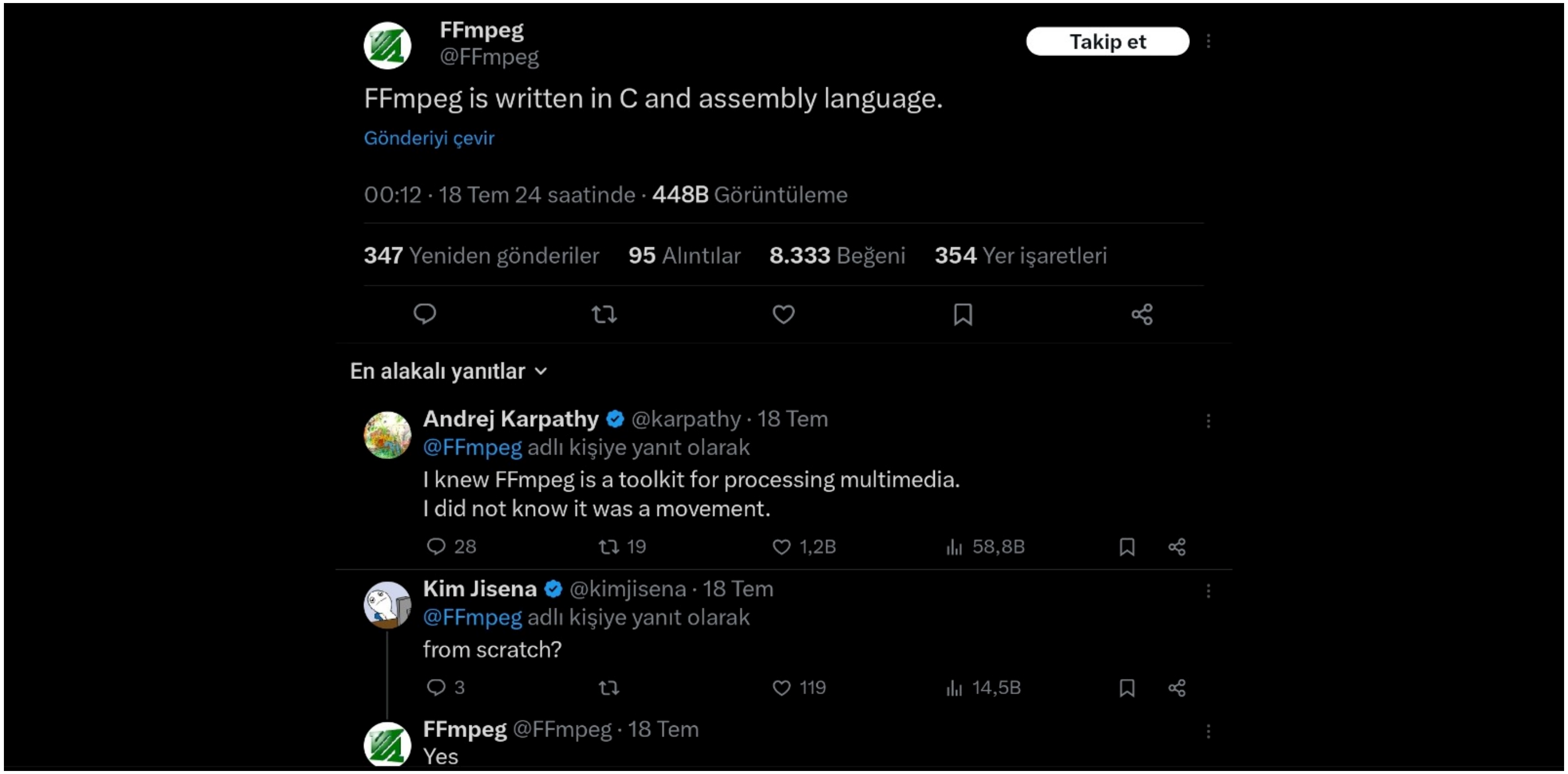
Microprocessor Systems

Res. Assist. M. Alpaslan TAVUKCU
tavukcu22@itu.edu.tr
2024



FFmpeg
@FFmpeg

FFmpeg is written in C and assembly language.

00:12 · 18 Tem 24 saatinde · **448B** Görüntüleme

347 Yeniden gönderiler **95** Alıntılar **8.333** Beğeni **354** Yer işaretleri

Andrej Karpathy @karpathy · 18 Tem
@FFmpeg adlı kişiye yanıt olarak
I knew FFmpeg is a toolkit for processing multimedia.
I did not know it was a movement.
28 19 1,2B 58,8B

Kim Jisena @kimjisena · 18 Tem
@FFmpeg adlı kişiye yanıt olarak
from scratch?
3 119 14,5B

FFmpeg @FFmpeg · 18 Tem
Yes

ARM CORTEX M0-PLUS

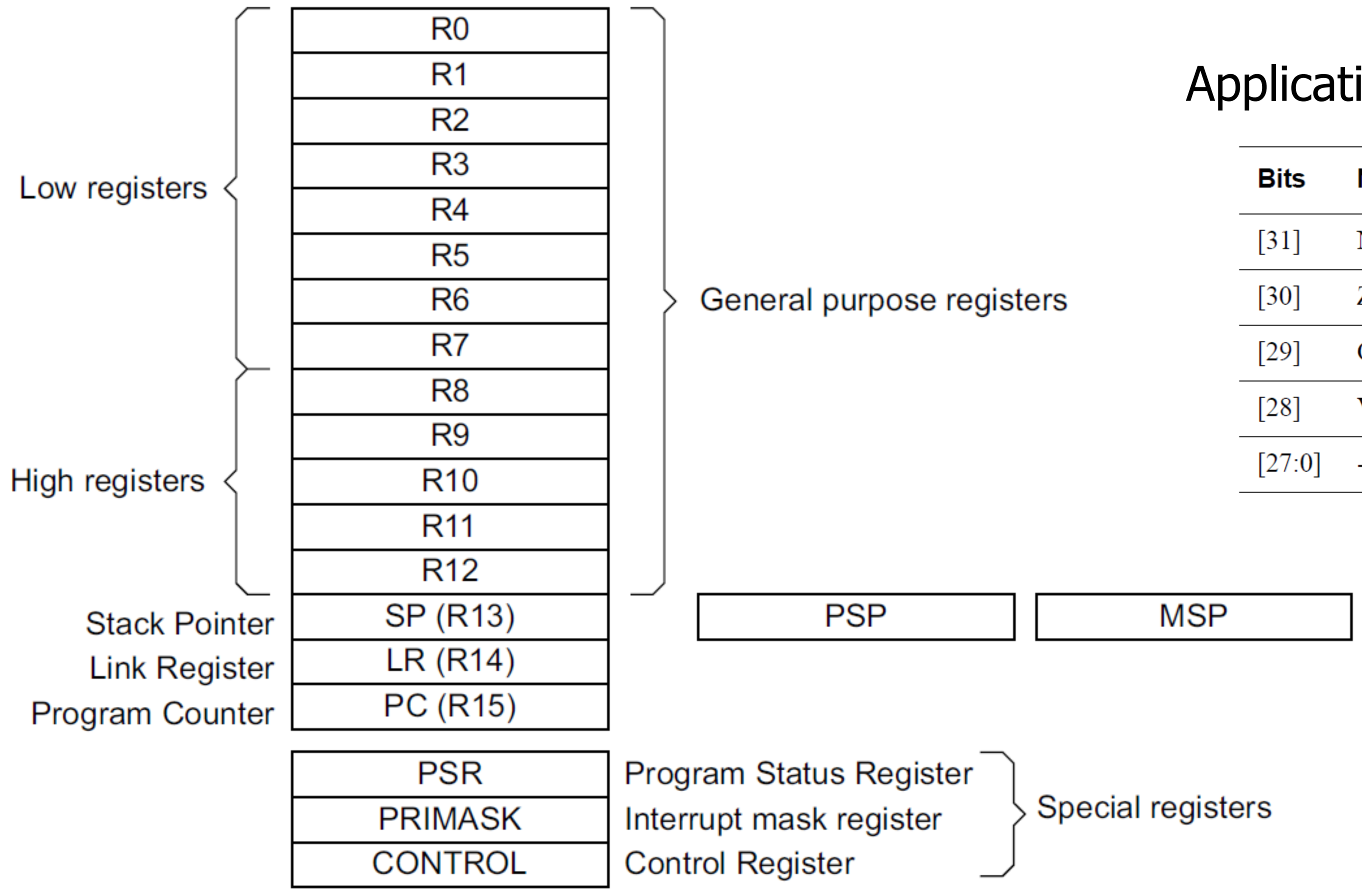
Introduction

- Arm Cortex-M0-plus
  - Armv6-M Architecture
  - 2-stages Pipeline
  - **Thumb** and Thumb-2 ISA Support
  - Non-maskable Interrupt (NMI) + 1 to 32 physical interrupts

- Keil µVision IDE
  - Programming Languages: C/C++ and Assembly
  - Onboard Arm Simulator
  - Debugging, Logic Analyzer, Source Browser

Core Registers

The processor core registers are:



Application Stasus Register

Bits	Name	Function
[31]	N	Negative flag
[30]	Z	Zero flag
[29]	C	Carry or borrow flag
[28]	V	Overflow flag
[27:0]	-	Reserved

Memory

Address range	Memory region	Memory type[a]	XN[a]	Description
0x00000000- 0x1FFFFFFF	Code	Normal	-	Executable region for program code. You can also put data here.
0x20000000- 0x3FFFFFFF	SRAM	Normal	-	Executable region for data. You can also put code here.
0x40000000- 0x5FFFFFFF	Peripheral	Device	XN	External device memory.
0x60000000- 0x9FFFFFFF	External RAM	Normal	-	Executable region for data.
0xA0000000- 0xDFFFFFFF	External device	Device	XN	External device memory.
0xE0000000- 0xE00FFFFF	Private Peripheral Bus	Strongly-ordered	XN	This region includes the NVIC, System timer, and System Control Block. Only word accesses can be used in this region.
0xE0100000- 0xFFFFFFFF	Device	Device	XN	Implementation-specific.

