Fakulti:	**FAKULTI KEJURUTERAAN ELEKTRIK**
Nama Matapelajaran: Kod Matapelajaran : SKEE 3732	Semakan : 1 Tarikh Keluaran : September 2016 Pindaan Terakhir : 09 Oktober 2017 No. Prosedur : **PK-UTM-FKE-(0)-10**

FAKULTI KEJURUTERAAN ELEKTRIK
UNIVERSITI TEKNOLOGI MALAYSIA
KAMPUS SKUDAI
JOHOR

SKEE 3732

MICROPROCESSOR LABORATORY

Laboratory 2: Digital Input/Output and Timer on ATmega32.

Prepared by: ZURAIMI BIN YAHYA Signature & Stamp: Date:	Approved by : Head of Department Name: Signature & Stamp: Date:

I. PRELIMINARY REPORT (15 marks)

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

1. Files "Appendix C For SKEE3732 Laboratory 1 Sheet.pdf", "Atmega32 Reference manual.pdf" and Appendix A For SKEE3732 Laboratory 2 Sheet.pdf" are required as reference for this Laboratory.

2. You should be able to create, build and debug an ATmega32A program using Atmel Studio 6 which has been covered in Laboratory 1 (refer Section C.3. of "Appendix C For SKEE3732 Laboratory 1 Sheet.pdf").

3. Referring to the program "`Lab2Exp1.c`" in Figure A.8(d) of "Appendix A For SKEE3732 Laboratory 2 Sheet.pdf", create the GCC C executable project named "`Lab2Exp1`" for device "ATmega32A" and write the program.

   You may cut and paste but format the structure indentation of the program as in original codes given in Figure A.8(d) of "Appendix A For SKEE3732 Laboratory 2 Sheet.pdf" to allow you to read the program structure thus the program logic easily.

   You are needed to read comments and refer Appendices stated in the "`Lab2Exp1.c`", to understand the program which is necessary and beneficial in this lab and the course SKEE3223.

4. Referring to Table A.8(a) of "Appendix A For SKEE3732 Laboratory 2 Sheet.pdf" and the execution of following statements found in the program "Lab2Exp1.c", which will connect the respective peripherals lines respective peripherals lines to the ATmega32 CPU.

```
#define OutPort 0xff //also as 0b11111111
#define InPort 0x00  //also as 0b0000000
.................
DDRD = 0x00;  //Set Port D as Input
PORTD = 0xff; //Enable pull-up resistor
.................
DDRA=OutPort;
.................
DDRB=OutPort;
DDRC=OutPort;
```

   Fill in Table 1 the name of the port register (or bits) on the ATmega32A microcontroller that identifies the connections to the respective peripherals lines. Some boxes are filled for your guidance. Identify also statement(s) that when executed, will initialised the direction of data in the connections.

   Assume that on RESET (or Restart) all Digital I/O registers are cleared (logic 0). So DDRC is assumed cleared on RESET.

   Refer section A.1 for of "Appendix A For SKEE3732 Laboratory 2 Sheet.pdf" and "AVR Instruction Set Reference manual".

5. Referring to the flowchart in Figure A.8(c) of "Appendix A For SKEE3732 Laboratory 2 Sheet.pdf", identify the codes in "`Lab2Exp1.c`" that implements the process as specified in Table 2.

6. Referring to section A.3 of "Appendix A For SKEE3732 Laboratory 2 Sheet.pdf", fill in under the column "the codes in program Lab2Exp1.c" of Table 3, the respective instruction implement the process under the "Function" column which implemented in function "DelaymsUsingTimer2" of program "Lab2Exp1.c".

7. Determine the code in program "Lab2Exp1.c" that sends the data to the SEGMENT data of the seven segment display panel.

8. Determine the code in program "Lab2Exp1.c" that selects Display DS1 of the seven segment display panel to display the data received by SEGMENT data.

9. Show in Table 4 the calculations that generate the delay by subroutine "Delay250ms". Omit the negligible delay generated by overhead instructions in the subroutine.

