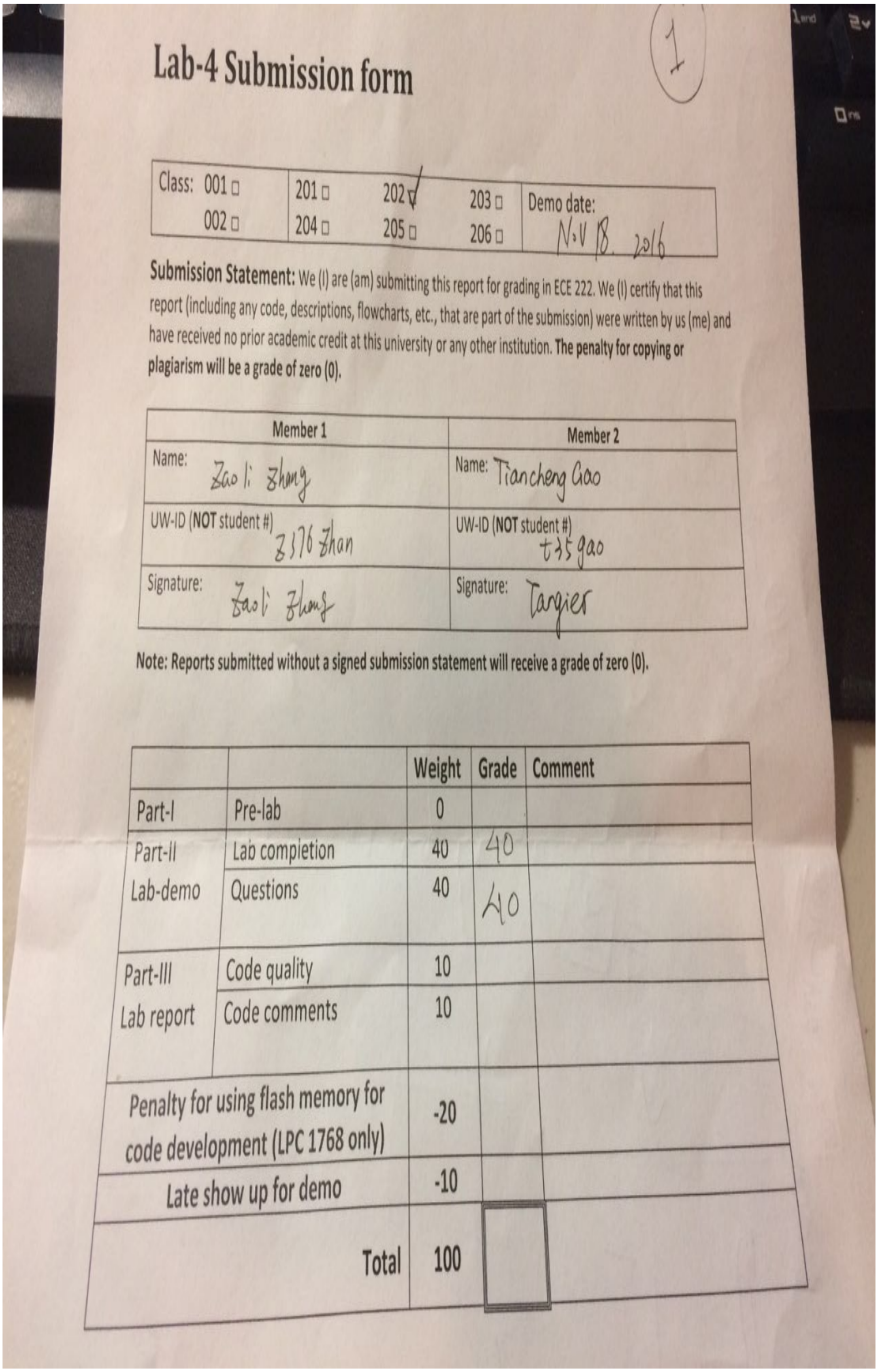
LAB4 postlab

```
; ECE-222 Lab ... Winter 2013 term
; Lab 4  code
                        THUMB
            ; Declare THUMB instruction set
                        AREA   My_code, CODE, READONLY          ;
                        EXPORT            __MAIN
            ; Label __MAIN is used externally
                        EXPORT            EINT3_IRQHandler      ;
without this the interupt routine will not be found

                        ENTRY

__MAIN

; The following lines are similar to previous labs.
; They just turn off all LEDs
                        LDR               R10, =LED_BASE_ADR
      ; R10 is a  pointer to the base address for the LEDs
                        MOV         R3, #0xB0000000         ; Turn off
three LEDs on port 1
                        STR         R3, [R10, #0x20]
                        MOV         R3, #0x0000007C
                        STR         R3, [R10, #0x40]    ; Turn off five
LEDs on port 2
                        MOV               R0, #0X0                  ;
initial all registors which we need
                        MOV               R4, #0X0
                        MOV               R7, #0X0
                        MOV               R8, #0X0
                        MOV               R12, #0X0

                        LDR               R3, =ISER0
                        MOV               R2, #0x00200000  ;21th
bit ,active the port---1
                        STR               R2, [R3]
                        LDR               R3, =IO2IntEnf
                        MOV               R2, #0x00000400  ;10th bit,
active the port ---1
                        STR               R2, [R3]

; This line is very important in your main program
; Initializes R11 to a 16-bit non-zero value and NOTHING else can write to R11 !!
                        MOV               R11, #0xABCD              ;
Init the random number generator with a non-zero number
LOOP              BL                RNG
                        MOV               R6, R11
      ; we do not change the random number but use another registor to save
it.
                        LSL               R6, #16 ; clear garbage bits
                        LSR               R6, #16
                        MOV               R9, #3
                        MUL               R6, R6, R9          ;
                        MOV         R9, #1000                       ;5-
25s delay
                        UDIV        R6,R6,R9                  ; get value
from 0-200
                        ADD         R6, #50; 50-250, 0.1s → 5s-25s
LOOP2             BL                DISPLAY_NUM
                        MOV               R0,#10
                        BL        DELAY
                        SUBS        R6, #10             ;R6 counter
decrease every 1 s
                        CMP               R6, #0
                        BGT               LOOP2
                        MOV               R9, #0        ;flag for ISR
                         B                FLASH         ;R6 less or equal
to 0

FLASH                   MOV               R11, #0       ;
                        STR               R11, [R10,#0x54]
                        STR               R11, [R10,#0x34] ; flsh
                        MOV               R0,#1
                        CMP               R9, #0 ; when ISR in this
subroutine, we need it come back
                        BNE               LOOP2
                        BL                DELAY
                        MOV               R11, #0x0000007C
                        STR               R11, [R10,#0x54]
                        MOV               R11, #0xB0000000
                        STR               R11, [R10,#0x34]
                        MOV               R0,#1
                        CMP               R9, #0 ; check it again
                        BNE               LOOP2
                        BL                DELAY
                        B                 FLASH
            ;
            ; Your main program can appear here
            ;


DISPLAY_NUM       STMFD       R13!,{R3-R5, R14}     ;display R3
                        MOV               R3, R6
                        MOV               R4, #0
      ; two inital registor we need to use
                        MOV               R5, #0
                        BFI               R5,R3,#0, #5
      ;Replace the bits from 0th to 4th(5 bits) in R5 by R3, which using the bits
in R3 from 0th to the 4th(5 bits). R9 contains the counter value