Memory

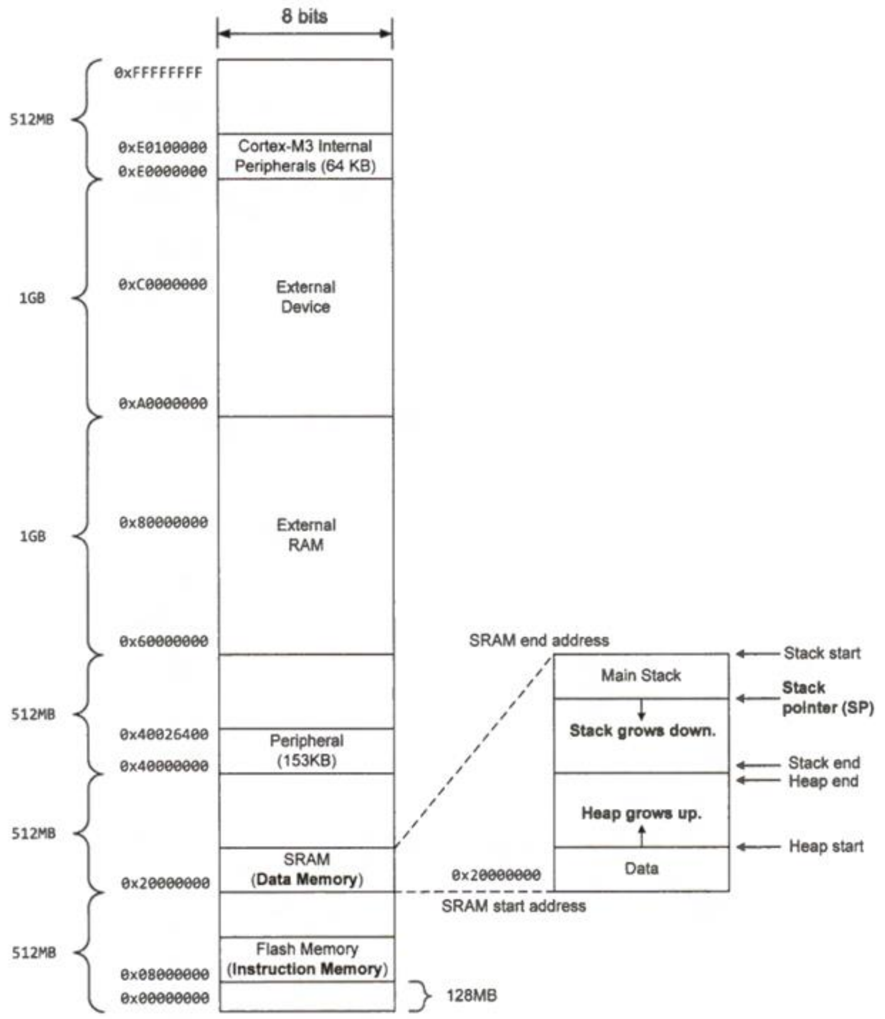


Figure 1-9. Example memory map of a 4GB memory space. The instruction memory, the data memory, and all peripherals share the same memory address space. The memory map is fixed and contains unused region.

Instruction Set

Operation	Description	Assembler	Cycles
Move	8-bit immediate	MOVS Rd, #<imm>	1
	Lo to Lo	MOVS Rd, Rm	1
	Any to Any	MOV Rd, Rm	1
	Any to PC	MOV PC, Rm	3
Add	3-bit immediate	ADDS Rd, Rn, #<imm>	1
	All registers Lo	ADDS Rd, Rn, Rm	1
	Any to Any	ADD Rd, Rd, Rm	1
	Any to PC	ADD PC, PC, Rm	3
	8-bit immediate	ADDS Rd, Rd, #<imm>	1
	With carry	ADCS Rd, Rd, Rm	1
	Immediate to SP	ADD SP, SP, #<imm>	1
	Form address from SP	ADD Rd, SP, #<imm>	1
	Form address from PC	ADR Rd, <label>	1
Subtract	Lo and Lo	SUBS Rd, Rn, Rm	1
	3-bit immediate	SUBS Rd, Rn, #<imm>	1
	8-bit immediate	SUBS Rd, Rd, #<imm>	1
	With carry	SBCS Rd, Rd, Rm	1
	Negate	RSBS Rd, Rn, #0	1
Multiply	Multiply	MULS Rd, Rm, Rd	1 or 32[a]

Operation	Description	Assembler	Cycles
Compare	Compare	CMP Rn, Rm	1
	Negative	CMN Rn, Rm	1
	Immediate	CMP Rn, #<imm>	1
Logical	AND	ANDS Rd, Rd, Rm	1
	Exclusive OR	EORS Rd, Rd, Rm	1
	OR	ORRS Rd, Rd, Rm	1
	Bit clear	BICS Rd, Rd, Rm	1
	Move NOT	MVNS Rd, Rm	1
	AND test	TST Rn, Rm	1
Shift	Logical shift left by immediate	LSLS Rd, Rm, #<shift>	1
	Logical shift left by register	LSLS Rd, Rd, Rs	1
	Logical shift right by immediate	LSRS Rd, Rm, #<shift>	1
	Logical shift right by register	LSRS Rd, Rd, Rs	1
	Arithmetic shift right	ASRS Rd, Rm, #<shift>	1
	Arithmetic shift right by register	ASRS Rd, Rd, Rs	1
Rotate	Rotate right by register	RORS Rd, Rd, Rs	1

Instruction Set

Operation	Description	Assembler	Cycles
Load	Word, immediate offset	LDR Rd, [Rn, #<imm>]	2
	Halfword, immediate offset	LDRH Rd, [Rn, #<imm>]	2
	Byte, immediate offset	LDRB Rd, [Rn, #<imm>]	2
	Word, register offset	LDR Rd, [Rn, Rm]	2
	Halfword, register offset	LDRH Rd, [Rn, Rm]	2
	Signed halfword, register offset	LDRSH Rd, [Rn, Rm]	2
	Signed byte, register offset	LDRSB Rd, [Rn, Rm]	2
	PC-relative	LDR Rd, <label>	2
	SP-relative	LDR Rd, [SP, #<imm>]	2
	Multiple, excluding base	LDM Rn!, {<loreglist>}	1+N[b]
	Multiple, including base	LDM Rn, {<loreglist>}	1+N[b]
Store	Word, immediate offset	STR Rd, [Rn, #<imm>]	2
	Halfword, immediate offset	STRH Rd, [Rn, #<imm>]	2
	Byte, immediate offset	STRB Rd, [Rn, #<imm>]	2
	Word, register offset	STR Rd, [Rn, Rm]	2
	Halfword, register offset	STRH Rd, [Rn, Rm]	2
	Byte, register offset	STRB Rd, [Rn, Rm]	2
	SP-relative	STR Rd, [SP, #<imm>]	2
	Multiple	STM Rn!, {<loreglist>}	1+N[b]

Operation	Description	Assembler	Cycles
Push	Push	PUSH {<loreglist>}	1+N[b]
	Push with link register	PUSH {<loreglist>, LR}	1+N[b]
Pop	Pop	POP {<loreglist>}	1+N[b]
	Pop and return	POP {<loreglist>, PC}	4+N[c]
Branch	Conditional	B<cc> <label>	1 or 3[d]
	Unconditional	B <label>	3
	With link	BL <label>	4
	With exchange	BX Rm	3
	With link and exchange	BLX Rm	3
Extend	Signed halfword to word	SXTH Rd, Rm	1
	Signed byte to word	SXTB Rd, Rm	1
	Unsigned byte	UXTB Rd, Rm	1
Reverse	Bytes in word	REV Rd, Rm	1
	Bytes in both halfwords	REV16 Rd, Rm	1
	Signed bottom half word	REVSH Rd, Rm	1
State change	Supervisor Call	SVC #<imm>	-[e]
	Disable interrupts	CPSID i	1
	Enable interrupts	CPSIE i	1
	Read special register	MRS Rd, <specreg>	4
	Write special register	MSR <specreg>, Rn	4
	Breakpoint	BKPT #<imm>	-[e]

Instruction Set

Operation	Description	Assembler	Cycles
Hint	Send event	SEV	1
	Wait for event	WFE	2f
	Wait for interrupt	WFI	2f
	Yield	YIELD[g]	1
	No operation	NOP	1
Barriers	Instruction synchronization	ISB	4
	Data memory	DMB	4
	Data synchronization	DSB	4

BRANCH CONDITION CODES

cond	Mnemonic extension	Meaning	Condition flags
0000	EQ	Equal	Z == 1
0001	NE	Not equal	Z == 0
0010	CS [a]	Carry set	C == 1
0011	CC [b]	Carry clear	C == 0
0100	MI	Minus, negative	N == 1
0101	PL	Plus, positive or zero	N == 0
0110	VS	Overflow	V == 1
0111	VC	No overflow	V == 0
1000	HI	Unsigned higher	C == 1 and Z == 0
1001	LS	Unsigned lower or same	C == 0 or Z == 1
1010	GE	Signed greater than or equal	N == V
1011	LT	Signed less than	N != V
1100	GT	Signed greater than	Z == 0 and N == V
1101	LE	Signed less than or equal	Z == 1 or N != V
1110[c]	None (AL) [d]	Always (unconditional)	Any

a. HS (unsigned higher or same) is a synonym for CS.
b. LO (unsigned lower) is a synonym for CC.
c. This value is never encoded in any ARMv6-M Thumb instruction.
d. AL is an optional mnemonic extension for always.

IMPORTANT!

Literals can be expressed as:

- Decimal numbers, for example `123`.
- Hexadecimal numbers, for example `0x7B`.
- Numbers in any base from `2` to `9`, for example `5_204` is a number in base `5`.
- Floating point numbers, for example `123.4`.
- Boolean values `{TRUE}` or `{FALSE}`.
- Single character values enclosed by single quotes, for example `'w'`.
- Strings enclosed in double quotes, for example `"This is a string"`.

EX. 01: MOVE & ADD

```
 AREA example, CODE, READONLY
        ENTRY
        ALIGN
__main  FUNCTION
        EXPORT __main
        MOVS     R0, #10                    ; Set up parameters
        MOVS     R1, #3
        ADD      R0, R0, R1                 ; R0 = R0 + R1

stop    B    stop                           ; Branch stop

        ENDFUNC
        END
```

EX. 02: SUBROUTINES

```
 AREA example, CODE, READONLY
        ENTRY
        ALIGN
__main  FUNCTION
        EXPORT __main
        MOVS     R0, #10                  ; Set up parameters
        MOVS     R1, #3
        BL      doadd                    ; Call subroutine

stop    B    stop                        ; Branch stop

doadd   ADD     R0, R0, R1               ; Subroutine code
        BX      LR                       ; Return from subroutine

        ENDFUNC                          ; Finish function
        END                              ; Finish assembly file
```

EX. 03: GCD

```
int gcd(int a, int b)
{
    while (a != b)
      {
        if (a > b)
            a = a - b;
        else
            b = b - a;
      }
    return a;
}
```

EX. 03: GCD

```
 AREA example, CODE, READONLY    ; Declare new area
        ENTRY                                ; Declare as entry point
        ALIGN                                ; Ensures that __main addresses the following instruction

__main  FUNCTION                             ; Enable Debug
        EXPORT __main                        ; Make __main as global to access from startup file
        MOVS     R0, #48                     ; Set up parameters (example values)
        MOVS     R1, #18
        BL       doGCD                        ; Call GCD subroutine

stop    B    stop                            ; Branch stop

doGCD   CMP     R0, R1                       ; Compare a and b (R0 and R1)
        BEQ     endGCD                       ; If they are equal, branch to endGCD
        BGT     subtractA                    ; If a > b, branch to subtractA
        B       subtractB                    ; Else, branch to subtractB

subtractA SUBS    R0, R0, R1                   ; a = a - b
          B       doGCD                        ; Repeat the loop

subtractB SUBS    R1, R1, R0                   ; b = b - a
          B       doGCD                        ; Repeat the loop

endGCD    MOV     R0, R0                        ; Move the result (GCD) into R0 for returning
          BX      LR                           ; Return from subroutine

        ENDFUNC                              ; Finish function
        END                                  ; Finish assembly file
```

Directives Reference

- Directives are instructions used by the **assembler** to help automate the assembly process and to improve program readability.
- Important directive for us:
- **AREA:** instructs the assembler to assemble a new code or data section.
  - Example Usage:
  - **AREA   Example, CODE, READONLY**
    - Example-> name of section
    - Code-> Contains machine instructions.
    - READONLY-> Indicates that this section must not be written to.

