Sekolah:	**SEKOLAH KEJURUTERAAN ELEKTRIK**
Nama Matapelajaran: Kod Matapelajaran : SKEE 3732	Semakan : 1 Tarikh Keluaran : September 2016 Pindaan Terakhir : September 2020 No. Prosedur : **PK-UTM-SKE-(0)-10**

SEKOLAH KEJURUTERAAN ELEKTRIK
FAKULTI KEJURUTERAAN
UNIVERSITI TEKNOLOGI MALAYSIA
KAMPUS SKUDAI
JOHOR

SKEE 3732

MICROPROCESSOR LABORATORY

Laboratory 1: Using the Atmel Studio

Prepared by:	Approved by : Director of Programme
Dr Mohd Afzan Othman	Name:
Signature & Stamp: DR. MOHD AFZAN BIN OTHMAN Pensyarah Kanan Jabatan Elektronik dan Kejuruteraan Komputer Fakulti Kejuruteraan Elektrik Universiti Teknologi Malaysia 81310 Johor Bahru Johor, Malaysia	Signature & Stamp: AP IR. DR. RUBITA SUDIRMAN *Director (Electronics & Computer Engineering)* School of Electrical Engineering, Faculty of Engineering Universiti Teknologi Malaysia 81310 UTM Johor Bahru, Johor, MALAYSIA
Date: 19/10/2020	Date: 19/10/2020

I. PRELIMINARY PREPARATION (15 marks)

Important Note: You are required to do following BEFORE the lab session.

1. Install Atmel Studio 6.2. You can download from https://www.microchip.com/mplab/avr-support/avr-and-sam-downloads-archive or get a copy from the Microprocessor Laboratory.

Section A

2. Referring to Section B.2 Step B2 (Apendix B for Appendix B for SKEE3732 Laboratory 1 Sheet) , **CREATE** a new project and **WRITE** the program "Lab1Pre.asm" as given in **Figure B2(B5)** on Atmel Studio 6.

Do not CUT and PASTE from the original pdf documents. A zero mark (0) will be awarded for an obvious CUT and PASTE.

Use Section A.1 through A.9 of file "Appendix A for SKEE3732 Laboratory 1 Sheet.pdf" as reference for the theory on the instructions in Lab1Pre.

Organise the code of the program as guided in section A.13 of "Appendix A For Skee3732 Laboratory 1 Sheet.pdf". **(2 marks will be deducted if not complied)**

3. Compile the program to see if the program is free from errors. It can be done by clicking **Build > Build Solution** to compile the program. Take a screenshot as shown in Figure 1 below.

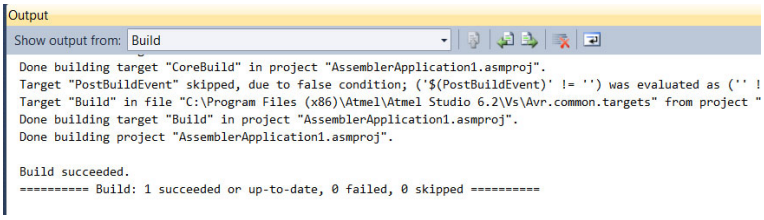

Figure 1

4. Print the Lab1Pre.asm program and screenshot of Figure 1 above and submit as your pre-lab.

Section B

5. Referring to section C.3, **WRITE** the program "Lab1PreCversion.ccp" given in Figure C3 of "Appendix C for SKEE3732 Laboratory 1 Sheet" on Atmel Studio 6.

Again Do not CUT and PASTE from the original pdf documents. A zero mark (0) will be awarded for an obvious CUT and PASTE.

Use file Section A and Section B of "Essential C.pdf" as reference for the theory on the statements in Lab1PreCversion.c.

Section C.1 and C.2 are addition reference that is specific for programming AVR's architecture using C/C++.

Organise the code of the program such that each structure in the codes is properly indented according to the levels. **(2 marks will be deducted if not complied).**

6. Compile the program to see if the program is free from errors. It can be done by clicking **Build > Build Solution** to compile the program. Take a screenshot of successfully compiled program.

7. Print the Lab1PreCversion.ccp program and screenshot of program build in step 6 and submit as your pre-lab.

II. LABORATORY SHEET

1 Title: Using the Atmel Studio 6: Programing in Assembly Language and C Language

2 Objective:

To use Atmel Studio 6:

1. To write, assemble and debug an assembly language program given in Figure B.2(B5) of "Appendix B for SKEE3732 Laboratory 1 Sheet".

   Describe (with evidence – snap shot) the operation completed by each process identified the assembly language program by debugging the program.

2. To write, compile and debug a given C language program given in Figure C.3 of "Appendix C for SKEE3732 Laboratory 1 Sheet" which perform the same function as the given assembly language program in Objective 1.