                        RBIT        R5,R5                           ;
reverse the bits for fitting the LED
                        LSR               R5,#25
            ;when we reverse, the last 5 significant bits become the most 5
signficant bits, so we shoudl shift right 25 to get bit 2th to 6th.
                        LSR               R3,#5
      ; need rest 3 bits
                        STR               R5, [R10,#0x54]
      ; port 2.2-2.6
```

```
                                MOV                     R5,#0
        ; reuse R5 again
                                BFI                     R5,R3, #0,#1                    ;
we first take 1 bits, 30th bits we do need it but we have to show it
                                LSL                     R3, #1
        ; get 0 in
                                ADD                     R3, R3,R5
        ; put the rest three bits in the R5
                                BFI                     R5,R3,#0,#4
        ; Replace the bits from 0th to 3th(4 bits) in R5 by R3, which using the bits
in R3 from 0th to the 3th(4 bits).
                                RBIT            R5,R5
        ;reverse
                                STR                     R5,[R10,#0x34]
        ; show the value

            LDMFD               R13!,{R3-R5, R15}
;*-----------------------------------------------------------------
-
; Subroutine RNG ... Generates a pseudo-Random Number in R11
;*-----------------------------------------------------------------
-
; R11 holds a random number as per the Linear feedback shift register (Fibonacci)
on WikiPedia
; R11 MUST be initialized to a non-zero 16-bit value at the start of the program
; R11 can be read anywhere in the code but must only be written to by this
subroutine
RNG                     STMFD           R13!,{R1-R3, R14}       ; Random Number
Generator
                                AND                     R1, R11, #0x8000
                                AND                     R2, R11, #0x2000
                                LSL                     R2, #2
                                EOR                     R3, R1, R2
                                AND                     R1, R11, #0x1000
                                LSL                     R1, #3
                                EOR                     R3, R3, R1
                                AND                     R1, R11, #0x0400
                                LSL                     R1, #5
                                EOR                     R3, R3, R1                      ;
The new bit to go into the LSB is present
                                LSR                     R3, #15
                                LSL                     R11, #1
                                ORR                     R11, R11, R3
                                LDMFD                   R13!,{R1-R3, R15}

;*-----------------------------------------------------------------
-
; Subroutine DELAY ... Causes a delay of 1ms * R0 times
;*-----------------------------------------------------------------
-
;               aim for better than 10% accuracy
DELAY                   STMFD           R13!,{R2,R4,R10, R14}
DELAY1                          MOV                     R4, #0x0084     ; 0.1s counter
                                MOV                     R10, #1000
                                MUL                     R4,R4,R10
```

```
DELAY_LOOP          SUBS            R4, #1                  ; two inserted for loops,
satisfies the 0.1ms *R0
                           BNE                     DELAY_LOOP

delay2                 SUBS            R0, #1
                           BNE                     DELAY1
                           B                        exitDelay
exitDelay            LDMFD                R13!,{R2,R4,R10, R15}

; The Interrupt Service Routine MUST be in the startup file for simulation
;   to work correctly.  Add it where there is the label "EINT3_IRQHandler
;
;*-------------------------------------------------------------------
-
; Interrupt Service Routine (ISR) for EINT3_IRQHandler
;*-------------------------------------------------------------------
-
; This ISR handles the interrupt triggered when the INT0 push-button is pressed
; with the assumption that the interrupt activation is done in the main program
EINT3_IRQHandler

                           STMFD          R13!,{R1-R4, R14}

                           LDR                     R7, =IO2INTCLR