Directives Reference

- **ALIGN:** aligns the current location to a specified boundary by padding with zeros or NOP instructions.
  - Example usage:

```
        ALIGN          ; ensures that subroutine1 addresses
subroutine1             ; the following instruction.
        MOV r5,#0x5
```

- **DCB:** allocates one or more bytes of memory, and defines the initial runtime contents of the memory.
- **DCW - DCWU:** works as DCB for 2 bytes data.
- **DCD and DCDU:** works as DCB for 4 bytes data.
- **DCQ - DCQU:** works as DCB for 8 bytes data.

```
C_string    DCB  "C_string",0
data        DCW  -225,2*number   ; number must already be defined
data1       DCD  1,5,20
data        DCQ  -225,2_101
```

Directives Reference

- **EQU:** gives a symbolic name to a numeric constant, a register-relative value or a PC-relative value.

```
value1      EQU 2
value2      EQU 0x2215, CODE32
value3      EQU 0x214456, DATA
```

- **FUNCTION or PROC:** marks the start of a function. PROC is a synonym for FUNCTION. They enable the debug.
- **ENDFUNC or ENDP:** marks the end of a function. ENDP is a synonym for ENDFUNC.
- **END:** informs the assembler that it has reached the end of a source file.
- **EXPORT or GLOBAL:** declares a symbol that can be used by the linker to resolve symbol references in separate object and library files.

Directives Reference

- **IMPORT and EXTERN:** provide the assembler with a name that is not defined in the current assembly.
- **ENTRY:** declares an entry point to a program.
- **THUMB:** instructs the assembler to interpret subsequent instructions as Thumb instructions, using the UAL syntax.

LDR

- LDR Rd, =const
- LDR Rd, =label

The LDR pseudo-instruction generates the most efficient single instruction for a specific constant:

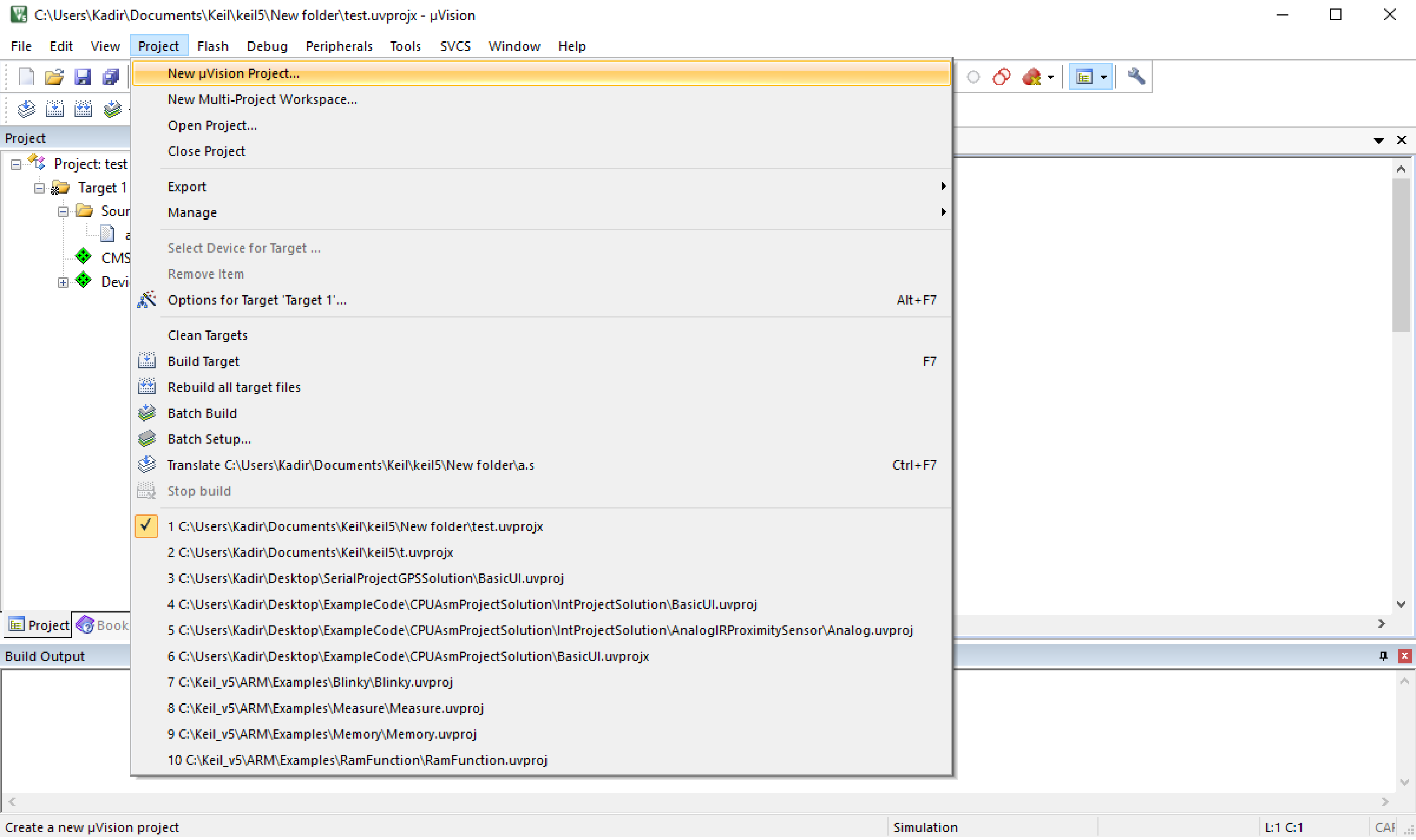
- If the constant can be constructed with a single MOV instruction, the assembler generates the appropriate instruction.
- If the constant cannot be constructed with a single MOV instruction, the assembler: places the value in a literal pool (a portion of memory embedded in the code to hold constant values) generates an LDR instruction with a program-relative address that reads the constant from the literal pool.

EX: LDR

```
 AREA example, CODE, READONLY     ; Declare new area
        ENTRY                     ; Declare as entry point
        ALIGN                     ; Ensures __main addresses the following instruction

__main  FUNCTION                  ; Enable Debug
        EXPORT __main             ; Make __main as global to access from startup file
        LDR R0, =var1             ; Load the address of 'var1' into r0
        LDR R1, [R0]              ; Load the value at the
                                  ; address in r0 into r1 (0x12345678)
        LDR R2, var1

stop    B    stop                        ; Branch stop

var1    DCD 0x12345678                   ; Define a 32-bit constant in memory

        ENDFUNC                          ; Finish function
        END                              ; Finish assembly file
```

DEMO

Start New µVision Project – Step 1



C:\Users\Kadir\Documents\Keil\keil5\New folder\test.uvprojx - µVision

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

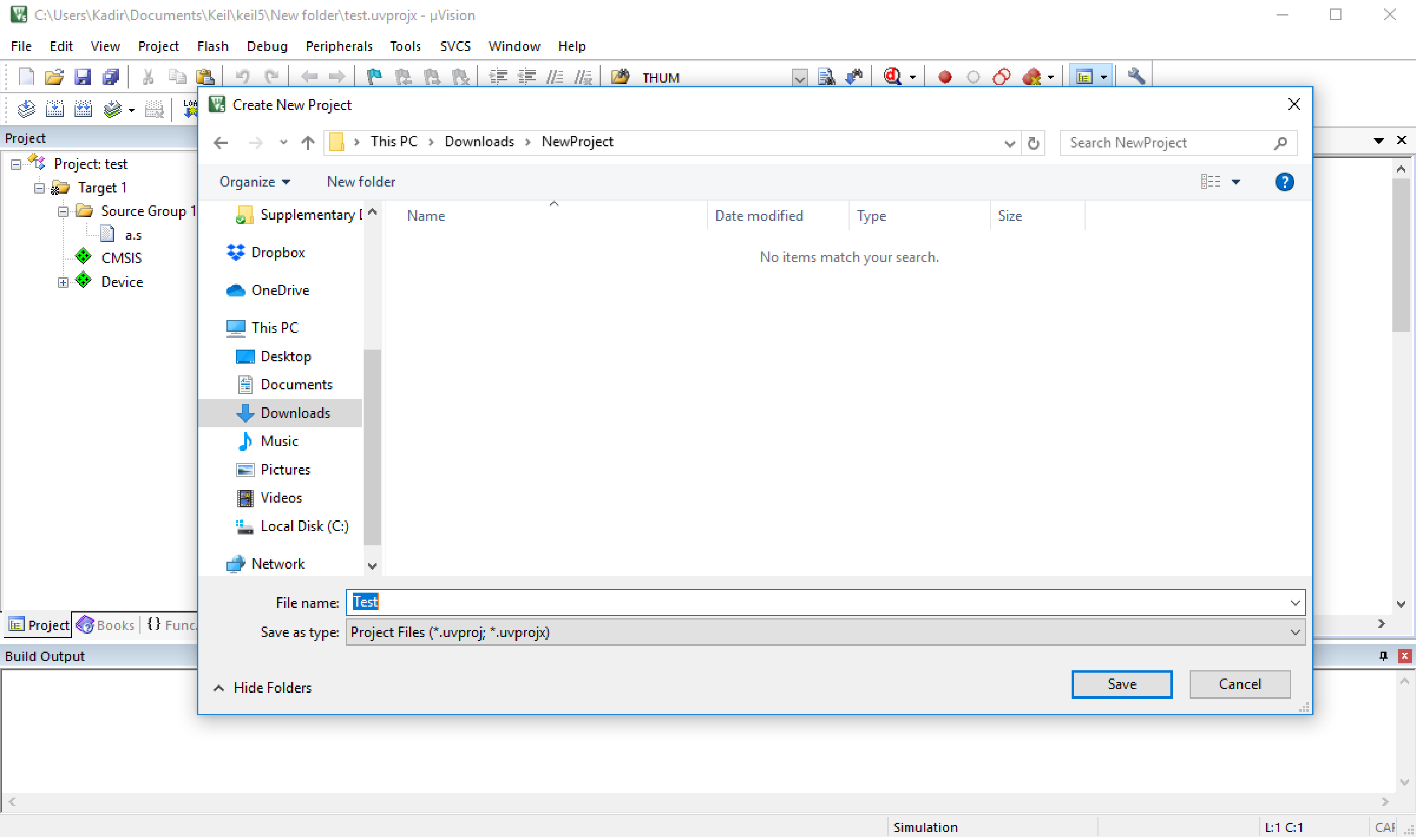
Project
Project: test
Target 1
Sour
CMS
Devi

New µVision Project...
New Multi-Project Workspace...
Open Project...
Close Project
Export
Manage
Select Device for Target ...
Remove Item
Options for Target 'Target 1'... Alt+F7
Clean Targets
Build Target F7
Rebuild all target files
Batch Build
Batch Setup...
Translate C:\Users\Kadir\Documents\Keil\keil5\New folder\a.s Ctrl+F7
Stop build
✓ 1 C:\Users\Kadir\Documents\Keil\keil5\New folder\test.uvprojx
2 C:\Users\Kadir\Documents\Keil\keil5\t.uvprojx
3 C:\Users\Kadir\Desktop\SerialProjectGPSSolution\BasicUI.uvproj
4 C:\Users\Kadir\Desktop\ExampleCode\CPUAsmProjectSolution\IntProjectSolution\BasicUI.uvproj
5 C:\Users\Kadir\Desktop\ExampleCode\CPUAsmProjectSolution\IntProjectSolution\AnalogIRProximitySensor\Analog.uvproj
6 C:\Users\Kadir\Desktop\ExampleCode\CPUAsmProjectSolution\BasicUI.uvprojx
7 C:\Keil_v5\ARM\Examples\Blinky\Blinky.uvproj
8 C:\Keil_v5\ARM\Examples\Measure\Measure.uvproj
9 C:\Keil_v5\ARM\Examples\Memory\Memory.uvproj
10 C:\Keil_v5\ARM\Examples\RamFunction\RamFunction.uvproj