Use the following formula given in Figure A.2(a)(iii) of "Appendix A For SKEE3732 Laboratory 2 Sheet.pdf"

$$\text{Time Count} = \frac{\text{Required Delay}}{\text{Clock Time Period}} - 1$$

Table 1 (Questions for No. 4 of Preliminary Report)

Peripheral Lines		**Port or bit connected on Atmega32A**	**The statement that implement the connection (if any)**
Individual Switch	SW7	PIND7	
	SW6		
	SW5		
	SW4		
8 bits LED panel connector			

Table 2 **(Questons for No. 5 of Preliminary Report)**

<table>
<tr><th colspan="3">Process</th><th>Fragment of the code in program “Lab2Exp1.c” (the statements that execute the complete process)</th></tr>
<tr><td colspan="3">Initialise peripherals’ controls</td><td>DDRD = 0x00;
PORTD = 0xff;</td></tr>
<tr><td colspan="3">Initialise environment variables</td><td></td></tr>
<tr><td rowspan="8">Shell Loop</td><td colspan="2"></td><td>while(1)
{</td></tr>
<tr><td colspan="2">Get Command (Any switched pressed</td><td>SWData=PIND|MaskofLowerNibbleHigh;</td></tr>
<tr><td rowspan="5">Execute Command (if a switch is pressed)</td><td>Get Switch Data</td><td>switch (SWData)</td></tr>
<tr><td>If SW7 Pressed</td><td>DoRunningLight();</td></tr>
<tr><td>If SW6 Pressed</td><td>Processed by “default” condition thus “break” is executed</td></tr>
<tr><td>If SW5 Pressed</td><td></td></tr>
<tr><td>If SW4 Pressed</td><td></td></tr>
<tr><td colspan="2">Display prompt (Indicating “Alive”)(If no switch is pressed)</td><td></td></tr>
<tr><td colspan="3"></td><td>}</td></tr>
</table>

Table 3 (Questions for No. 6 of Preliminary Report)

Function	**The codes in program "Lab2Exp1.c" (Do not include the comments)**
Set Starting count in TCNT2	`TCNT2 = 255-CountforDelayms;`
Start Timer which by setting prescaler	
Wait for TOV2 bit (overflow bit) is set (while TCNT2 counts up in background)	
Stop Timer	
When TOV2 bit is set, Clear TOV2 bi	

Table 4 (Questions for No. 9 of Preliminary Report)

Calculatation of Delay.	
TCNT2 initialised in Delay250ms()=	
Timer Count =	
F_CPU=	
Prescaler=	
Period of Timer Clock=	
Delay=	

II. LABORATORY SHEET

1 Title: Using Input/Ouput, Timer on ATmega32A Target Board.

2 Objective:

1. To connect Atmel Studio to an ATmega32A target board via a JTAGICE mkII debugger.
2. To inspect a given ATmega32A firmware using polling method to receive command.
3. To use debugging method to inspect a delay operation.
4. To add additional code to upgrade the operation of firmware using polling method to receive additional commands by polling and execute the commands.

3 Equipment/Software/Reference:

1. A computer system running either Window 7 or Window 8.
2. Atmel Studio 6 installed on the system.

   Atmel Studio 7 (as-installer-7.0.634-full.exe) is needed for Windows 10.
3. AVR JTAGICE mkII Debugger
4. Gotronik Atmega32A Target Board
5. Reference 1 - Appendix C For SKEE3732 Laboratory 1 Sheet.pdf
6. Reference 2- Appendix A For SKEE3732 Laboratory 2 Sheet.pdf
7. Reference 3– Atmega32 Reference manual **.pdf**
8. Reference 4– Embedded C Programming and the Atmel AVR, Second Edition.pdf
9. Reference 5– AVR libc function reference.pdf

All softcopy of the references can be sourced in the “Reference For Laboratory 2” sub-folder of the folder site where this Laboratory 2 Sheet is located.

4 Procedures

Note: You must complete Preliminary Preparation before proceeding this section. Make sure that AVR Studio 6.2 has been installed on your window system.

I. To Open Project (create during pre-lab session)

Open Project named "Lab2Exp1" which you have created as instructed in section 1 of "Preliminary Report" by double clicking "Lab2Exp1.atsln" in the "Lab2Exp1" directory

II. To Wire up peripherals on Target board

Referring to Section A.8 of "Appendix A For SKEE3732 Laboratory 2 Sheet" connect the wiring of the peripherals to ATmega32A chip as specified in Table A.8(a).

III. To connect Atmel Studio Project "Lab2Exp1" to an ATmega32A target board via a JTAGICE mkII debugger (Software connection of current project in Atmel Studio to the JTAGICE MkII)

Referring to Section A.9 of "Appendix A For SKEE3732 Laboratory 2 Sheet":

a. Connect project "Lab2Exp1" to the JTAGICE mkII and Target board

b. Select "Selected Debugger/programmer" as JTAGICE mkII and the Interface is JTAG.

c. Confirm that the CPU frequency (F_CPU) is 1 MHz

IV. Programming "Lab2Exp1" project to the ATmega32A chip

a. Referring to A.9(e) of "Appendix A For SKEE3732 Laboratory 2 Sheet.pdf", program the Atmega32A chip after successfully compiling the project "Lab2Exp1".

b. Report result as required In Table 5.

