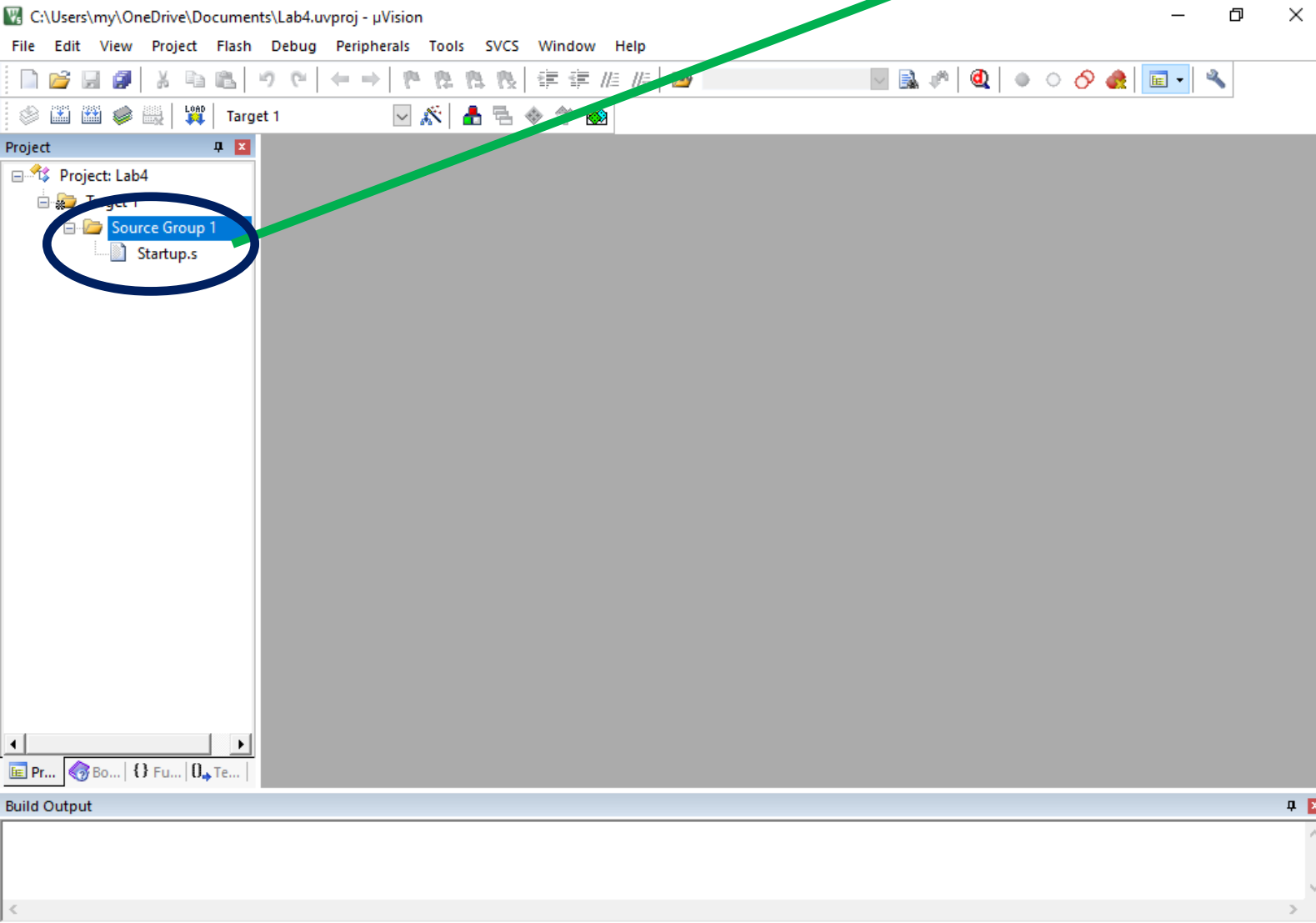
Project has been created

Project created

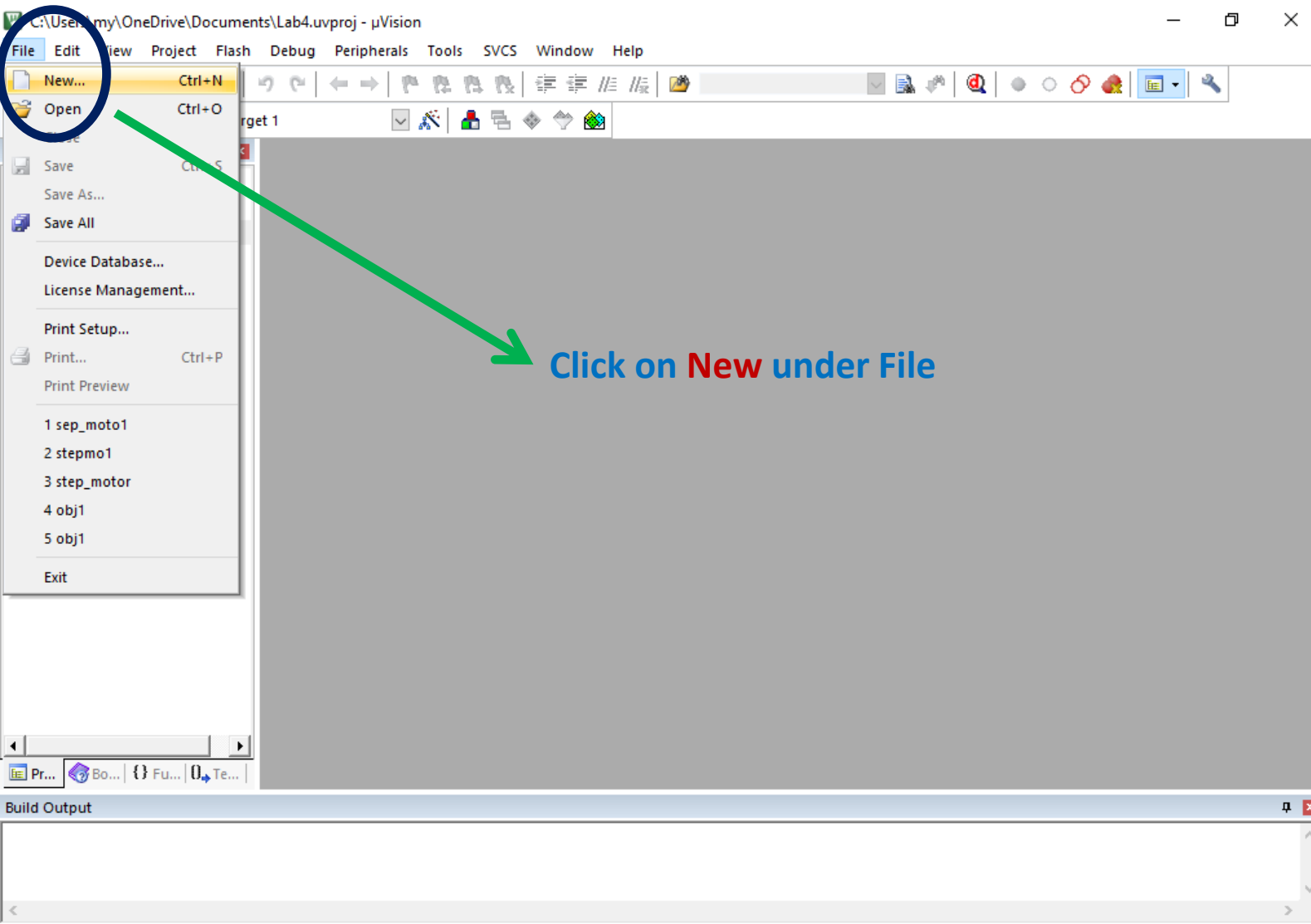


Project created

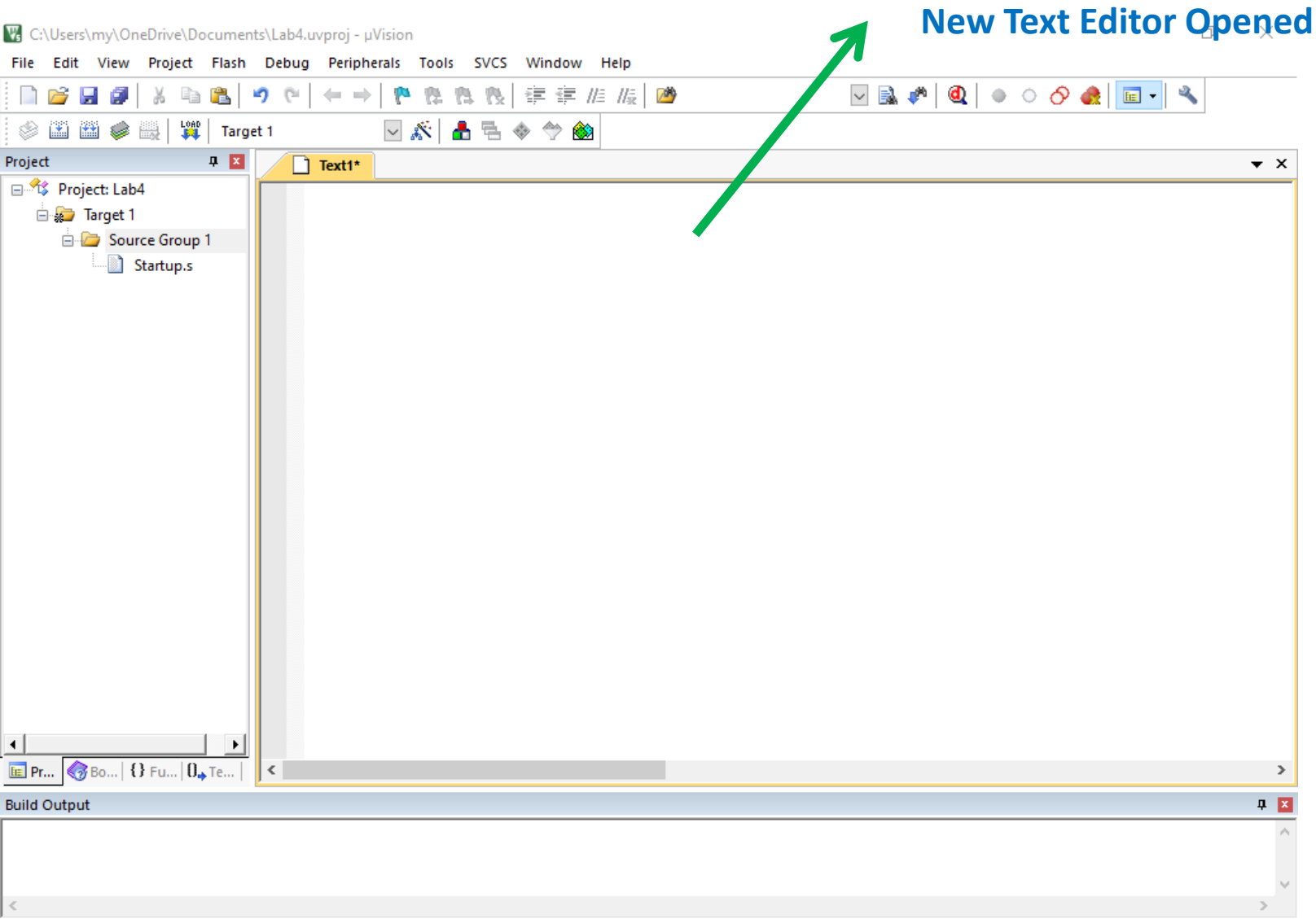
Startup.s has been added



Create a File to write the program

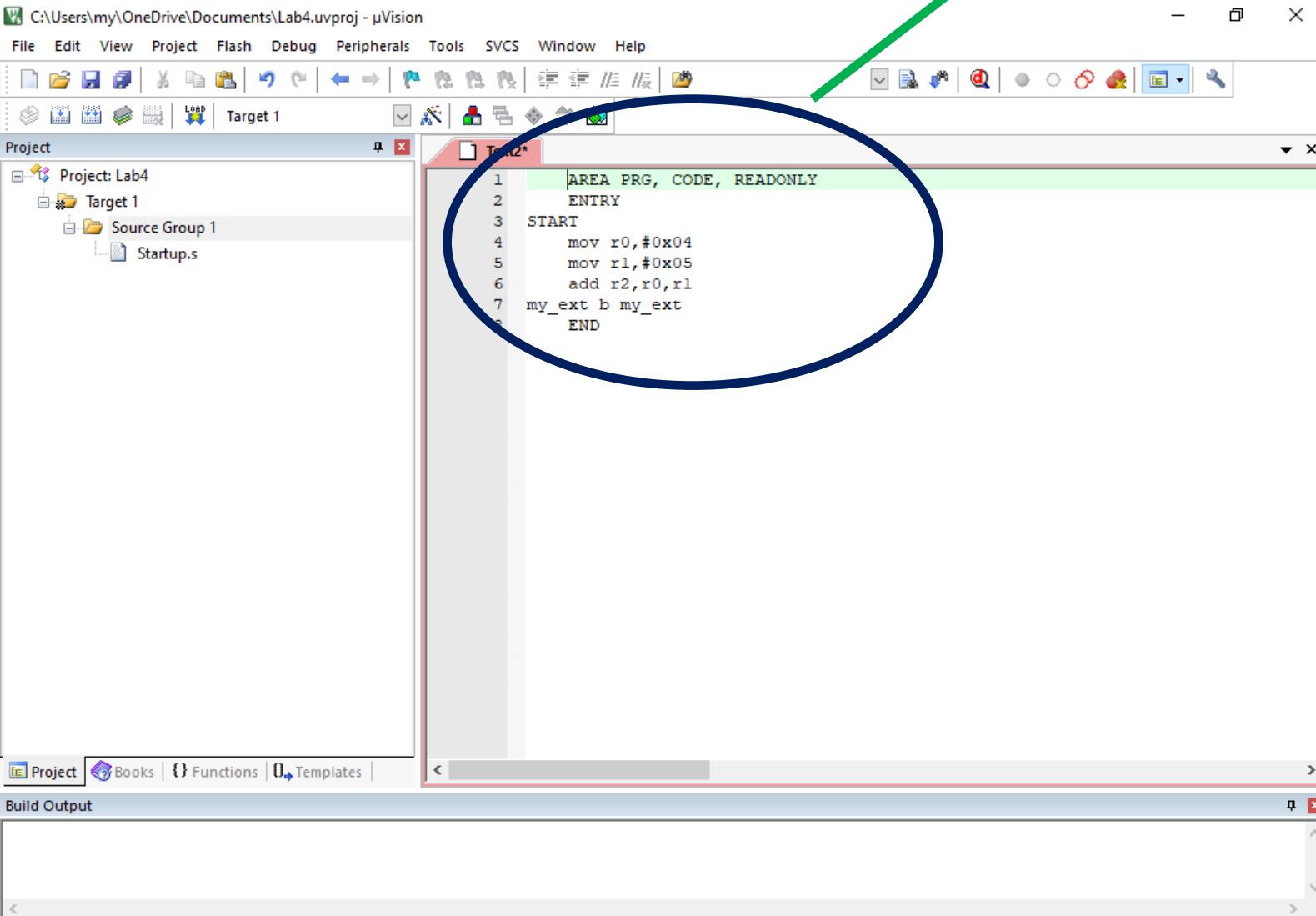


Create a File to write the program



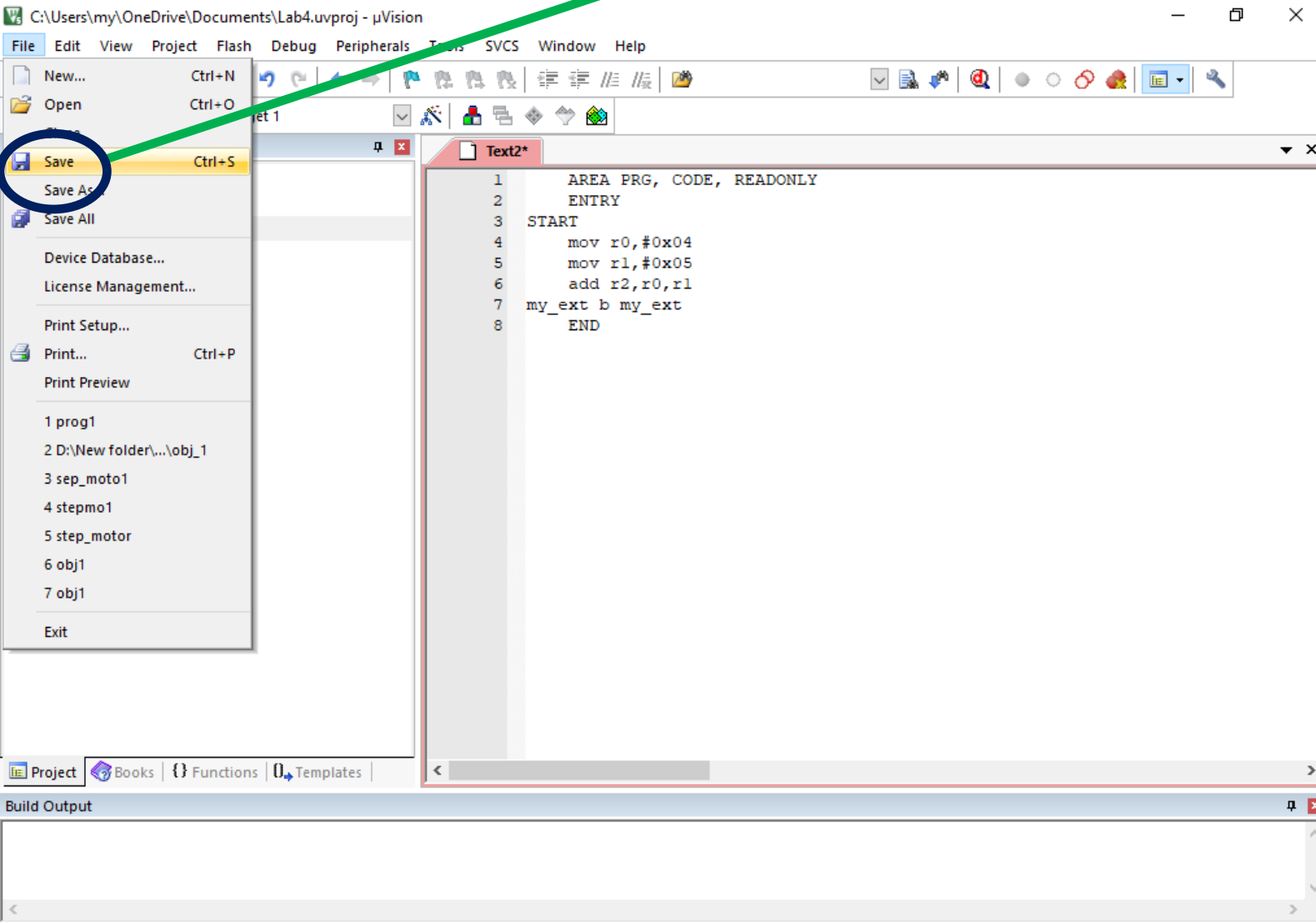
Write the program

Program



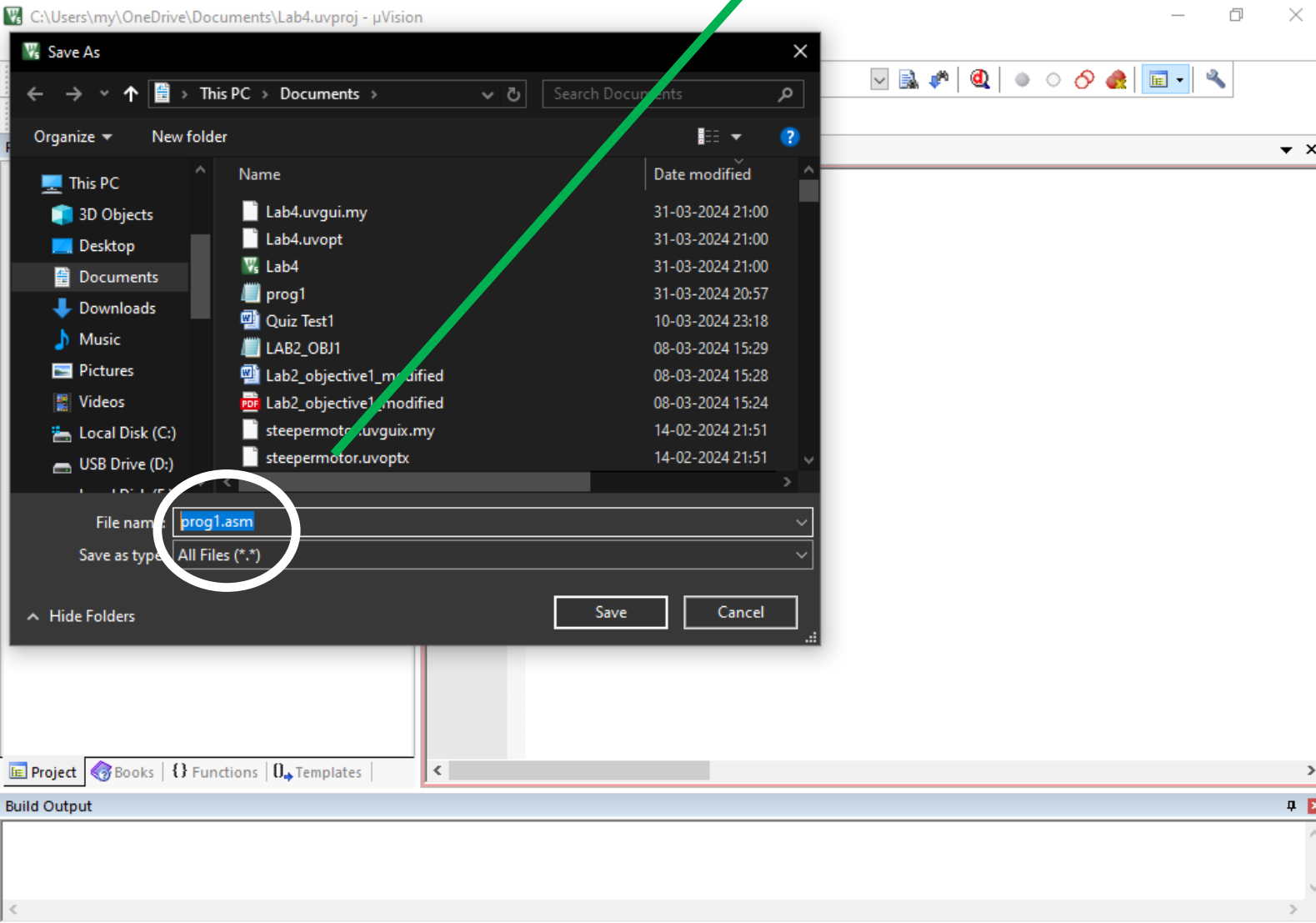
Save the program

Click on **save** under File