Project Book
Build Output

Create a new µVision project Simulation L:1 C:1 CAP

Start New µVision Project – Step 2

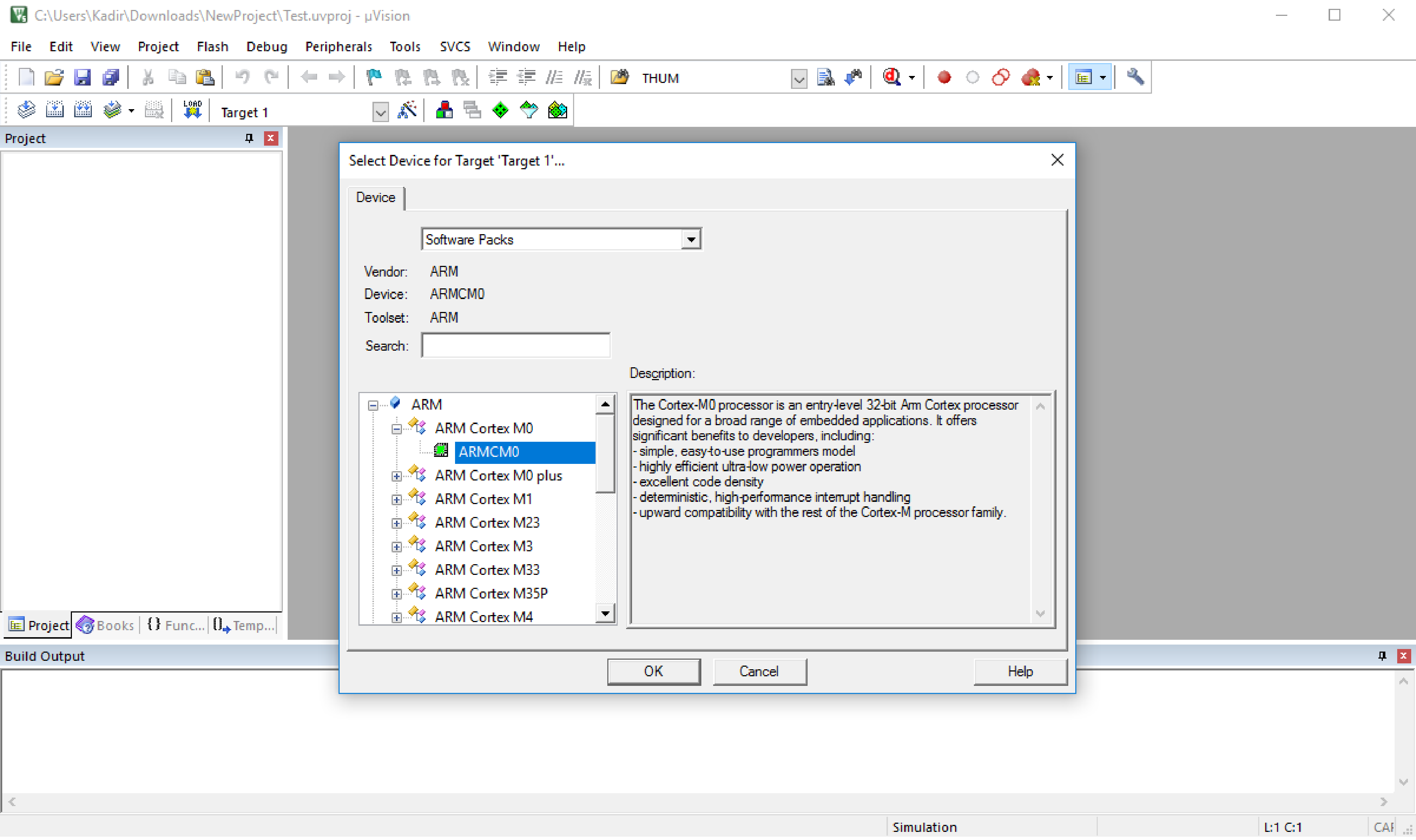


No items match your search.

File name: Test

Save as type: Project Files (*.uvproj; *.uvprojx)

Start New µVision Project – Step 3


C:\Users\Kadir\Downloads\NewProject\Test.uvproj - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
THUM
Target 1
Project
Project Books Func... Temp...
Build Output
Simulation
L:1 C:1
CAP

Select Device for Target 'Target 1'...

Device

Software Packs

Vendor: ARM
Device: ARMCM0
Toolset: ARM
Search:

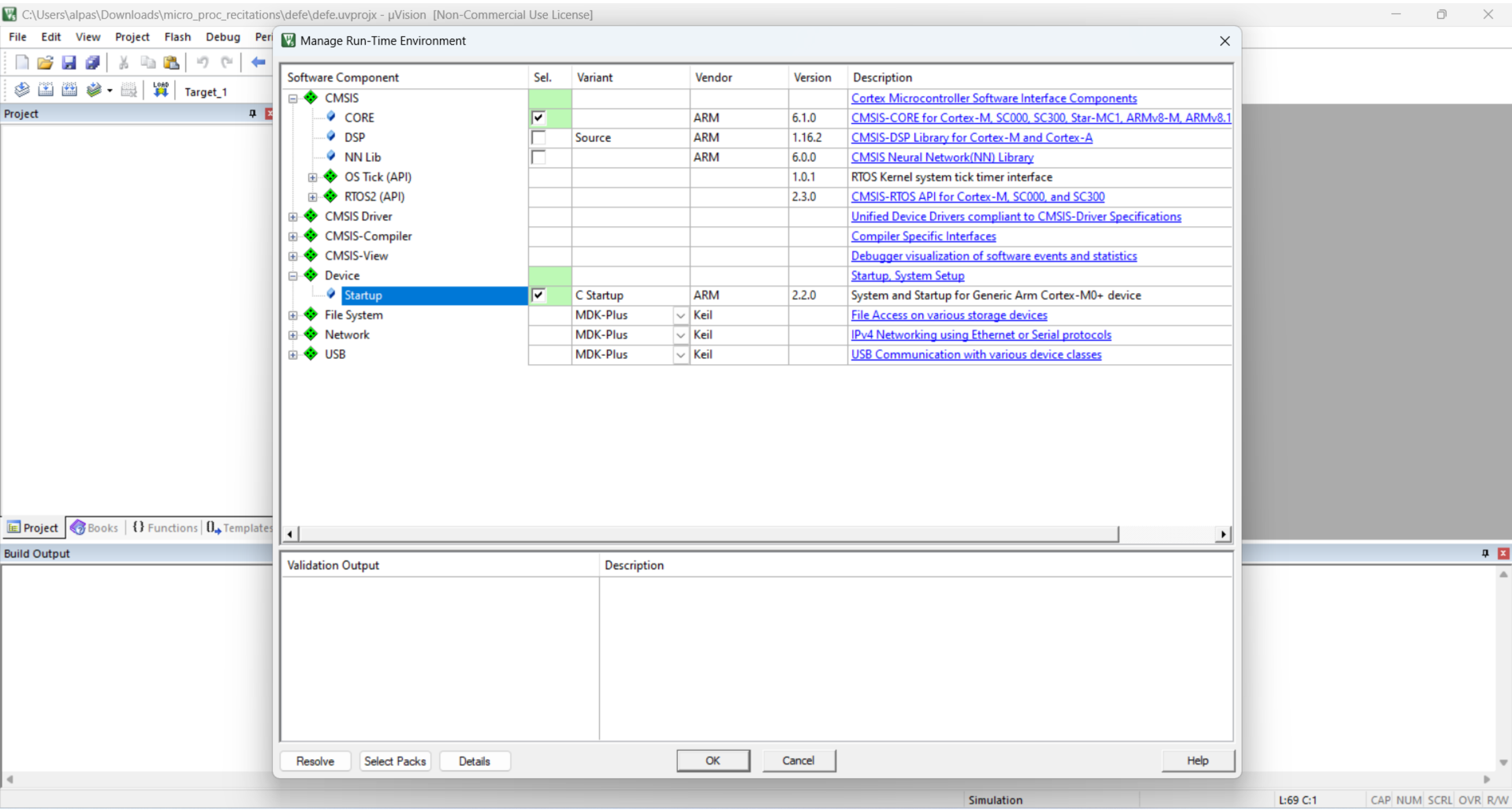
- ARM
  - ARM Cortex M0
    - ARMCM0
  - ARM Cortex M0 plus
  - ARM Cortex M1
  - ARM Cortex M23
  - ARM Cortex M3
  - ARM Cortex M33
  - ARM Cortex M35P
  - ARM Cortex M4

Description:

The Cortex-M0 processor is an entry-level 32-bit Arm Cortex processor designed for a broad range of embedded applications. It offers significant benefits to developers, including:
- simple, easy-to-use programmers model
- highly efficient ultra-low power operation
- excellent code density
- deterministic, high-performance interrupt handling
- upward compatibility with the rest of the Cortex-M processor family.

OK Cancel Help

Start New µVision Project – Step 4



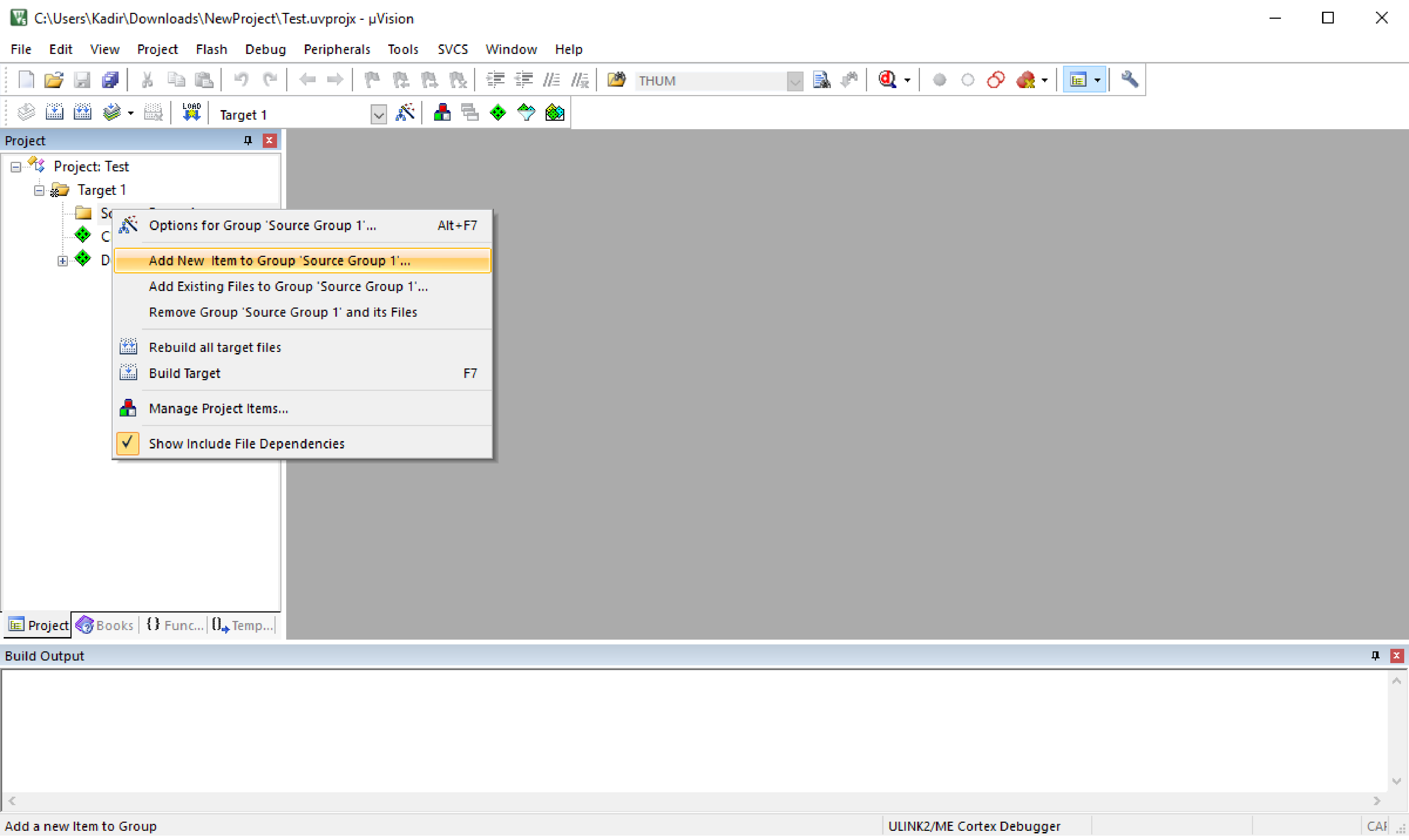
C:\Users\alpas\Downloads\micro_proc_recitations\defe\defe.uvprojx - µVision [Non-Commercial Use License]