V. To use debugging process (with the assistance of flowchart in Figure A.8(c) of "Appendix A For SKEE3732 Laboratory 2 Sheet.pdf") to identify the program flow of the "Lab2Exp1.c".

a. Referring to Table 5, if the board is functioning according to the specification in section A.8(b)(v) of "Appendix A For SKEE3732 Laboratory 2 Sheet.docx", select Debug → Reset (or Restart).

b. Referring to Table 2, set breakpoints (as shown in Figure 2) on the first statement of the code that execute commands from Push Button presses, assuming that you have filled correctly the empty boxes of Table 2. Check with supervising lecturer to confirm the correctness of your answer. Refer to Section A.10 on guidance how to implement this procedure.

c. Record the result based on the condition given in Table 6.

VI. Confirming Actual Delay Timer with Calculated Timer Delay in Table 4 (Question for No. 9 of Preliminary Report).

Refer Section A.10(d) of **"**Appendix A For SKEE3732 Laboratory 2 Sheet.docx" for the procedure to confirm actual Delay.

VII. Implementing the Complete Function of "Lab2Exp1.c" as required in Table A.8(b).

Modify the program such that the program will executes the pattern sequence as specified in Table A.8(b) for command if SW6 and SW4 is pressed. Run the program after modification and get an endorsement in **Figure 1** from the supervising lecturer after showing the working program.

Hint: You will need an additional Function and Case conditions.

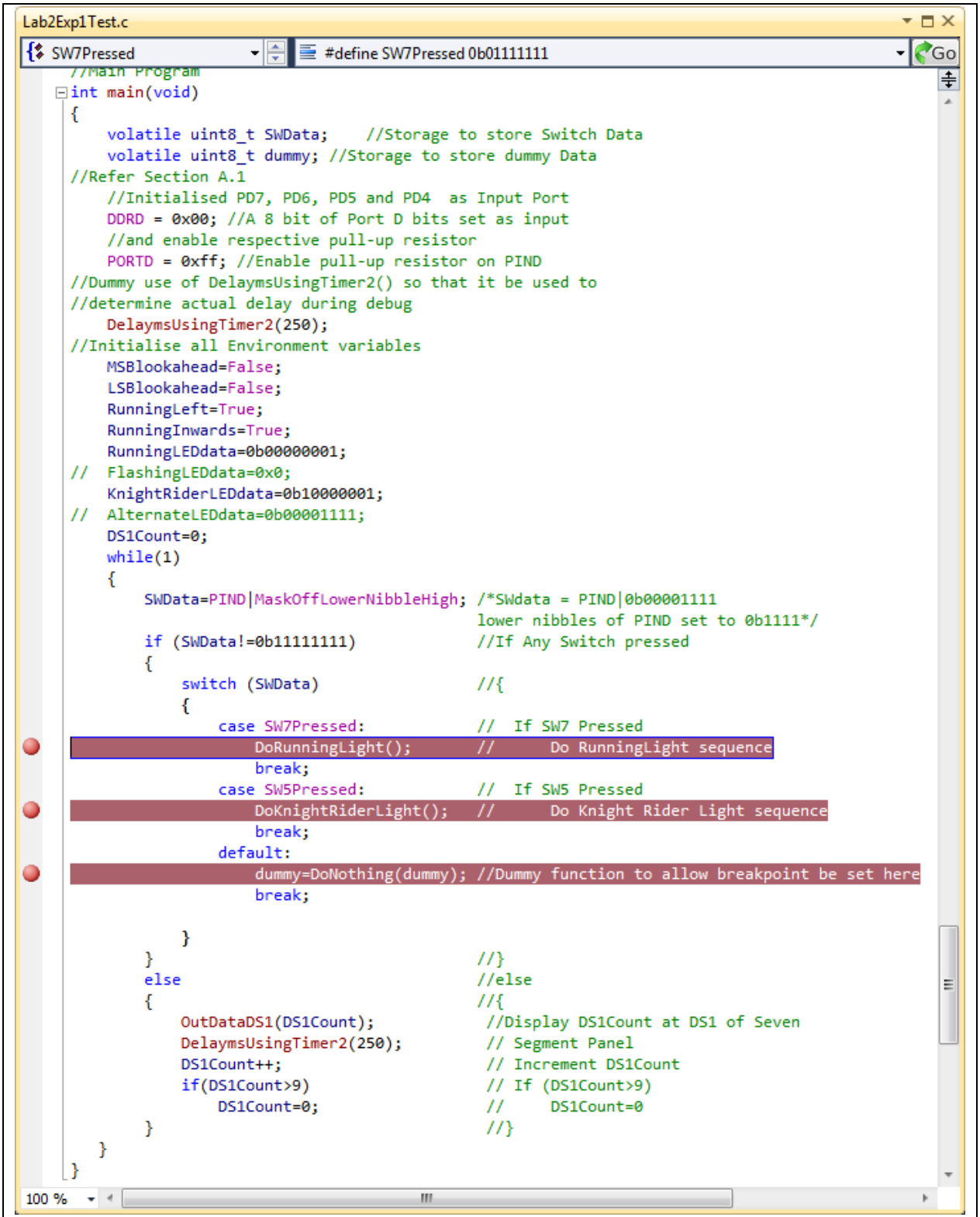
Lab2Exp1Test.c
SW7Pressed
#define SW7Pressed 0b01111111
Go
//Main Program
int main(void)
{
volatile uint8_t SWData; //Storage to store Switch Data
volatile uint8_t dummy; //Storage to store dummy Data
//Refer Section A.1
//Initialised PD7, PD6, PD5 and PD4 as Input Port
DDRD = 0x00; //A 8 bit of Port D bits set as input
//and enable respective pull-up resistor
PORTD = 0xff; //Enable pull-up resistor on PIND
//Dummy use of DelaymsUsingTimer2() so that it be used to
//determine actual delay during debug
DelaymsUsingTimer2(250);
//Initialise all Environment variables
MSBlookahead=False;
LSBlookahead=False;
RunningLeft=True;
RunningInwards=True;
RunningLEDdata=0b00000001;
// FlashingLEDdata=0x0;
KnightRiderLEDdata=0b10000001;
// AlternateLEDdata=0b00001111;
DS1Count=0;
while(1)
{
SWData=PIND|MaskOffLowerNibbleHigh; /*SWdata = PIND|0b00001111
lower nibbles of PIND set to 0b1111*/
if (SWData!=0b11111111) //If Any Switch pressed
{
switch (SWData) //{
{
case SW7Pressed: // If SW7 Pressed
DoRunningLight(); // Do RunningLight sequence
break;
case SW5Pressed: // If SW5 Pressed
DoKnightRiderLight(); // Do Knight Rider Light sequence
break;
default:
dummy=DoNothing(dummy); //Dummy function to allow breakpoint be set here
break;
}
} //}
else //else
{ //{
OutDataDS1(DS1Count); //Display DS1Count at DS1 of Seven
DelaymsUsingTimer2(250); // Segment Panel
DS1Count++; // Increment DS1Count
if(DS1Count>9) // If (DS1Count>9)
DS1Count=0; // DS1Count=0
} //}
}
}
100 %