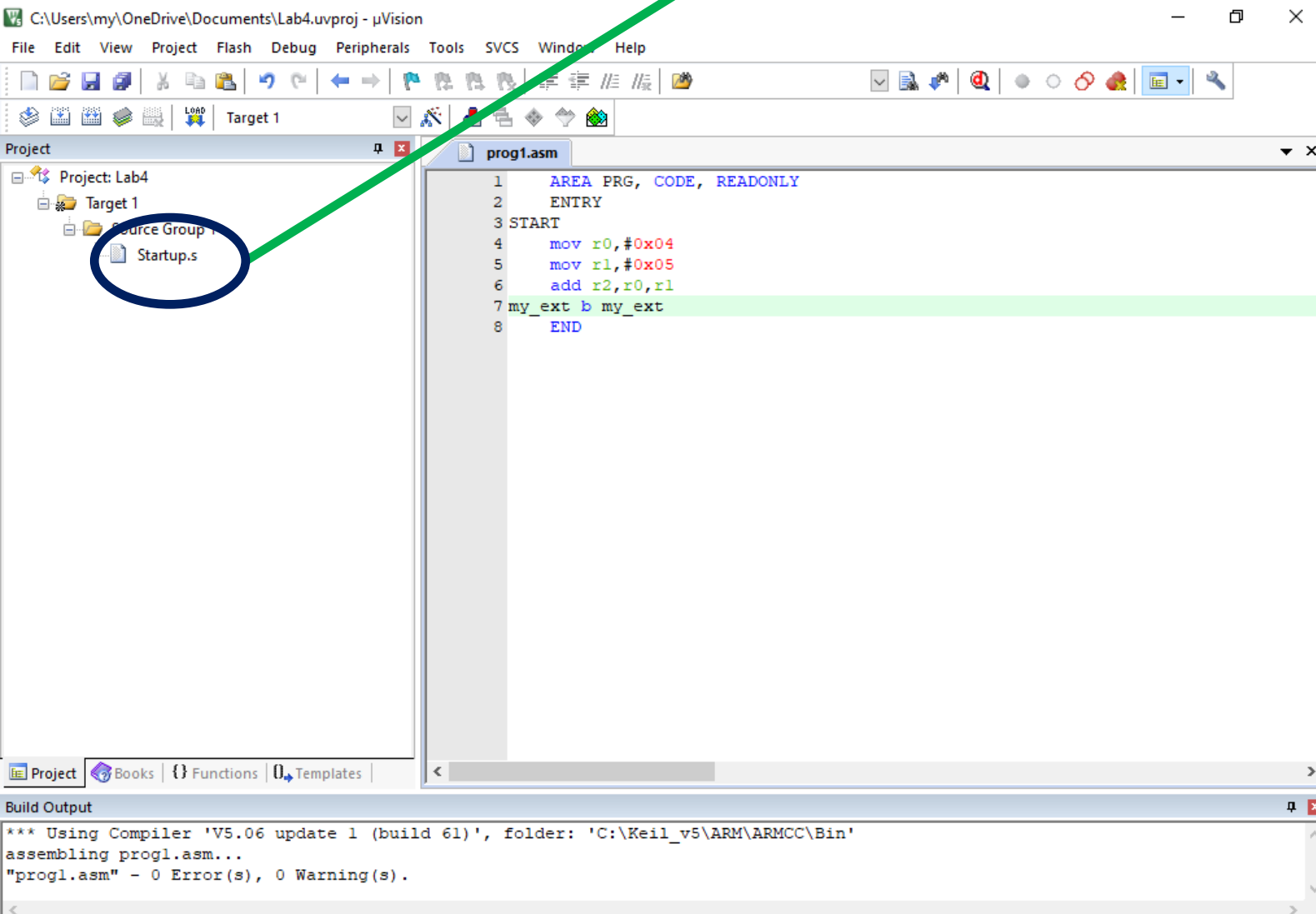
Save the program

Save the program with ... **.asm** in the project folder location



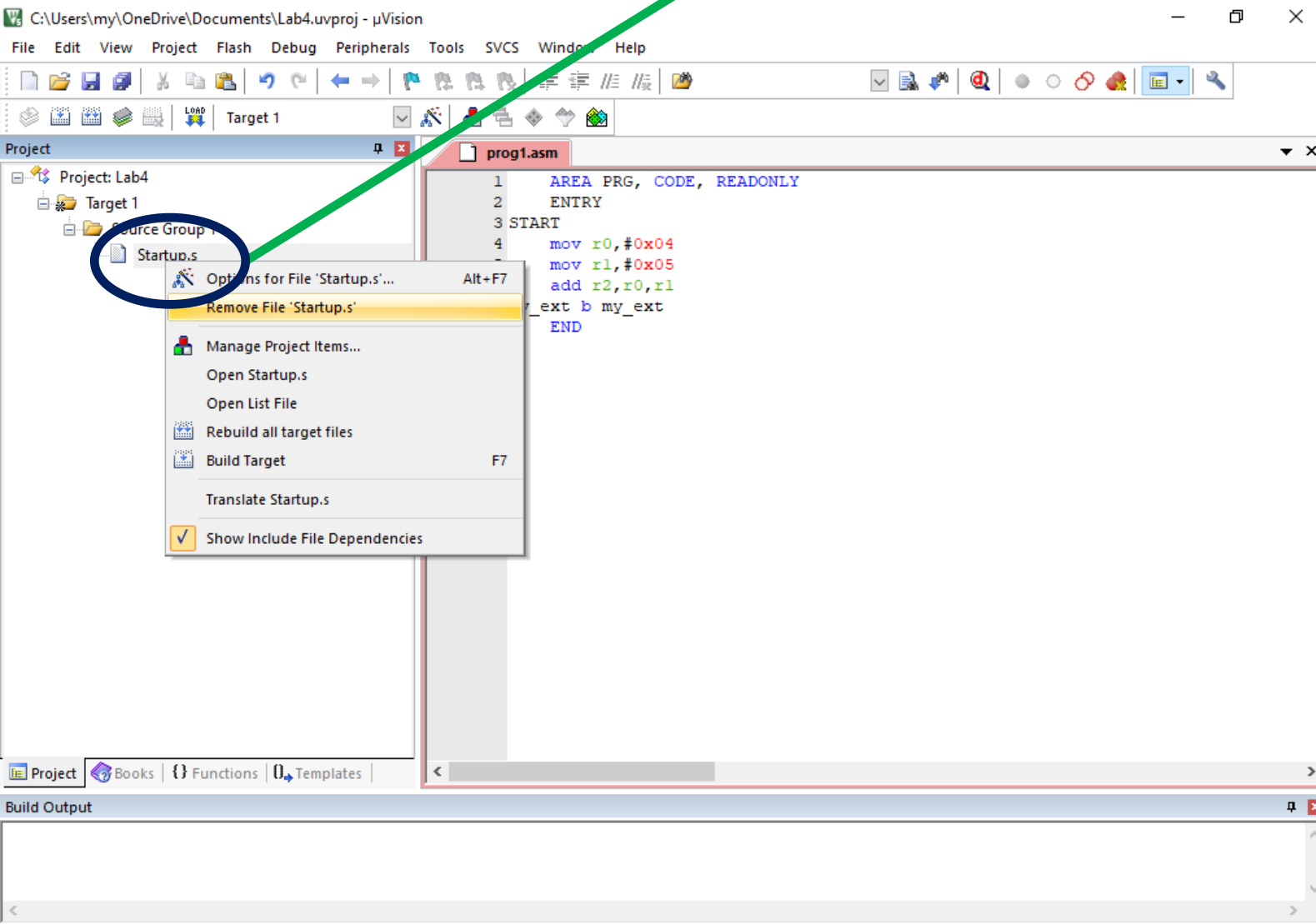
Add the program

First remove the Startup.s file from Source Group

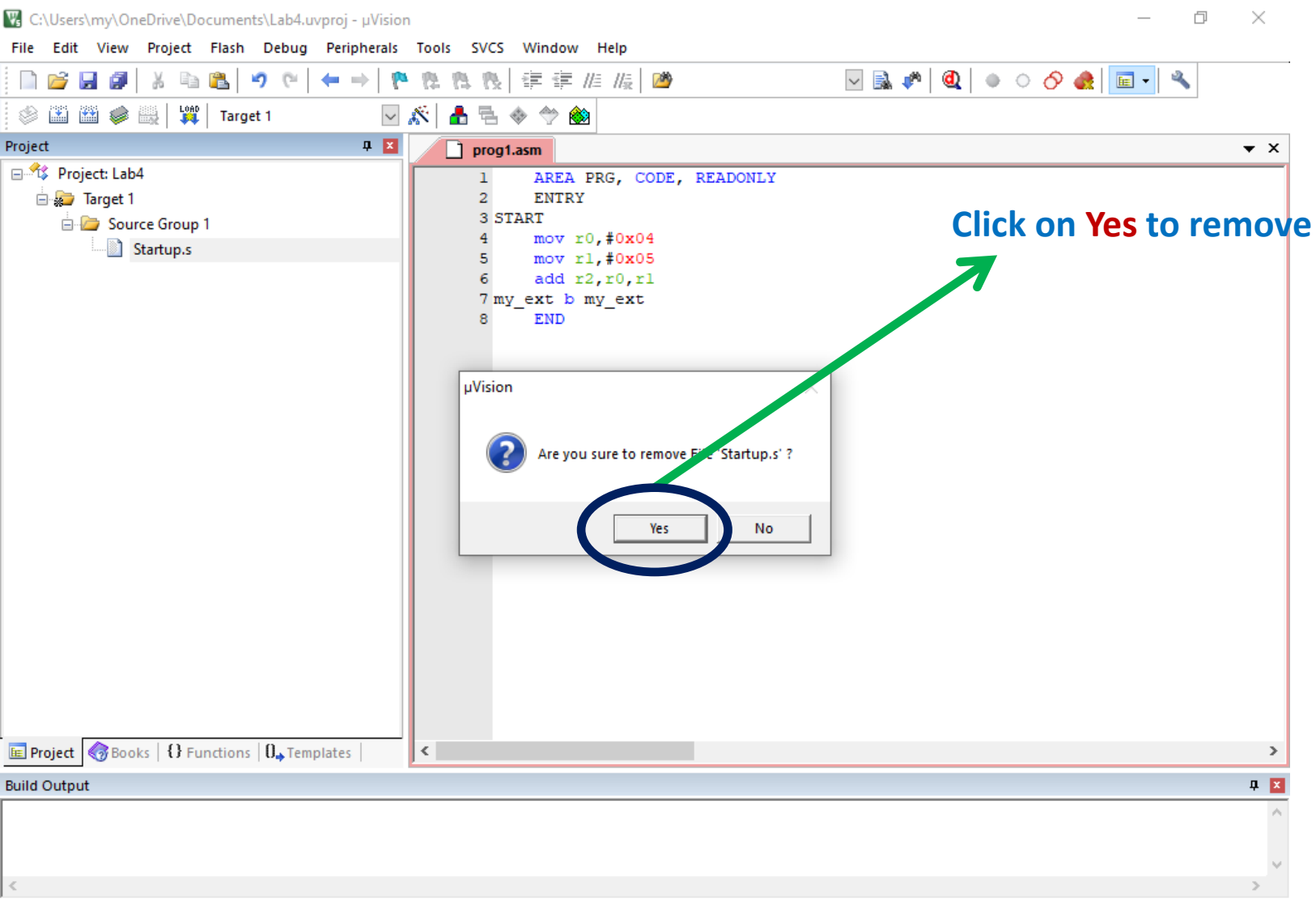


Add the program

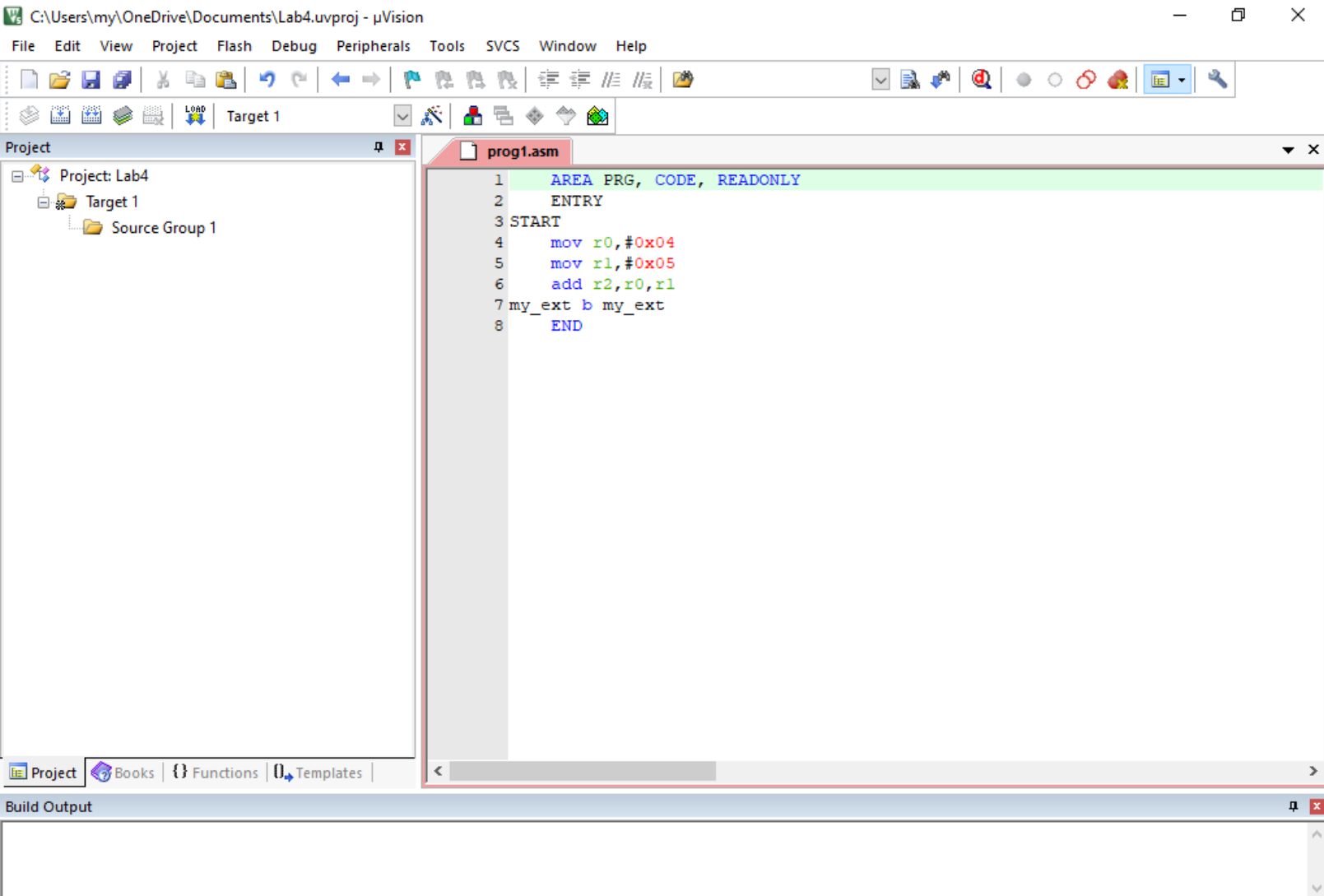
Right click on Startup.s and then
Remove File



Add the program

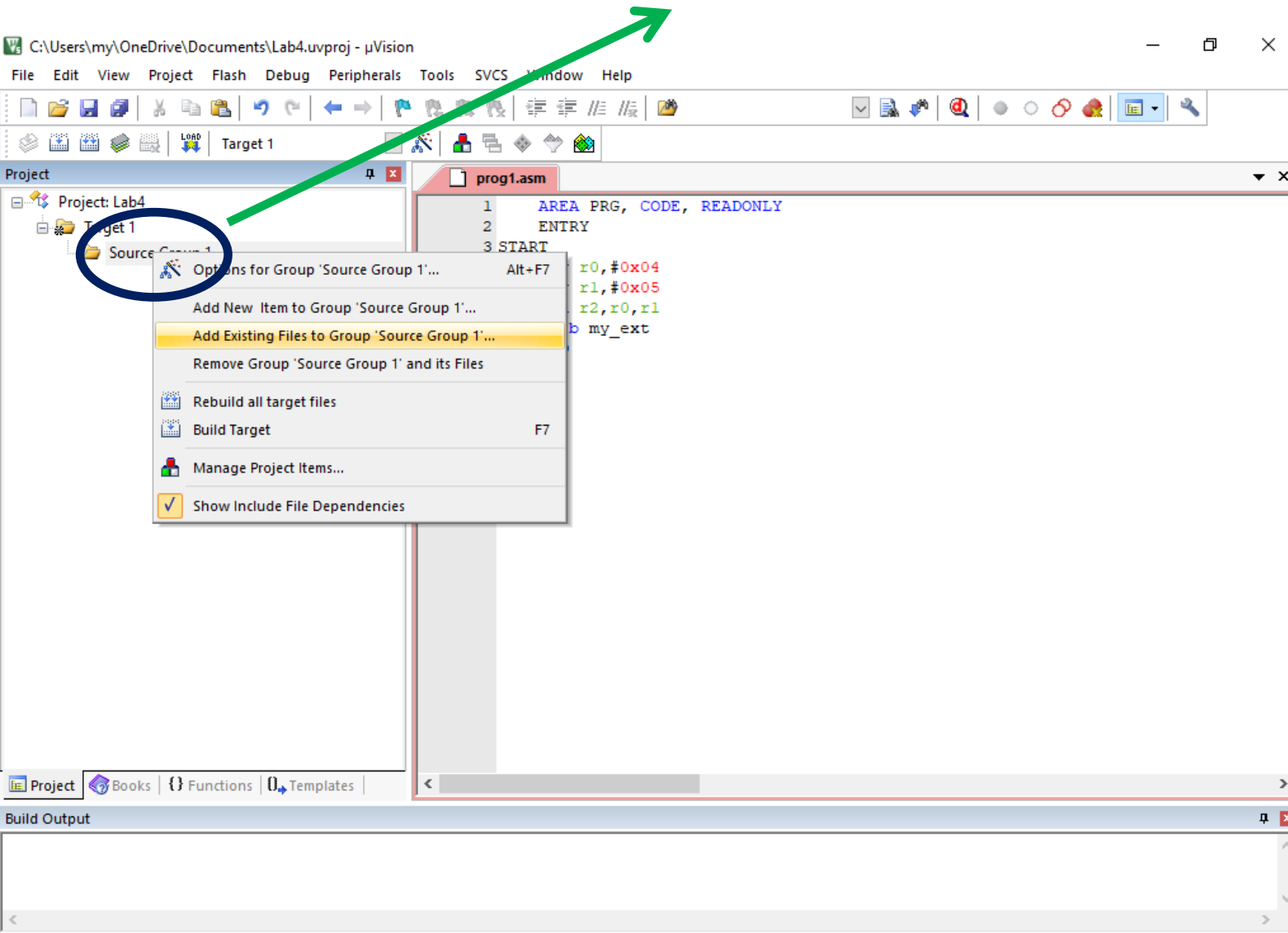


Add the program



Add the program