Manage Run-Time Environment

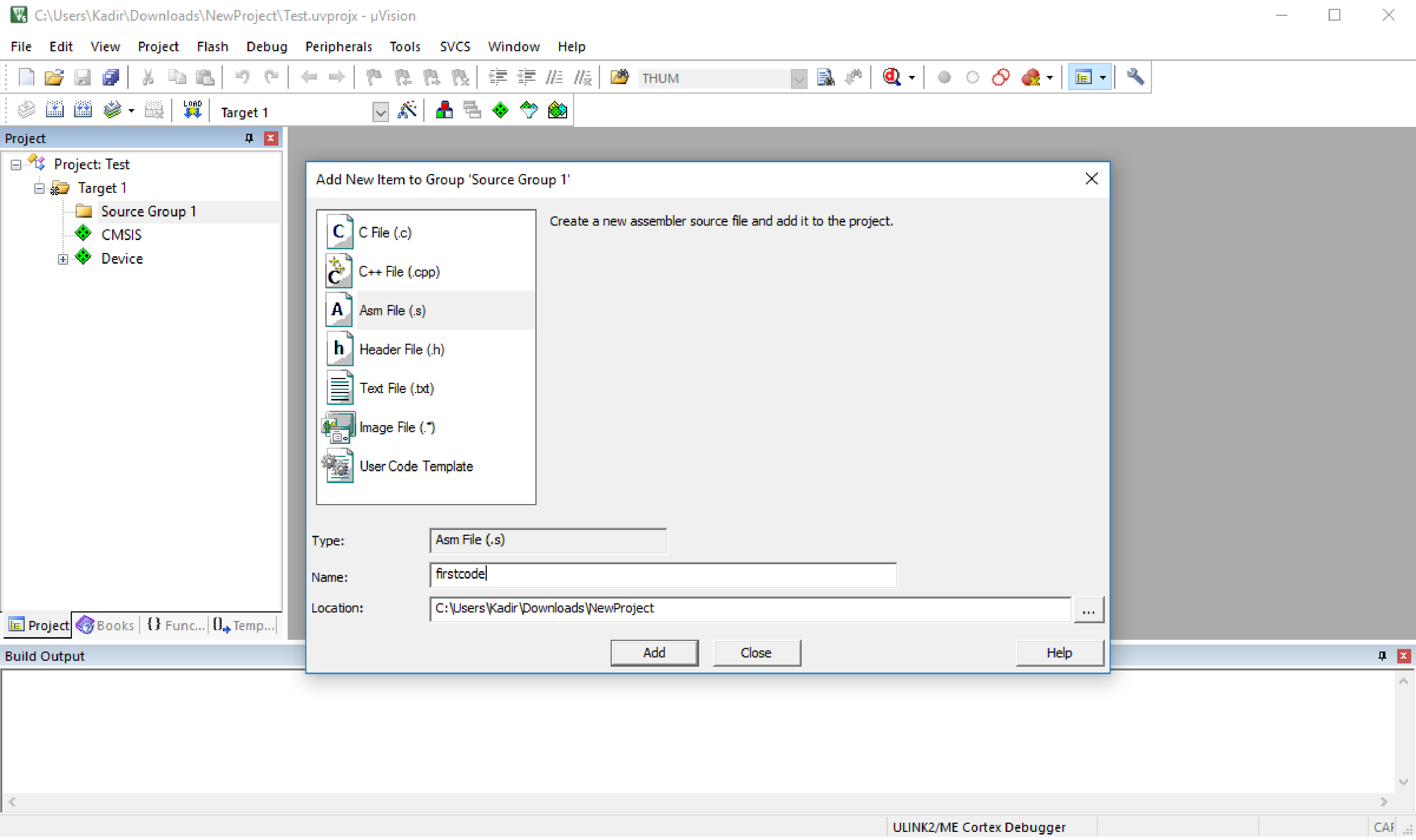
Software Component	Sel.	Variant	Vendor	Version	Description
CMSIS					Cortex Microcontroller Software Interface Components
CORE	✓		ARM	6.1.0	CMSIS-CORE for Cortex-M, SC000, SC300, Star-MC1, ARMv8-M, ARMv8.1
DSP	☐	Source	ARM	1.16.2	CMSIS-DSP Library for Cortex-M and Cortex-A
NN Lib	☐		ARM	6.0.0	CMSIS Neural Network(NN) Library
OS Tick (API)				1.0.1	RTOS Kernel system tick timer interface
RTOS2 (API)				2.3.0	CMSIS-RTOS API for Cortex-M, SC000, and SC300
CMSIS Driver					Unified Device Drivers compliant to CMSIS-Driver Specifications
CMSIS-Compiler					Compiler Specific Interfaces
CMSIS-View					Debugger visualization of software events and statistics
Device					Startup, System Setup
Startup	✓	C Startup	ARM	2.2.0	System and Startup for Generic Arm Cortex-M0+ device
File System		MDK-Plus	Keil		File Access on various storage devices
Network		MDK-Plus	Keil		IPv4 Networking using Ethernet or Serial protocols
USB		MDK-Plus	Keil		USB Communication with various device classes

Validation Output	Description

Adding new assembly file – Step 1


C:\Users\Kadir\Downloads\NewProject\Test.uvprojx - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
THUM
Target 1
Project
Project: Test
Target 1
Options for Group 'Source Group 1'... Alt+F7
Add New Item to Group 'Source Group 1'...
Add Existing Files to Group 'Source Group 1'...
Remove Group 'Source Group 1' and its Files
Rebuild all target files
Build Target F7
Manage Project Items...
Show Include File Dependencies
Project Books Func... Temp...
Build Output
Add a new Item to Group
ULINK2/ME Cortex Debugger

Adding new assembly file – Step 2


C:\Users\Kadir\Downloads\NewProject\Test.uvprojx - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
THUM
Target 1
Project
Project: Test
Target 1
Source Group 1
CMSIS
Device
Project Books Func... Temp...
Build Output
ULINK2/ME Cortex Debugger
CAP

Add New Item to Group 'Source Group 1'

- C File (.c)
- C++ File (.cpp)
- Asm File (.s)
- Header File (.h)
- Text File (.txt)
- Image File (.*)
- User Code Template

Create a new assembler source file and add it to the project.

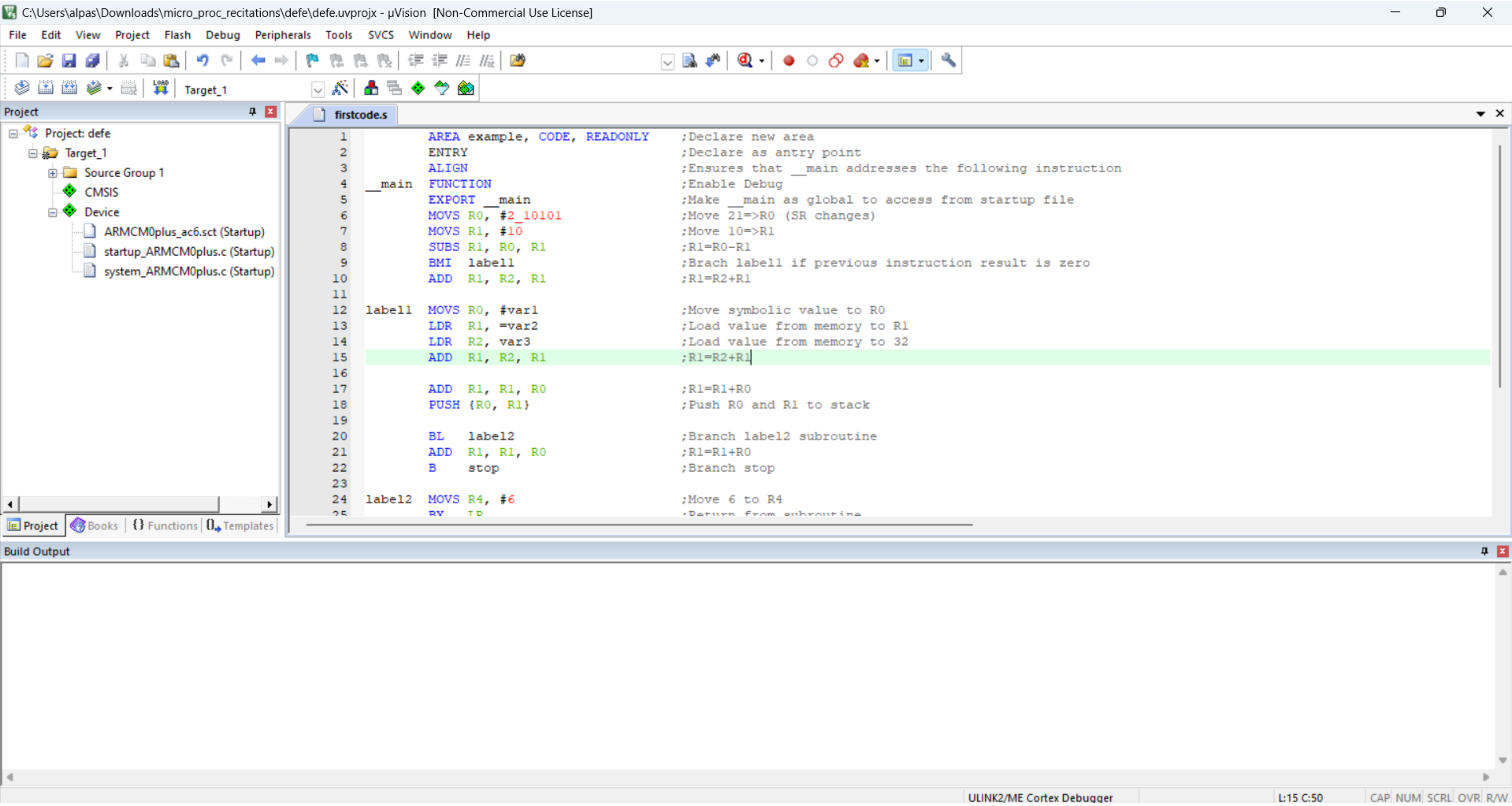
Type: Asm File (.s)