                           MOV                     R8, #0x00000400
                           STR                     R8, [R7]

                           MOV                     R11, #0xABCD
                           AND                     R1, R11, #0x8000
                           AND                     R2, R11, #0x2000
                           LSL                     R2, #2
                           EOR                     R3, R1, R2
                           AND                     R1, R11, #0x1000
                           LSL                     R1, #3
                           EOR                     R3, R3, R1
                           AND                     R1, R11, #0x0400
                           LSL                     R1, #5
                           EOR                     R3, R3, R1                ;
The new bit to go into the LSB is present
                           LSR                     R3, #15
                           LSL                     R11, #1
                           ORR                     R11, R11, R3

                           MOV                     R6, R11
      ; we do not change the random number but use another registor to save
it.
                           LSL                     R6, #16
                           LSR                     R6, #16
                           MOV                     R9, #3
                           MUL                     R6, R6, R9            ;
                           MOV             R9, #1000                          ;5-
25s delay
                           ADD                     R6, #50
                           UDIV            R6,R6,R9                    ; random
number R6
```

```
                              MOV                    R9, #1 ; flag to check where
and how to come back

                              LDMFD                  R13!,{R1-R4, R15}
;*-------------------------------------------------------------------------
-
; Below is a list of useful registers with their respective memory addresses.
;*-------------------------------------------------------------------------
-
LED_BASE_ADR          EQU     0x2009c000             ; Base address of the memory
that controls the LEDs
PINSEL3                       EQU    0x4002C00C          ; Pin Select Register 3
for P1[31:16]
PINSEL4                       EQU    0x4002C010          ; Pin Select Register 4
for P2[15:0]
FIO1DIR                       EQU            0x2009C020          ; Fast Input
Output Direction Register for Port 1
FIO2DIR                       EQU            0x2009C040          ; Fast Input
Output Direction Register for Port 2
FIO1SET                       EQU            0x2009C038          ; Fast Input
Output Set Register for Port 1
FIO2SET                       EQU            0x2009C058          ; Fast Input
Output Set Register for Port 2
FIO1CLR                       EQU            0x2009C03C          ; Fast Input
Output Clear Register for Port 1
FIO2CLR                       EQU            0x2009C05C          ; Fast Input
Output Clear Register for Port 2
IO2IntEnf             EQU            0x400280B4          ; GPIO Interrupt Enable
for port 2 Falling Edge
ISER0                 EQU            0xE000E100          ; Interrupt Set-Enable
Register 0
IO2INTCLR             EQU            0x400280AC     ; Interrupt Port 2 Clear
Register

                              ALIGN

                              END
```



1

Lab-4 Submission form

Class: 001 □	201 □	202 ☑	203 □	Demo date:
002 □	204 □	205 □	206 □	NoV 18. 2016

Submission Statement: We (I) are (am) submitting this report for grading in ECE 222. We (I) certify that this report (including any code, descriptions, flowcharts, etc., that are part of the submission) were written by us (me) and have received no prior academic credit at this university or any other institution. **The penalty for copying or plagiarism will be a grade of zero (0).**

Member 1	Member 2
Name: Zao li Zhang	Name: Tiancheng Gao
UW-ID (**NOT** student #) z376zhan	UW-ID (**NOT** student #) t35gao
Signature: Zao li Zhang	Signature: Targier

Note: Reports submitted without a signed submission statement will receive a grade of zero (0).

		Weight	Grade	Comment
Part-I	Pre-lab	0		
Part-II Lab-demo	Lab completion	40	40	
	Questions	40	40	
Part-III Lab report	Code quality	10		
	Code comments	10		
Penalty for using flash memory for code development (LPC 1768 only)		-20		
Late show up for demo		-10		
	Total	100		