   Describe (with evidence – snap shot) the operation completed by each process identified the C program by debugging the program.

3. To compare the Assembly Language program generated by the C program in Objective 2 with the assembly language program in Objective 1.

   Use Table 1 and Table 2 to compare.

3 Equipment/Software/Reference:

1. A computer system running on Windows 10.
2. Atmel Studio 6.2 installed on the system. You may install Atmel Studio 7 if necessary.
3. Reference 1 – Atmega32 Reference manual
4. Reference 2 - AVR-Instruction-Set
5. Reference 3- AVR Assembler User Guide
6. Reference 4- Essential C.pdf

4 Procedures

Note: You must complete Preliminary Preparation before proceeding this section. Make sure that Atmel Studio 6.2 has been installed on your Window PC.

A. Procedure 1:

1. By referring to Section B.3 (Step B7 to B12) of "Appendix B for SKEE3732 Laboratory 1 Sheet", debug the Lab1pre.asm program on the Atmel Studio 6 and trace the output of each processes #.

   Fill Table 1 (column 1) as you debug through each process – there are 11 processes in total. Give evidence of the final result of each process identified in the program by taking snapshot of the ***Processor View*** and ***Memory View***. Highlight Register/Memory location affected the by completion of the process in snapshot taken.

2. To give evidence, take snap shots of relevant registers/memory locations affected **BEFORE** the process and **AFTER** the process is complete. These snapshots can be used to support "Discussion" materials in of your report. **Highlight the register or memory location affected in the snapshot** for the respective.

3. However, the relevant view to snapshot depends on what register or which memory location(s) that you need to read.

B. Procedure 2:

1. By referring to Section C.5 (Step C11) to Section C.8 (Step C17) of "Appendix C for SKEE3732 Laboratory 1 Sheet", debug the Lab1PreCversion.ccp program on the Atmel Studio 6 and trace the output of each processes #.

2. Use ***Watch View*** (Section C.8, Step 15) to see value of the affected variables and take snapshot **BEFORE** the process and **AFTER** the process is complete.

C. Procedure 3:

1. By referring to Section C.6, Step C13 of "Appendix C for SKEE3732 Laboratory 1 Sheet", determine the assembly language instruction generated by the Atmel Studio C/C++ from the Lab1PreCversion.ccp program, for each of the process given in Table 1 by inspecting the disassembly view. Fill Table 2.

5 Discussion:

Describe/discuss from result taken the following:

a. Describe briefly what the outcome of registers or memory address of each process for both codes written in assembly language and the codes written in C language

b. The differences between the codes written in assembly language and the codes written in C language and the use of GPRs in the codes.

c. The similarities or differences between assembly language codes written by programmer and C compiler generated assembly language code.

6 Conclusion

Conclude you report by citing facts given in result and discussion. You do not conclude by just by writing "Objective achieve". Referring/citing any evidence that is in Result or Discussion:

a. Conclude any similarity or difference the use of General Purpose Register as intermediate storage, or as pointer to locations between assembly Language and C language program.

b. Conclude advantage and disadvantages that you observe between assembly Language and C Language program.

7 Report Writing:

For Short Report:

Date of Laboratory Experiment

Make/Model/Serial No. of Machine or Platform/OS version of computer/software used must be given.

Theory: Just give titles relevant theory material used in the laboratory experiment.

Procedure: Report step taken to fulfill requirement in objectives including design of program to fulfill objective. Use flowchart if necessary. Procedure may be summarized.

Result: Report all result with evidences specified this laboratory sheet by taking snapshots of relevant Codes, Register, or Memory windows.

Discussion: Discuss technically on the procedures taken and result observed **based on Objectives, Theories and Results**.

Conclusion: Give Conclusion **based on Objectives, Theories, Procedures, Results and Discussions**.

For Long Report:

Date of Laboratory Experiment

Make/Model/Serial No. of Machine or Platform/OS version of computer/software used must be given.

Theory: A summary of the relevant theory material used in the laboratory experiment. Irrelevant and massive material that is a cut and paste from the internet will leads to deductions of mark.

Procedure: Give brief description of the steps taken by numbering steps.

Result: Report all result (citing the step number in Procedure) with evidences specified this laboratory sheet by taking snapshots of relevant Codes, Register, or Memory windows **based on objectives**.

Discussion: Discuss technically on the procedures taken and result observed **based on Objectives, Theories and Results**.

Conclusion: Give Conclusion **based on Objectives, Theories, Procedures, Results and Discussions**.