Right Click on **Source Group1** to add the program



Add the program

The screenshot shows the uVision IDE interface. The main window displays the assembly code for 'prog1.asm':

```
1 AREA PRG, CODE, READONLY
2 ENTRY
3 START
4 mov r0, #0x04
5 mov r1, #0x05
6 add r2, r0, r1
7 my_ext b my_ext
8
```

The 'Add Files to Group' dialog box is open, showing the 'Documents' folder. The 'Files of type' dropdown is open, and 'Asm Source file (*.s; *.src; *.a)' is selected. A green arrow points from the text 'Select the File Type as .s,..., .a' to the selected option.

Project: Lab4
Target 1
Source Group 1

Build Output

Add the program

The screenshot shows the µVision IDE interface. The main window displays the assembly code for 'prog1.asm':

```
1 AREA PRG, CODE, READONLY
2 ENTRY
3 START
4 mov r0, #0x04
5 mov r1, #0x05
6 add r2, r0, r1
7 my_ext b my_ext
8
```

The 'Project' window on the left shows the project structure:

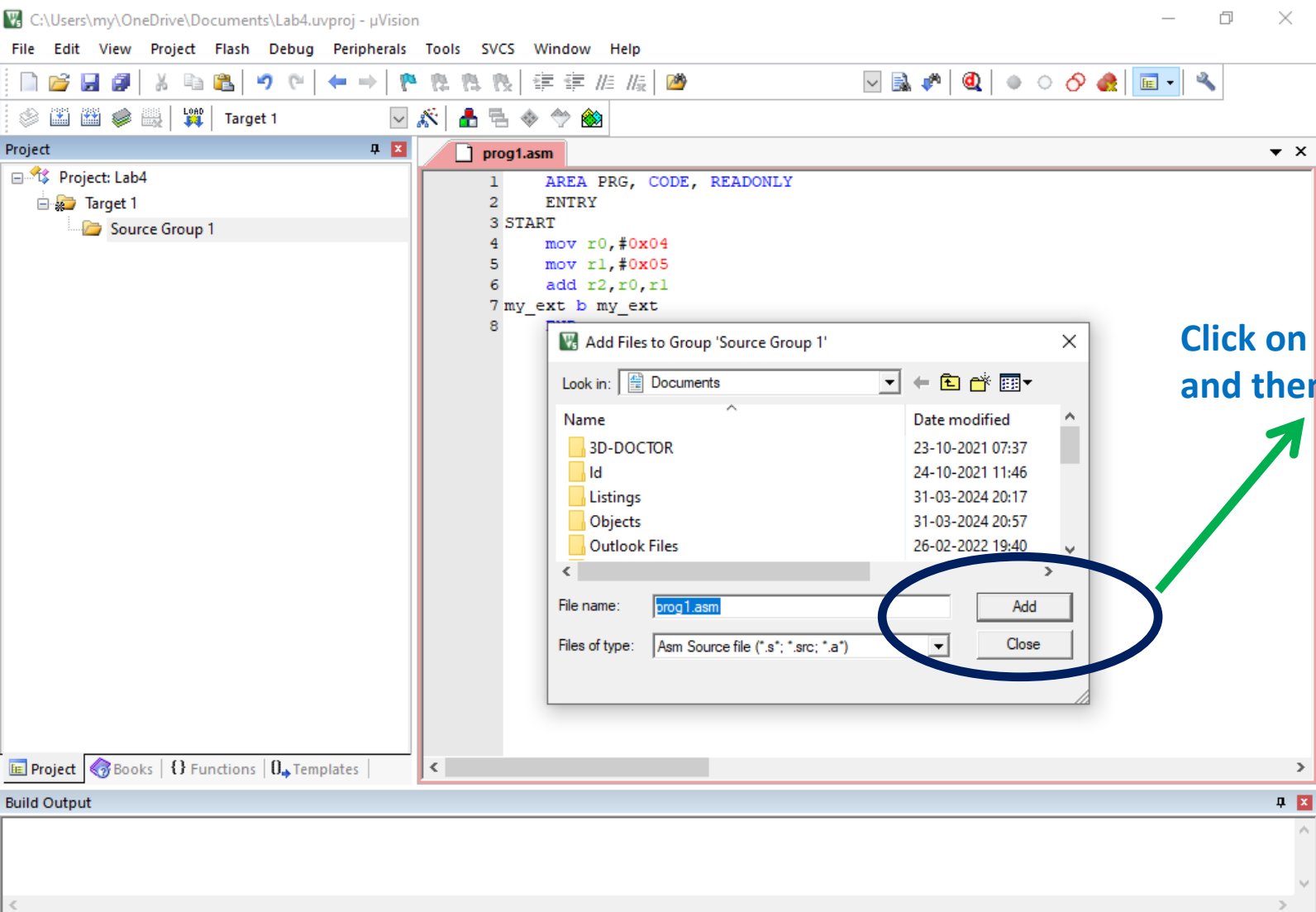
- Project: Lab4
 - Target 1
 - Source Group 1