Name: firstcode

Location: C:\Users\Kadir\Downloads\NewProject

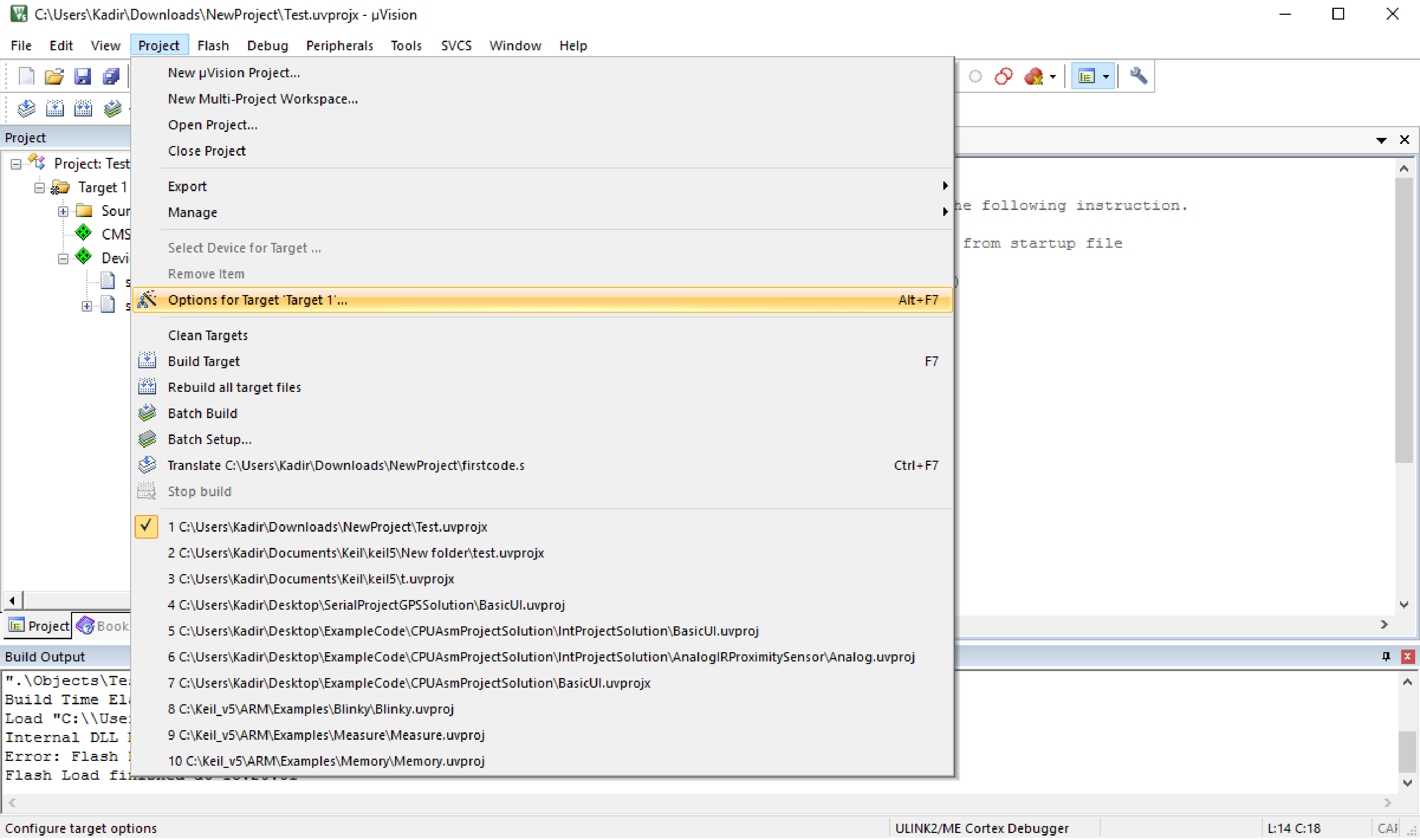
Add Close Help

Write your code in your assembly file


C:\Users\alpas\Downloads\micro_proc_recitations\defe\defe.uvprojx - µVision [Non-Commercial Use License]
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Target_1
Project
Project: defe
Target_1
Source Group 1
CMSIS
Device
ARMCM0plus_ac6.sct (Startup)
startup_ARMCM0plus.c (Startup)
system_ARMCM0plus.c (Startup)
Project
Books
Functions
Templates
firstcode.s
Build Output
ULINK2/ME Cortex Debugger
L:15 C:50
CAP NUM SCRL OVR R/W

```
        AREA example, CODE, READONLY    ;Declare new area
        ENTRY                           ;Declare as antry point
        ALIGN                           ;Ensures that __main addresses the following instruction
__main  FUNCTION                        ;Enable Debug
        EXPORT __main                   ;Make __main as global to access from startup file
        MOVS R0, #2_10101               ;Move 21=>R0 (SR changes)
        MOVS R1, #10                    ;Move 10=>R1
        SUBS R1, R0, R1                 ;R1=R0-R1
        BMI  label1                     ;Brach label1 if previous instruction result is zero
        ADD  R1, R2, R1                 ;R1=R2+R1

label1  MOVS R0, #var1                  ;Move symbolic value to R0
        LDR  R1, =var2                  ;Load value from memory to R1
        LDR  R2, var3                   ;Load value from memory to 32
        ADD  R1, R2, R1                 ;R1=R2+R1

        ADD  R1, R1, R0                 ;R1=R1+R0
        PUSH {R0, R1}                   ;Push R0 and R1 to stack

        BL   label2                     ;Branch label2 subroutine
        ADD  R1, R1, R0                 ;R1=R1+R0
        B    stop                       ;Branch stop

label2  MOVS R4, #6                     ;Move 6 to R4
        BX   LR                         ;Return from subroutine
```

Configure simulator – Step 1



C:\Users\Kadir\Downloads\NewProject\Test.uvprojx - µVision

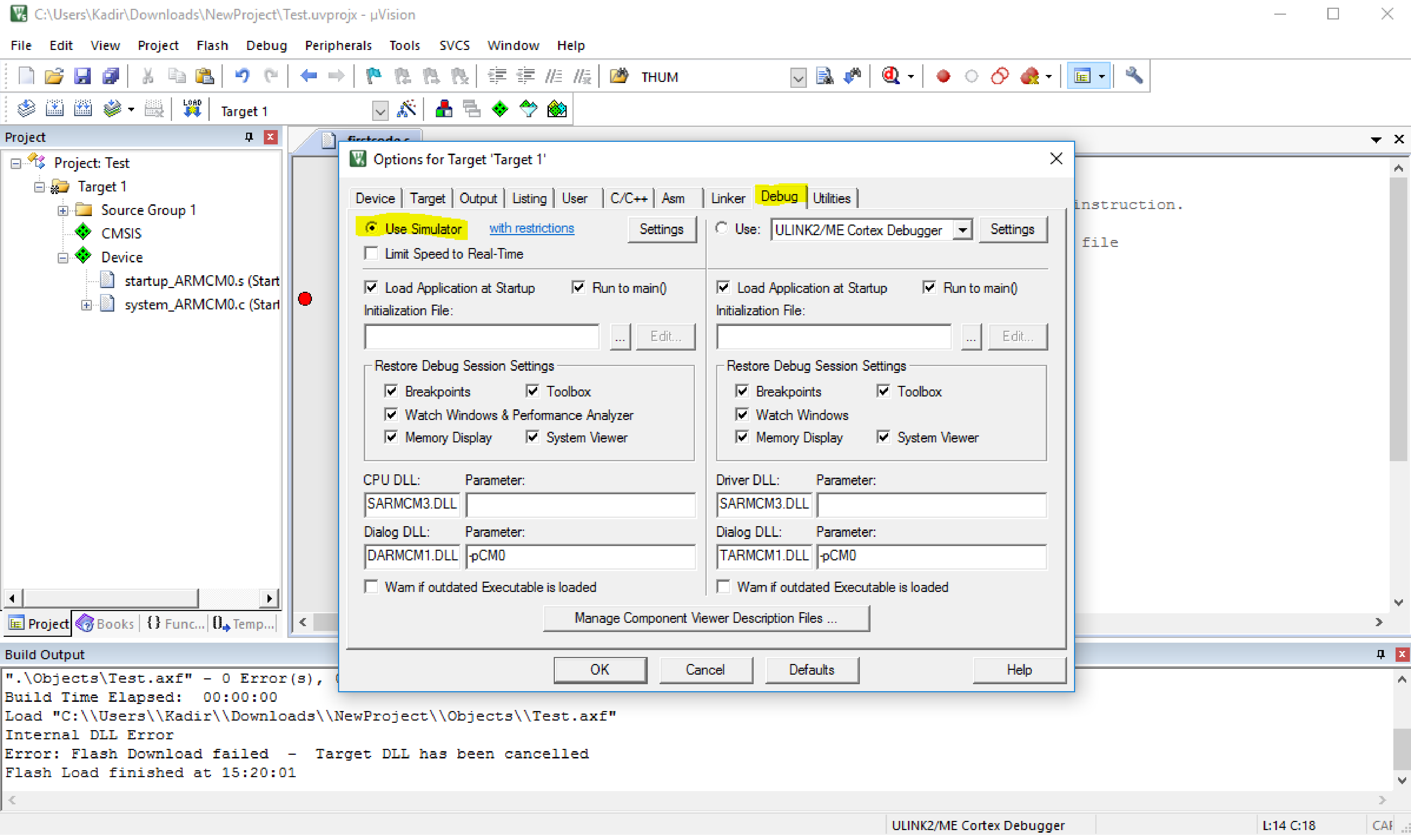
New µVision Project...
New Multi-Project Workspace...
Open Project...
Close Project
Export
Manage
Select Device for Target ...
Remove Item
Options for Target 'Target 1'... Alt+F7
Clean Targets
Build Target F7
Rebuild all target files
Batch Build
Batch Setup...
Translate C:\Users\Kadir\Downloads\NewProject\firstcode.s Ctrl+F7
Stop build
✓ 1 C:\Users\Kadir\Downloads\NewProject\Test.uvprojx
2 C:\Users\Kadir\Documents\Keil\keil5\New folder\test.uvprojx
3 C:\Users\Kadir\Documents\Keil\keil5\t.uvprojx
4 C:\Users\Kadir\Desktop\SerialProjectGPSSolution\BasicUI.uvproj
5 C:\Users\Kadir\Desktop\ExampleCode\CPUAsmProjectSolution\IntProjectSolution\BasicUI.uvproj
6 C:\Users\Kadir\Desktop\ExampleCode\CPUAsmProjectSolution\IntProjectSolution\AnalogIRProximitySensor\Analog.uvproj
7 C:\Users\Kadir\Desktop\ExampleCode\CPUAsmProjectSolution\BasicUI.uvprojx
8 C:\Keil_v5\ARM\Examples\Blinky\Blinky.uvproj
9 C:\Keil_v5\ARM\Examples\Measure\Measure.uvproj
10 C:\Keil_v5\ARM\Examples\Memory\Memory.uvproj

```
he following instruction.
 from startup file
```

```
".\Objects\Te
Build Time El
Load "C:\\Use
Internal DLL 
Error: Flash 
Flash Load fi
```