Table 1: To be filled for Procedure 1 and Procedure 3. (Information pre-filled is as guidance)

Process No	Final result in Memory Location or Register affected from Process (Value under Address column or Identifier name under Variable column must be filled if empty.	
	From Lab1Pre.asm	From Lab1PreCversion.c Identify affected variables. Address variables is not significant (Use the declared variable name in "Watch View" to get values of data in memory locations/registers
1	Register Content SP 085F	Refer to Processor view before Process #2 is executed SP 085D
2	Address Content 0x0060	Variable Content num1
3	Address Content _____	Variable Content temp1
4	Address Content _____	Variable Content sum
5	Register Content _____	Variable Content ptrnumbers3
6	Address Content 0x0080	Variable Content numbers3[0]
7	Address Content _____	Variable Content numbers3[3]
8	Address Content 0x0080 0x0081 0x0082 0x0083 0x0084	Variable Content numbers3[0] numbers3[1] numbers3[2] numbers3[3] numbers3[4]
9	Address Content _____ _____ _____ _____	Variable Content numbers2[0] numbers2[1] numbers2[2] numbers2[3]
10	Address Content _____ _____ _____ _____	Variable Content numbers4[0] numbers4[1] numbers4[2] numbers4[3]
11	Register Content PC	Cannot be traced (Step Over) under Debug
	Hint: You can find the value of address "here" from the Disassembly View	

Table 2: **To be filled for Procedure 1 and Procedure 3 .** (Information pre-filled is as guidance)

Process No	Final Result of each Process **Assembly language generated from rom Lab1PreCversion from author's computer may be different from your computer)**		
	From Lab1Pre.asm	**Assembly language generated from rom Lab1PreCversion (from author's computer)**	**Assembly language generated from rom Lab1PreCversion (from your computer)**

	`ldi R16,low(RAMEND)`
	`out SPL,R16`
	`ldi r16,high(RAMEND)`
	`out SPH,R16`
		LDI R28,0x5F
		LDI R29,0x08
		OUT 0x3E,R29
		OUT 0x3D,R28	

2	`ldi r17,$2a` `sts num1,r17`	**LDI R24,0x2A** **STS 0x007D,R24**	
3	`lds r17,num1` `sts temp1,r17`	**LDS R24,0x007D** **STS 0x007C,R24**	

4	`lds r0,num1`
	`lds r1,temp1`
	`add r0,r1`
	`ldi r16,5`
	`add r0,r16`
	`sts sum,r0`
		LDS R25,0x007D
		LDS R24,0x007C
		ADD R24,R25
		SUBI R24,0xFB
		STS 0x007B,R24	

5	`ldi YL,low(numbers3)` `ldi YH,high(numbers3)`	**LDI R30,0x74** **LDI R31,0x00** **STS 0x007A,R31** **STS 0x0079,R30**	
6	`ldi r16,$53` `st Y,r16`	**LDI R24,0x53** **STD Z+0,R24**	
7	`ldi r16,$35` `std Y+3,r16`	**LDI R24,0x35** **STD Z+3,R24**	

8	ldi XL,low(numbers3)
	ldi XH,high(numbers3)
	ldi r20,ten
	st X+,r20
	inc r20
	st X+,r20
	inc r20
	st X+,r20
	inc r20
	st X+,r20
		LDI R24,0x00
		LDI R25,0x00
		LDS R18,0x0060
		MOVW R30,R24
		SUBI R30,0x8C
		SBCI R31,0xFF
		STD Z+0,R18
		LDS R18,0x0060
		SUBI R18,0xFF
		STS 0x0060,R18
		ADIW R24,0x01
		CPI R24,0x05
		CPC R25,R1
		BRNE PC-0x0E	

9	ldi XL,low(numbers2+4)
	ldi XH,high(numbers2+4)
	ld r16,Y+
	st -X,r16
	ld r16,Y+
	st -X,r16
	ld r16,Y+
	st -X,r16
	ld r16,Y+
	st -X,r16
		LDS R24,0x0074
		STS 0x0070,R24
		LDS R24,0x0075
		STS 0x006F,R24
		LDS R24,0x0076
		STS 0x006E,R24
		LDS R24,0x0077
		STS 0x006D,R24	

10	ldi ZH,high(numbers1<<1)
	ldi ZL,low(numbers1<<1)
	
	ldi YH,high(numbers4)
	
	ldi YL,low(numbers4)
	ldi r17,4
	repeat:
	lpm r18,Z+
	st Y+,r18
	dec r17
	brne repeat
		LDI R24,0x00
		LDI R25,0x00
		MOVW R30,R24
		SUBI R30,0xAC
		SBCI R31,0xFF
		LPM R30,Z
		
		STS 0x006C,R30
		LDS R18,0x006C
		MOVW R30,R24
		SUBI R30,0x9E
		SBCI R31,0xFF
		STD Z+0,R18
		ADIW R24,0x01
		CPI R24,0x04
		CPC R25,R1
		BRNE PC-0x0F	

11	here: rjmp here	**RJMP PC-0x0000**	