The 'Add Files to Group' dialog is open, showing the 'Documents' folder. The file list includes:

Name	Date modified
3D-DOCTOR	23-10-2021 07:37
Id	24-10-2021 11:46
Listings	31-03-2024 20:17
Objects	31-03-2024 20:17
Outlook Files	26-02-2022 19:40

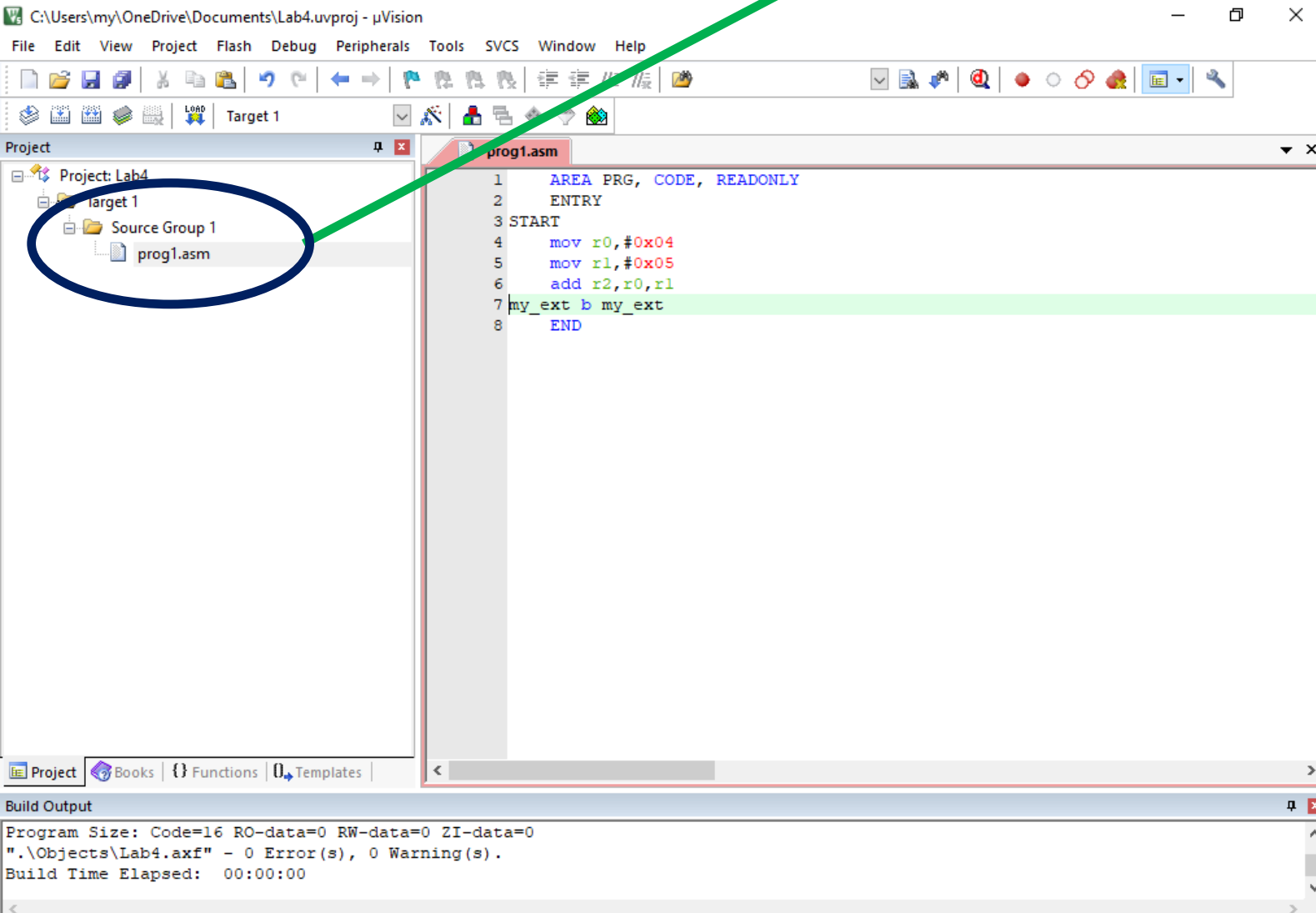
The file 'prog1.asm' is selected in the list. A green arrow points from the text 'Select the program from the Your Folder' to the selected file.

Add the program

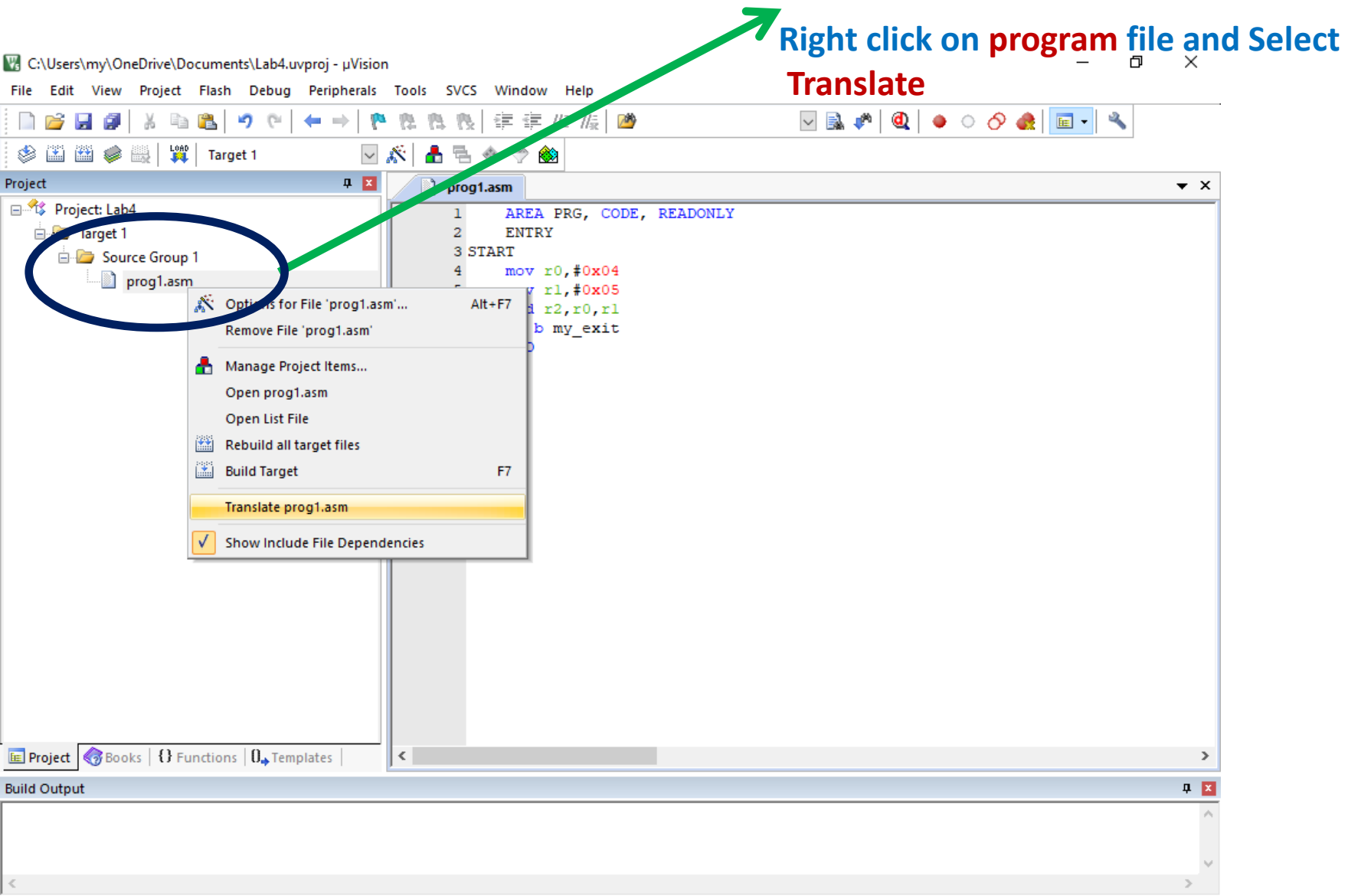


Add the program

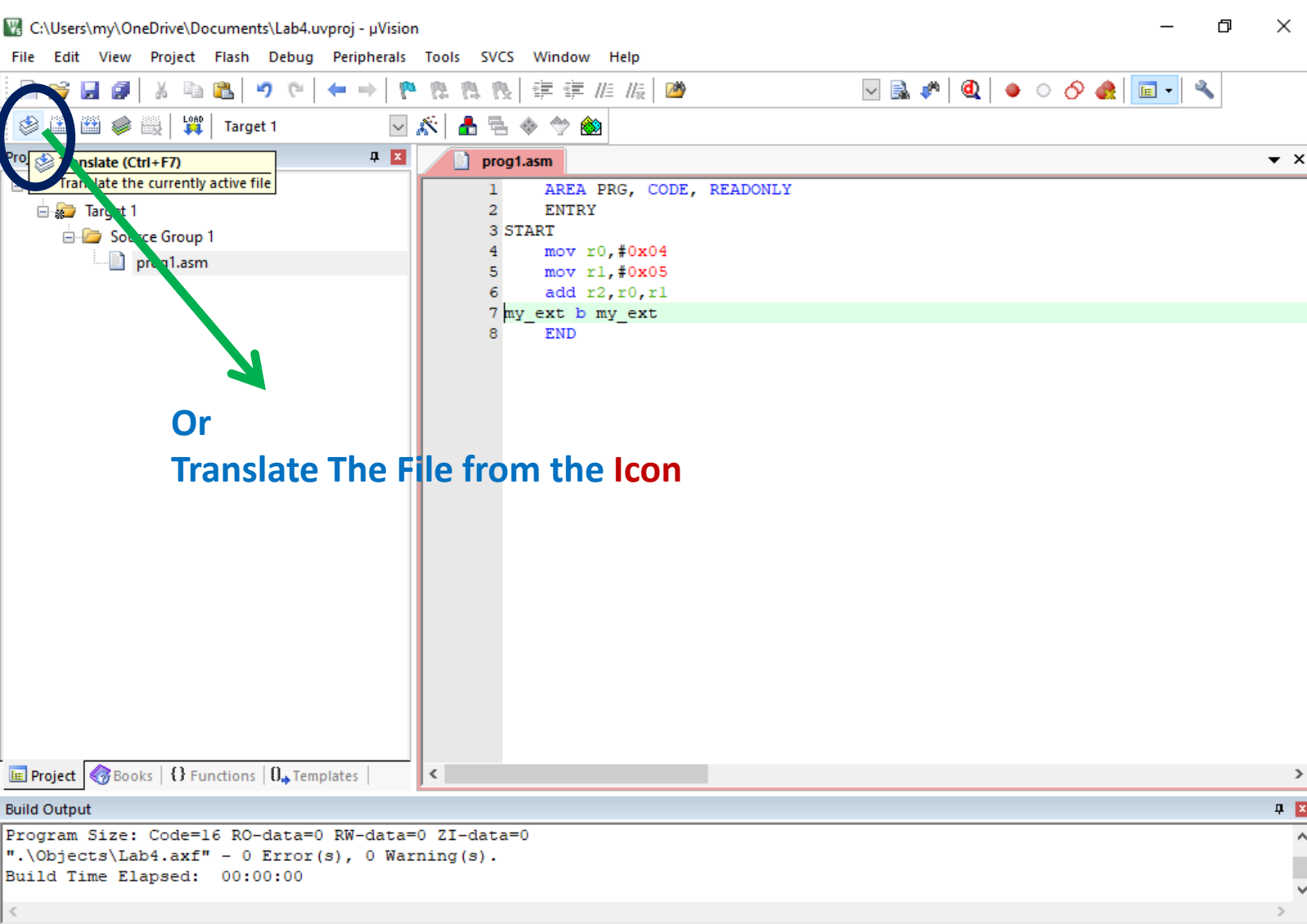
Program has been added under Source Group1