Figure 2

Table 5: Result to be taken in Procedure IV (some are filled for your guidance)

Input	Describe Display at Seven Segment Panel	Describe Pattern generated at LED Panel
No Push Button pressed		
SW7 pressed		
SW6 pressed	Counts Stop	Do not give any effect (changes)
SW5 pressed		
SW4 pressed		
SW3 pressed	Do not give any effect (changes)	Do not give any effect (changes)
SW2 pressed		
SW1 pressed		
SW0 pressed		

Explain why when either one of SW3 or SW2 is pressed, there is no effect at all in the program (Hint: The statement "PIND|MaskOffLowerNibbleHigh") You may debug the flow when one of the above switches is pressed.	
Explain why when either one of SW6 is pressed, there is no effect at all in the program. (Hint: The statements "PIND|MaskOffLowerNibbleHigh" and the "SWITCH". You may debug the flow when the above switches is pressed.	

Table 6: Result of Procedure V (some are filled for your guidance)

Condition	**Statement where Program breaks**	**Content of SWdata**
SW7 **pressed**	`DoRunningLight`	0x7f (0b01111111)
SW6 **pressed**		
SW5 **pressed**		
SW4 pressed		

Table 7:

Operation	**Result**
Stop Watch Counter at statement "DelaymsUsingTimer2"	
Stop Watch Counter at statement after "DelaymsUsingTimer2"	
Difference of Stop Watch Counter between the two statement	
Calculated delay from Table 4	
Explain the difference between the calculate delay and actual delay	

Wirite the additional code and its location in the program "Lab2Exp1.c" that need to be made to implement Procedure VII .

You may Cut and Paste from modified program:

Execution of Command when SW6 pressed: Working/Not working

Execution of Command when SW4 pressed: Working/Not working

Name and Signature of Lecturer:

Figure 1

5 Report Writing

Title:

Objective:

Equipment/Software Used:

Procedure:

Written in third person (reporting) speech:

Result:

Fill Table and Figures for Result and attach with result. Readings from I/O views, Watch etc. will require snapshot as evidence.

Discussion:

Discuss based on Objective and Result. There shouldn't be any alien Objective and Result.

Conclusion:

Conclude based on Objective, Result and Discussion.