Configure target options   ULINK2/ME Cortex Debugger   L:14 C:18   CAP

Configure simulator – Step 2

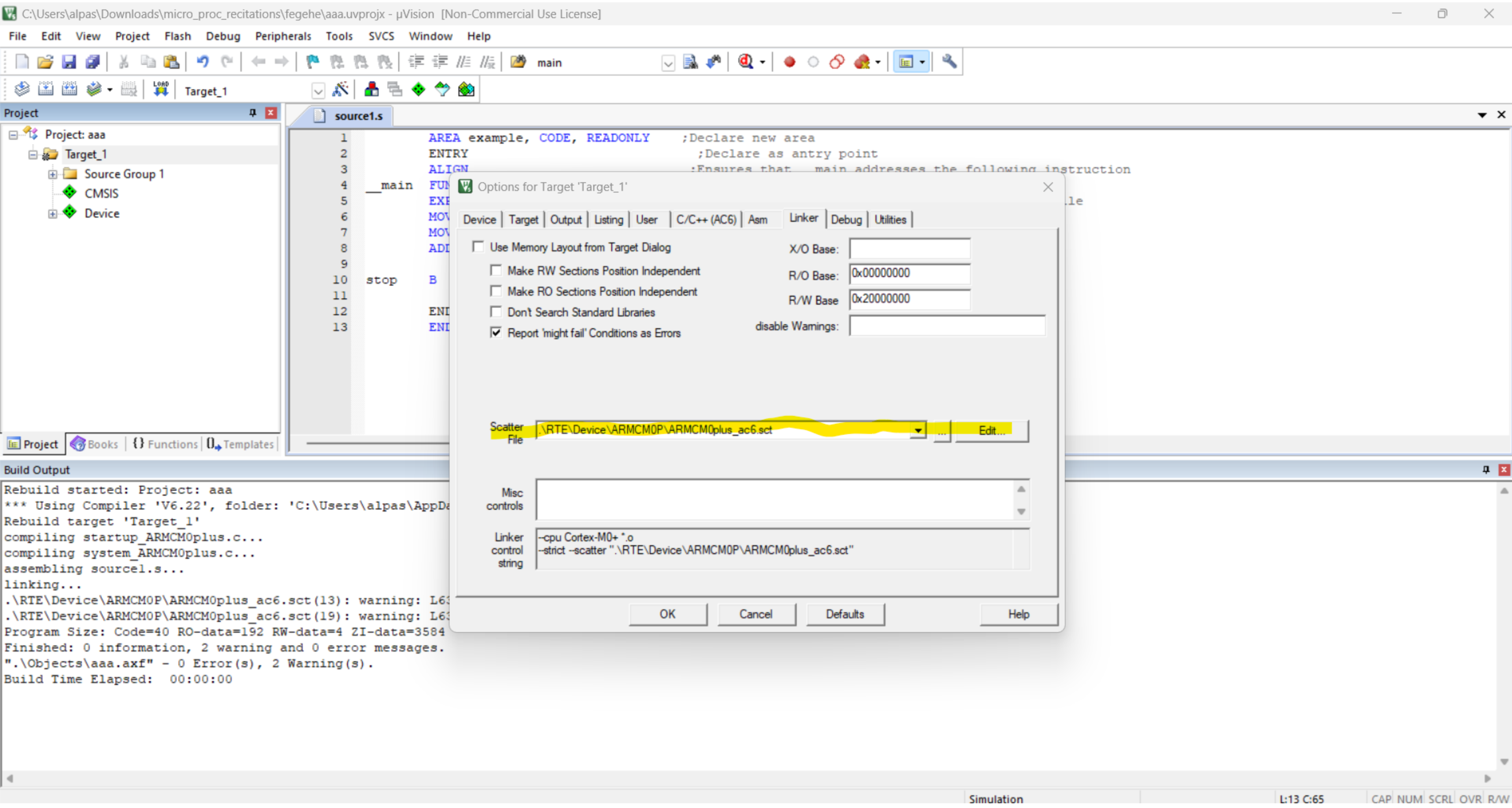

C:\Users\Kadir\Downloads\NewProject\Test.uvprojx - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Target 1
Project
Project: Test
Target 1
Source Group 1
CMSIS
Device
startup_ARMCM0.s (Start
system_ARMCM0.c (Start
Options for Target 'Target 1'
Device Target Output Listing User C/C++ Asm Linker Debug Utilities
with restrictions
Settings
ULINK2/ME Cortex Debugger
Manage Component Viewer Description Files ...
OK
Cancel
Defaults
Help
Build Output
ULINK2/ME Cortex Debugger
L:14 C:18

[✓ Use Simulator] with restrictions | Use: ULINK2/ME Cortex Debugger

Simulator	ULINK2/ME Cortex Debugger
☐ Limit Speed to Real-Time	
☑ Load Application at Startup ☑ Run to main()	☑ Load Application at Startup ☑ Run to main()
Initialization File:	Initialization File:
Restore Debug Session Settings: ☑ Breakpoints ☑ Toolbox ☑ Watch Windows & Performance Analyzer ☑ Memory Display ☑ System Viewer	Restore Debug Session Settings: ☑ Breakpoints ☑ Toolbox ☑ Watch Windows ☑ Memory Display ☑ System Viewer
CPU DLL: SARMCM3.DLL Parameter:	Driver DLL: SARMCM3.DLL Parameter:
Dialog DLL: DARMCM1.DLL Parameter: -pCM0	Dialog DLL: TARMCM1.DLL Parameter: -pCM0
☐ Warn if outdated Executable is loaded	☐ Warn if outdated Executable is loaded

```
".\Objects\Test.axf" - 0 Error(s), 
Build Time Elapsed:  00:00:00
Load "C:\\Users\\Kadir\\Downloads\\NewProject\\Objects\\Test.axf"
Internal DLL Error
Error: Flash Download failed  -  Target DLL has been cancelled
Flash Load finished at 15:20:01
```

Configure simulator – Step 3



C:\Users\alpas\Downloads\micro_proc_recitations\fegehe\aaa.uvprojx - µVision [Non-Commercial Use License]

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
main
Target_1
Project
Project: aaa
Target_1
Source Group 1
CMSIS
Device
Project Books Functions Templates
source1.s

```
        AREA example, CODE, READONLY     ;Declare new area
        ENTRY                             ;Declare as antry point
        ALIGN                             ;Ensures that __main addresses the following instruction
__main  FUN                                                                                  le
        EXE
        MOV
        MOV
        ADD

stop    B

        END
        END
```

Options for Target 'Target_1'

Device | Target | Output | Listing | User | C/C++ (AC6) | Asm | Linker | Debug | Utilities

- [ ] Use Memory Layout from Target Dialog
- [ ] Make RW Sections Position Independent
- [ ] Make RO Sections Position Independent
- [ ] Dont Search Standard Libraries
- [✓] Report 'might fail' Conditions as Errors

X/O Base:
R/O Base: 0x00000000
R/W Base 0x20000000
disable Warnings:

Scatter File: .\RTE\Device\ARMCM0P\ARMCM0plus_ac6.sct

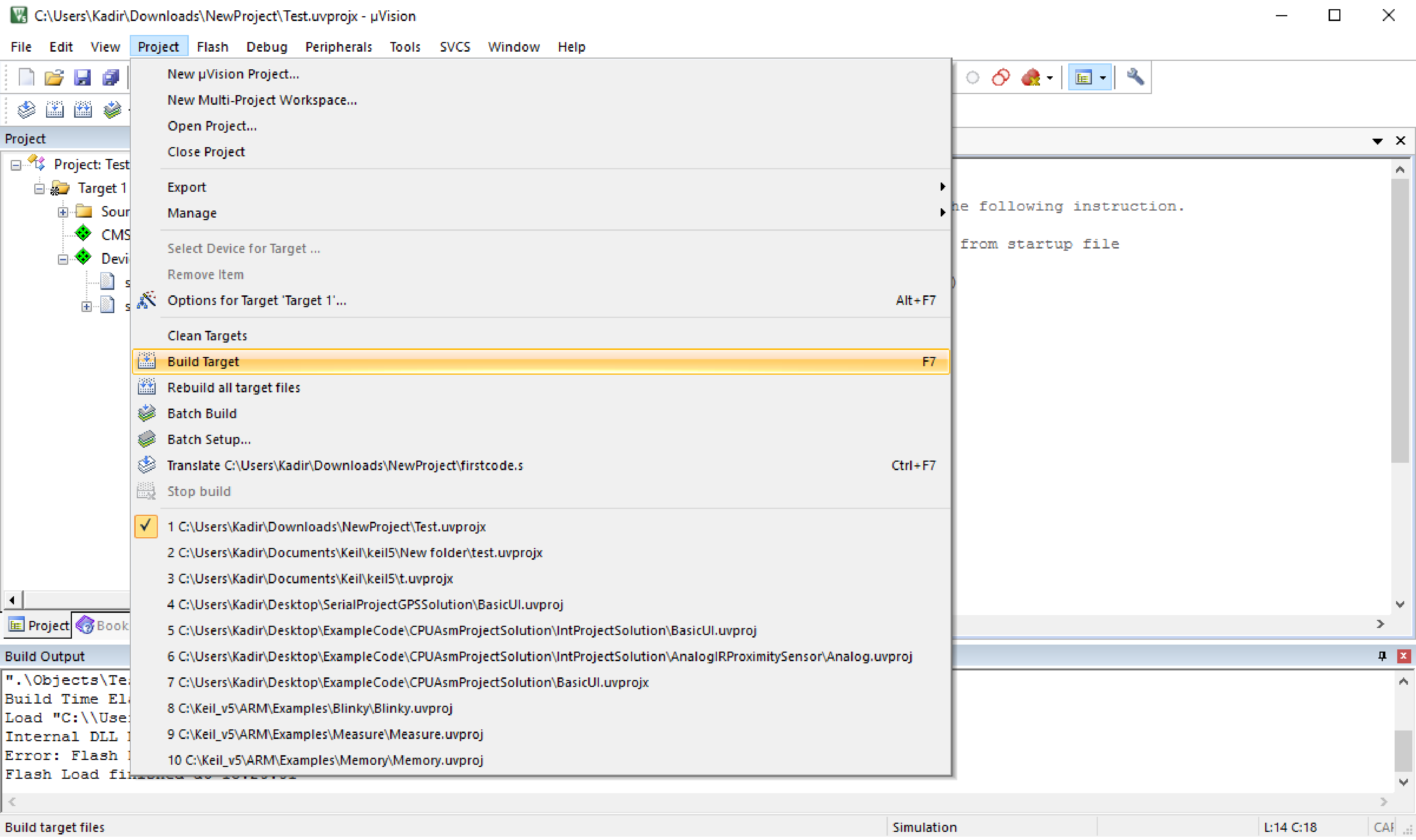
Misc controls:

Linker control string:
--cpu Cortex-M0+ *.o
--strict --scatter ".\RTE\Device\ARMCM0P\ARMCM0plus_ac6.sct"

Build Output

```
Rebuild started: Project: aaa
*** Using Compiler 'V6.22', folder: 'C:\Users\alpas\AppDa
Rebuild target 'Target_1'
compiling startup_ARMCM0plus.c...
compiling system_ARMCM0plus.c...
assembling source1.s...
linking...
.\RTE\Device\ARMCM0P\ARMCM0plus_ac6.sct(13): warning: L6
.\RTE\Device\ARMCM0P\ARMCM0plus_ac6.sct(19): warning: L6
Program Size: Code=40 RO-data=192 RW-data=4 ZI-data=3584
Finished: 0 information, 2 warning and 0 error messages.
".\Objects\aaa.axf" - 0 Error(s), 2 Warning(s).
Build Time Elapsed:  00:00:00
```