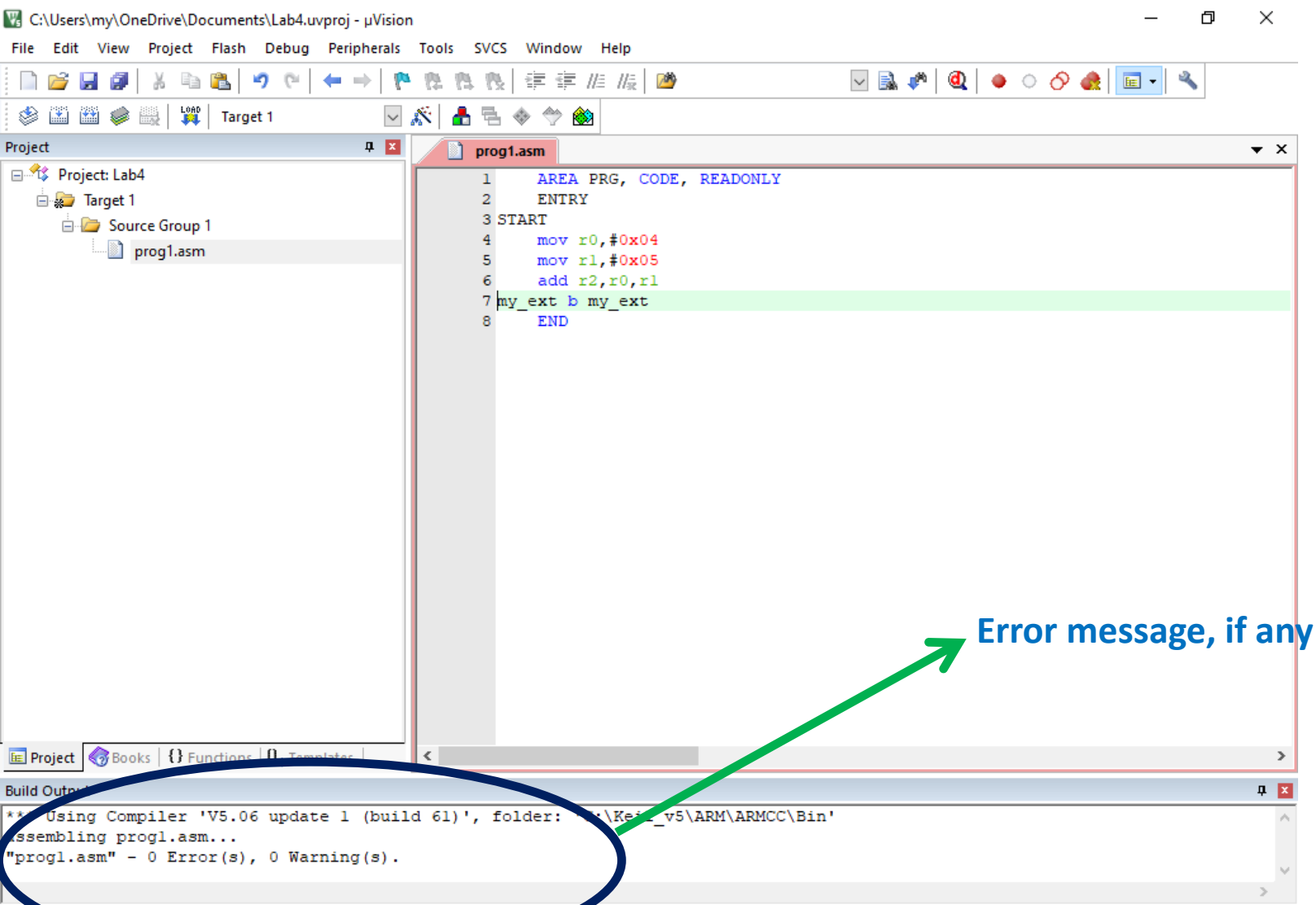
Compile the program: 1. Translate the program



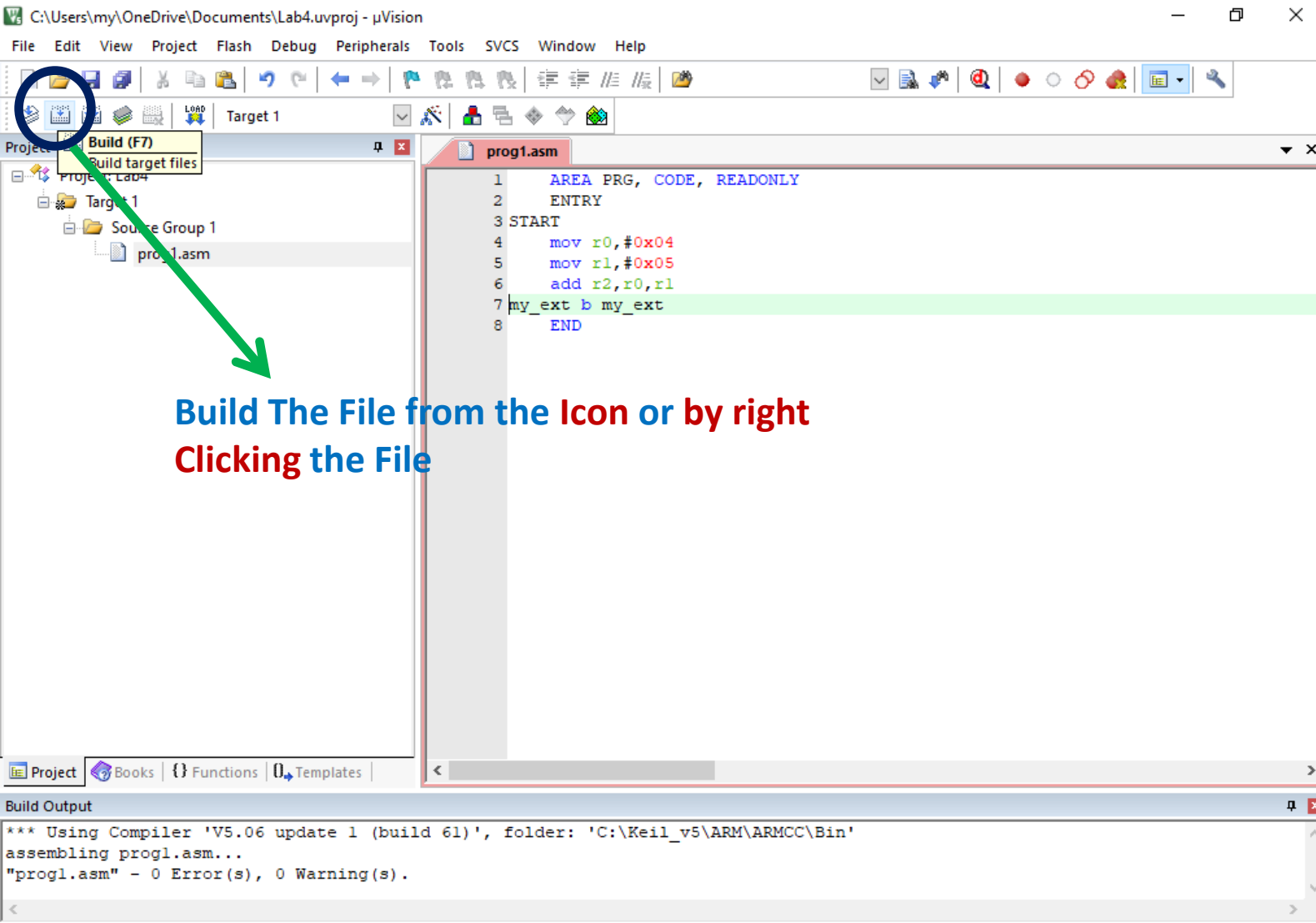
Compile the program: 1. Translate the program



Compile the program: 1. Translate the program

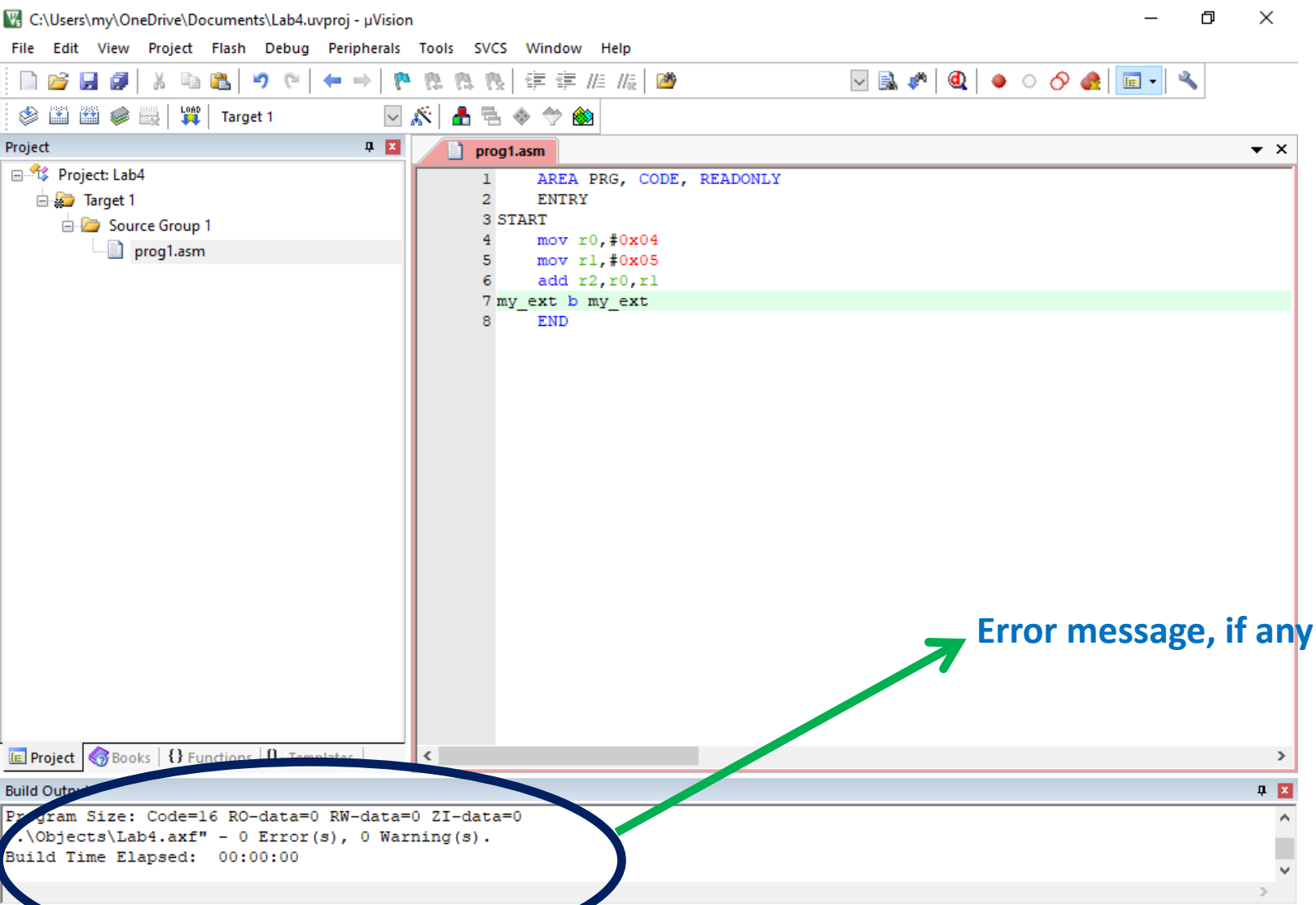


Compile the program: 2. Build the program to create Target

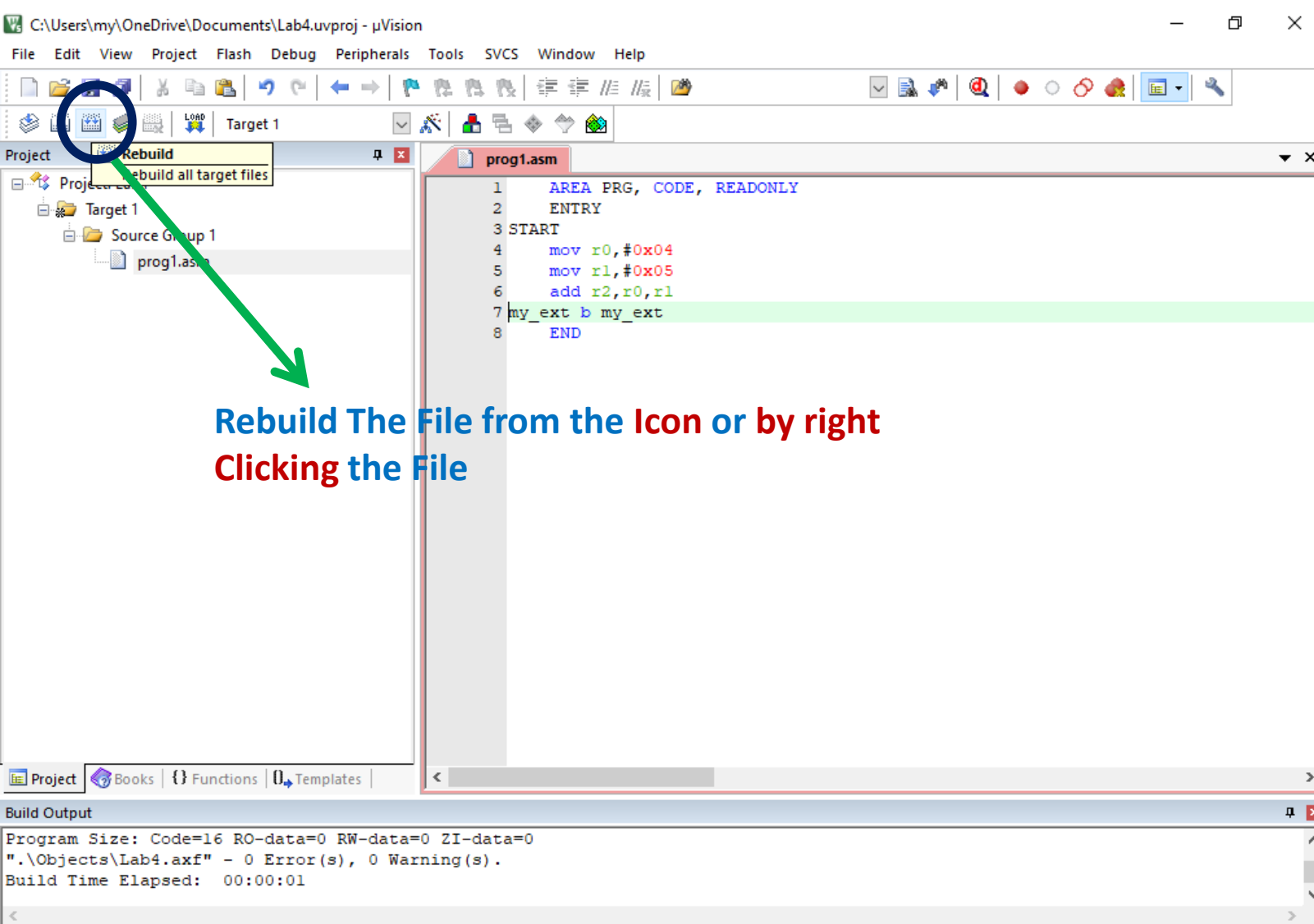


Build The File from the Icon or by right
Clicking the File

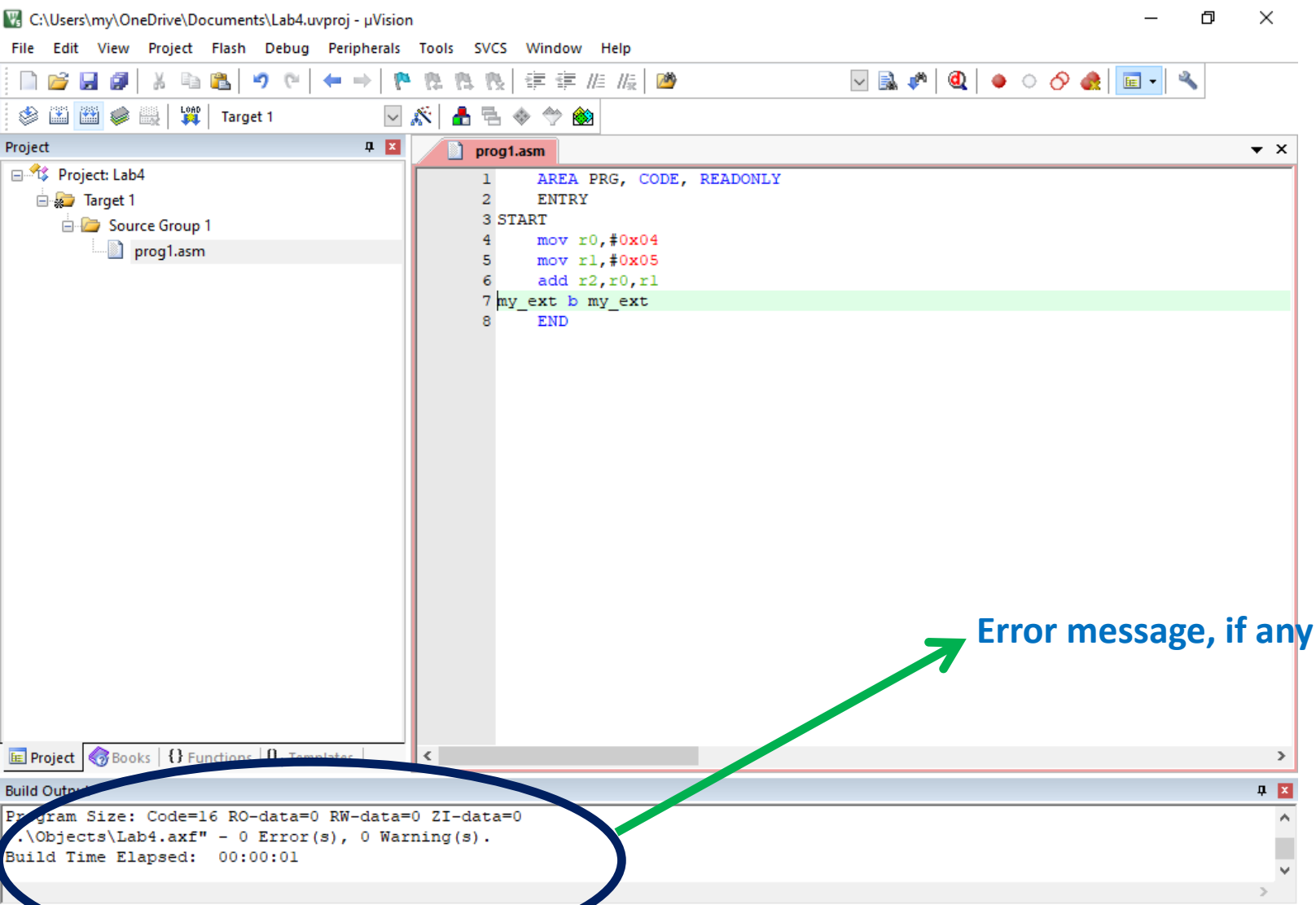
Compile the program: 2. Build the program to create Target