Build your project to run

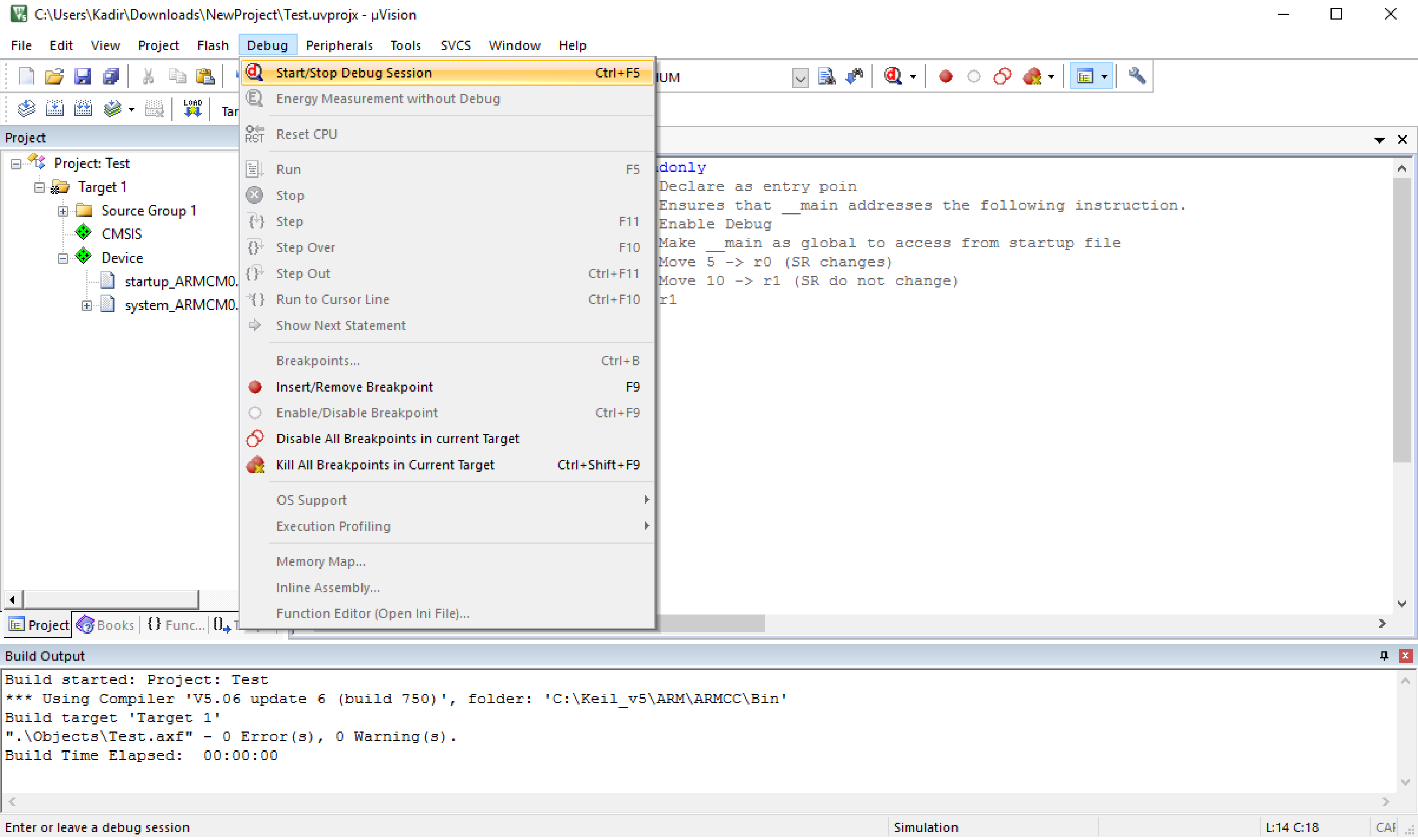


C:\Users\Kadir\Downloads\NewProject\Test.uvprojx - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
New µVision Project...
New Multi-Project Workspace...
Open Project...
Close Project
Export
Manage
Select Device for Target ...
Remove Item
Options for Target 'Target 1'... Alt+F7
Clean Targets
Build Target F7
Rebuild all target files
Batch Build
Batch Setup...
Translate C:\Users\Kadir\Downloads\NewProject\firstcode.s Ctrl+F7
Stop build
1 C:\Users\Kadir\Downloads\NewProject\Test.uvprojx
2 C:\Users\Kadir\Documents\Keil\keil5\New folder\test.uvprojx
3 C:\Users\Kadir\Documents\Keil\keil5\t.uvprojx
4 C:\Users\Kadir\Desktop\SerialProjectGPSSolution\BasicUI.uvproj
5 C:\Users\Kadir\Desktop\ExampleCode\CPUAsmProjectSolution\IntProjectSolution\BasicUI.uvproj
6 C:\Users\Kadir\Desktop\ExampleCode\CPUAsmProjectSolution\IntProjectSolution\AnalogIRProximitySensor\Analog.uvproj
7 C:\Users\Kadir\Desktop\ExampleCode\CPUAsmProjectSolution\BasicUI.uvprojx
8 C:\Keil_v5\ARM\Examples\Blinky\Blinky.uvproj
9 C:\Keil_v5\ARM\Examples\Measure\Measure.uvproj
10 C:\Keil_v5\ARM\Examples\Memory\Memory.uvproj
Project
Project: Test
Target 1
Sour
CMS
Devi
Project
Book
Build Output
Build target files
Simulation
L:14 C:18
CAI

```
he following instruction.
 from startup file
```

```
".\Objects\Te
Build Time El
Load "C:\\Use
Internal DLL 
Error: Flash 
Flash Load fi
```

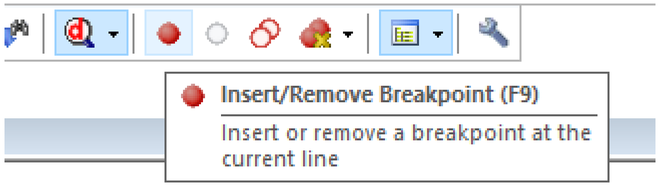
Start your program in debug mode

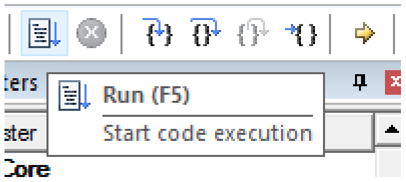

C:\Users\Kadir\Downloads\NewProject\Test.uvprojx - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Start/Stop Debug Session Ctrl+F5
Energy Measurement without Debug
Reset CPU
Run F5
Stop
Step F11
Step Over F10
Step Out Ctrl+F11
Run to Cursor Line Ctrl+F10
Show Next Statement
Breakpoints... Ctrl+B
Insert/Remove Breakpoint F9
Enable/Disable Breakpoint Ctrl+F9
Disable All Breakpoints in current Target
Kill All Breakpoints in Current Target Ctrl+Shift+F9
OS Support
Execution Profiling
Memory Map...
Inline Assembly...
Function Editor (Open Ini File)...
Project
Project: Test
Target 1
Source Group 1
CMSIS
Device
startup_ARMCM0.
system_ARMCM0.
Project Books Func...
Build Output
Enter or leave a debug session
Simulation
L:14 C:18

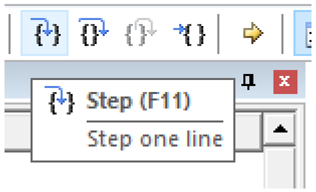
```
donly
Declare as entry poin
Ensures that __main addresses the following instruction.
Enable Debug
Make __main as global to access from startup file
Move 5 -> r0 (SR changes)
Move 10 -> r1 (SR do not change)
r1
```

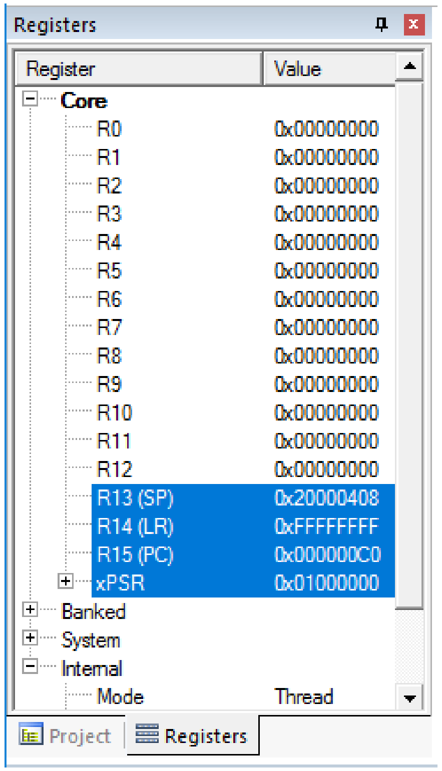
```
Build started: Project: Test
*** Using Compiler 'V5.06 update 6 (build 750)', folder: 'C:\Keil_v5\ARM\ARMCC\Bin'
Build target 'Target 1'
".\Objects\Test.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed:  00:00:00
```

Enjoy your new IDE…

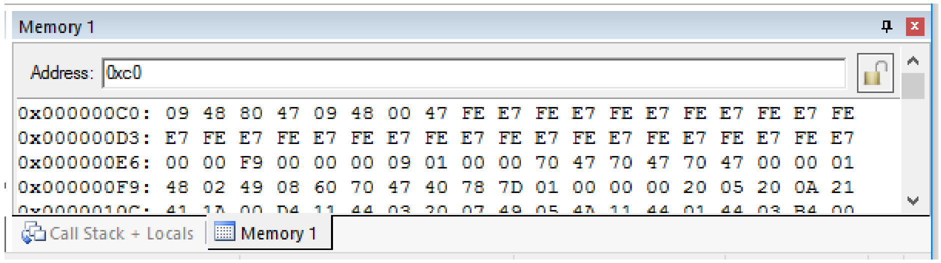

Insert/Remove Breakpoint (F9)
Insert or remove a breakpoint at the current line


Run (F5)
Start code execution


Step (F11)
Step one line

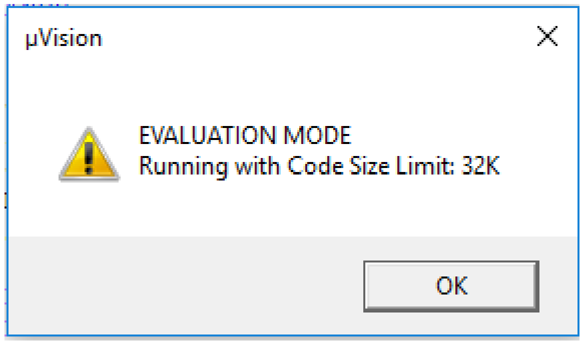

Registers

Register	Value
Core	
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)	0x20000408
R14 (LR)	0xFFFFFFFF
R15 (PC)	0x000000C0
xPSR	0x01000000
Banked	
System	
Internal	
Mode	Thread


Memory 1

Address: 0xc0

```
0x000000C0: 09 48 80 47 09 48 00 47 FE E7 FE E7 FE E7 FE E7 FE E7 FE
0x000000D3: E7 FE E7 FE E7 FE E7 FE E7 FE E7 FE E7 FE E7 FE E7 FE E7
0x000000E6: 00 00 F9 00 00 00 09 01 00 00 70 47 70 47 70 47 00 00 01
0x000000F9: 48 02 49 08 60 70 47 40 78 7D 01 00 00 00 20 05 20 0A 21
0x0000010C: 41 1A 00 D4 11 44 03 20 07 49 05 4A 11 44 01 44 03 B4 00
```


µVision

EVALUATION MODE
Running with Code Size Limit: 32K

If you get this error at the beginning of debugging, you are probably using free mode. Just click Ok and continue your work ☺

Refences

- http://www.keil.com/support/man/docs/armasm/
- http://infocenter.arm.com
- http://www.ece.utep.edu/courses/web3376/Directives.html