Compile the program: 3. Rebuild the program to create Target



Compile the program: 3. Rebuild the program to create Target



Execute the program: 1. Start Debug Session

Click on Start/Stop Debug session

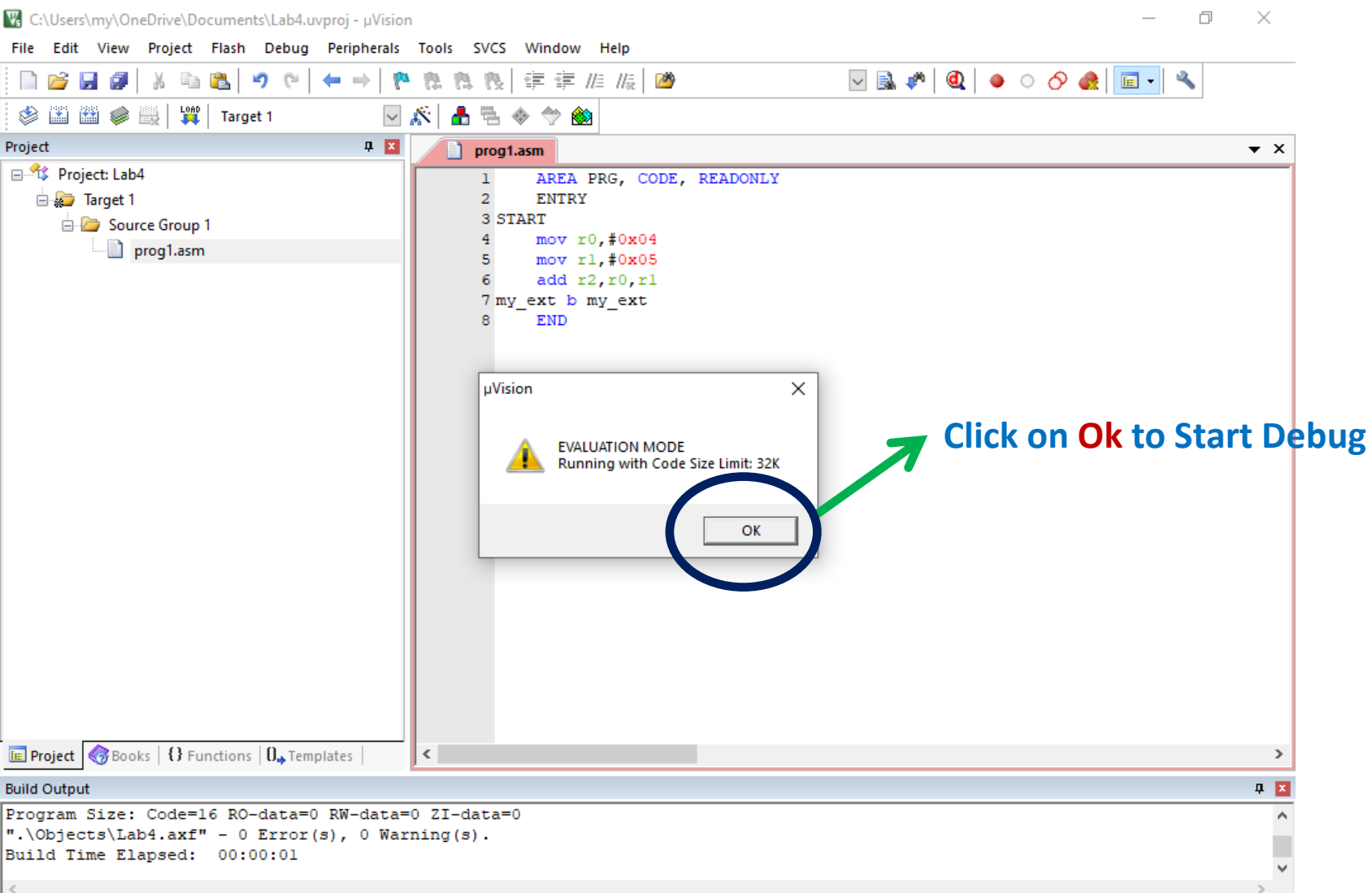
Start/Stop Debug Session (Ctrl+F5)
Enter or leave a debug session

```
1 AREA PRG, CODE, READONLY
2 ENTRY
3 START
4 mov r0, #0x04
5 mov r1, #0x05
6 add r2, r0, r1
7 my_ext b my_ext
8 END
```

Project: Lab4
Target 1
Source Group 1
prog1.asm

Build Output
Program Size: Code=16 RO-data=0 RW-data=0 ZI-data=0
".\\Objects\\Lab4.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed: 00:00:01

Execute the program: 1. Start Debug Session



Execute the program: 1. Start Debug Session

Register Window

Disassembly Window

The screenshot displays the uVision IDE interface with three windows highlighted by green arrows:

- Register Window:** Located on the left, it shows the current state of registers R0 through R15, CPSR, and SPSR. All registers are at 0x00000000 except for CPSR, which is 0x000000D3.
- Disassembly Window:** Located in the center, it shows the disassembled code for the program. The first instruction, `MOV R0, #0x00000004`, is highlighted in yellow.
- Memory Window:** Located at the bottom right, it shows the memory address field and a lock icon.

The main window displays the assembly code for `prog2.asm`:

```
1 AREA PRG, CODE, READONLY
2 ENTRY
3 START
4 mov r0, #0x04
5 mov r1, #0x05
6 add r2, r0, r1
7 my_ext b my_ext
8 END
```

The Command window at the bottom left shows the following text:

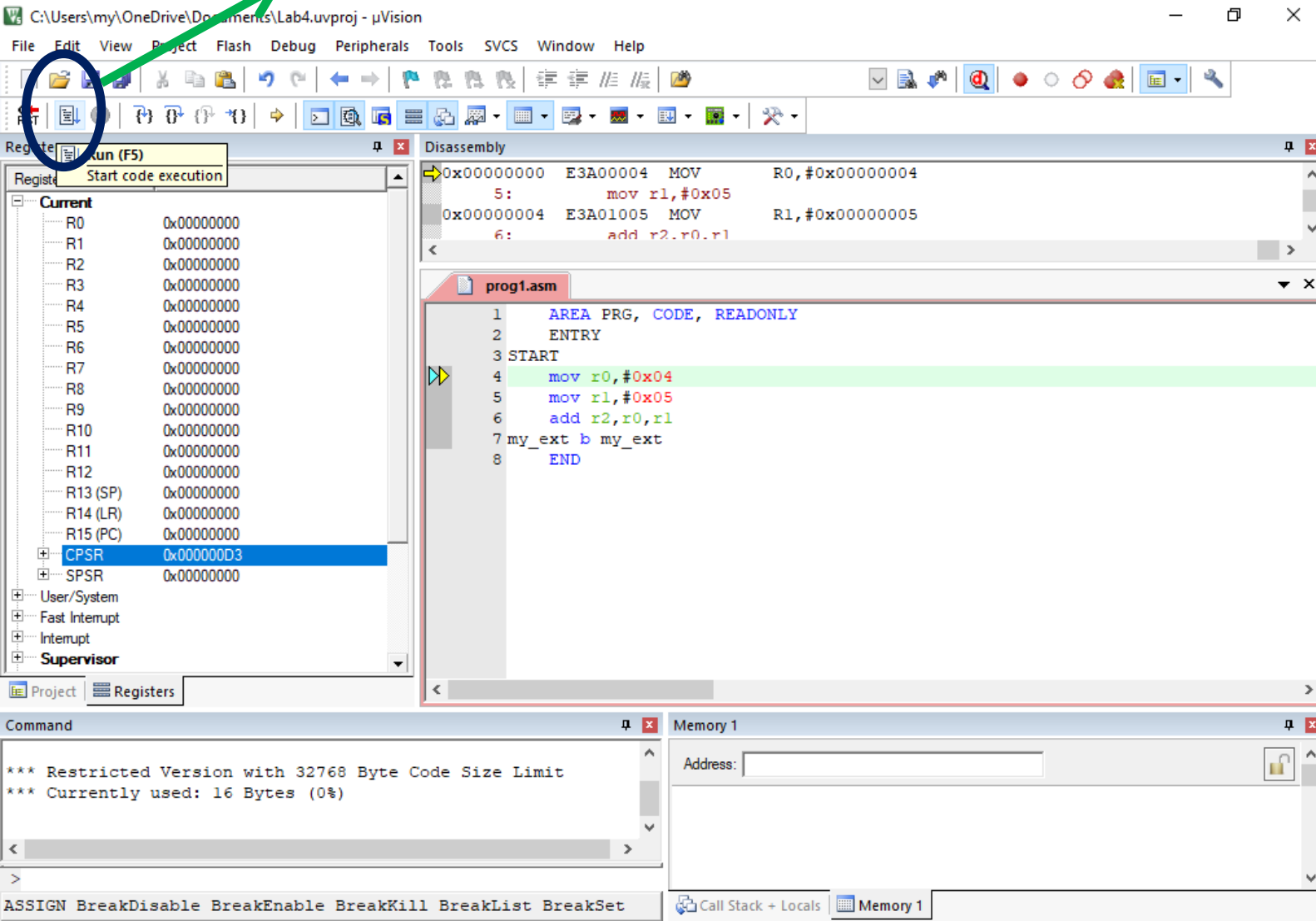
```
*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 16 Bytes (0%)
```

The bottom status bar shows the following text:

```
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet
```

Execute the program: 2. Run

Click on Run icon



The screenshot shows the uVision IDE interface. A green arrow points to the 'Run' button (a green play icon) in the top toolbar. The 'Run' button is also highlighted in the 'Project' menu. The 'Registers' window on the left shows the current state of the registers, with the 'CPSR' register highlighted. The 'Disassembly' window shows the assembly code for the program, with the first instruction highlighted. The 'prog1.asm' window shows the source code, with the first instruction highlighted. The 'Command' window at the bottom shows the output of the program, indicating a restricted version with a 32768 byte code size limit.

Registers (F5)

Register	Value
R0	0x00000000
R1	0x00000000
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000000
CPSR	0x000000D3
SPSR	0x00000000

Disassembly

Address	Instruction	Comment
0x00000000	E3A00004	MOV R0, #0x00000004
5:	mov r1, #0x05	
0x00000004	E3A01005	MOV R1, #0x00000005
6:	add r2, r0, r1	

prog1.asm

```
1 AREA PRG, CODE, READONLY
2 ENTRY
3 START
4 mov r0, #0x04
5 mov r1, #0x05
6 add r2, r0, r1
7 my_ext b my_ext
8 END
```

Command

```
*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 16 Bytes (0%)
```

Memory 1

Address:

ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet

Execute the program: 3. Stop the Run to check the result

Click on Stop icon

The screenshot shows the uVision IDE interface with the following components:

- Registers:** A list of registers (R0-R15, CPSR, SPSR) with their current values. CPSR is highlighted.
- Disassembly:** A window showing the disassembled code. The current instruction is `0x00000004 E3A00004 MOV R0, #0x00000004`.
- Source:** A window showing the assembly code for `prog1.asm`. The current line is `4 mov r0, #0x04`.
- Command:** A window showing the code size limit: `*** Restricted Version with 32768 Byte Code Size Limit` and `*** Currently used: 16 Bytes (0%)`.
- Memory:** A window showing the memory address and content.

The Stop icon (a red square with a white 'X') in the toolbar is circled in blue, and a green arrow points from the text "Click on Stop icon" to it.

Execute the program: 3. Stop the Run to check the result

The screenshot shows the uVision IDE interface. The **Registers** window on the left lists registers R0 through R15, with R15 (PC) highlighted. A blue circle is drawn around the R0-R15 section. The **Disassembly** window on the right shows the assembly code for `prog1.asm`, with the instruction `my_ext b my_ext` highlighted. A green arrow points from the R0-R15 section of the Registers window to the highlighted instruction in the Disassembly window. The **Command** window at the bottom shows the output of the program, indicating a restricted version with 32768 Byte Code Size and 16 Bytes (0%) currently used.

Registers

Register	Value
R0	0x00000004
R1	0x00000005
R2	0x00000009
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x0000000C
CPSR	0x000000D3
SPSR	0x00000000

Disassembly

```
7: my_ext b my_ext
0x0000000C EAffffff B 0x0000000C
0x00000010 00000000 ANDEQ R0,R0,R0
0x00000014 00000000 ANDFO R0,R0,R0
```

prog1.asm

```
1 AREA PRG, CODE, READONLY
2 ENTRY
3 START
4 mov r0,#0x04
5 mov r1,#0x05
6 add r2,r0,r1
7 my_ext b my_ext
8 END
```

Command

```
*** Restricted Version with 32768 Byte Code Size I
*** Currently used: 16 Bytes (0%)
```

Results available in Registers

Stop Execution of the program: Stop Debug session

C:\Users\my\OneDrive\Documents\Lab4.uvproj - µVision

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

Start/Stop Debug Session (Ctrl+F5)
Enter or leave a debug session

Registers

Register	Value
Current	
R0	0x00000004
R1	0x00000005
R2	0x00000009
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x0000000C
CPSR	0x000000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	

Disassembly

```
7: my_ext b my_ext  
0x0000000C EAffffff B 0x0000000C  
0x00000010 00000000 ANDEQ R0,R0,R0  
0x00000014 00000000 ANDEQ R0,R0,R0
```

prog1.asm

```
1 AREA PRG, CODE, READONLY  
2 ENTRY  
3 START  
4 mov r0,#0x04  
5 mov r1,#0x05  
6 add r2,r0,r1  
7 my_ext b my_ext  
8 END
```

Command

```
*** Restricted Version with 32768 Byte Code Size Limit  
*** Currently used: 16 Bytes (0%)
```

Memory 1

Address:

ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet

Call Stack + Locals Memory 1

Stop Execution of the program

