Leonardo Fusser (1946995) November 12th, 2020

Microcontroller & Microprocessor Systems

Lab 4

Day Yann Fong

PART A:

3)

Address	Name	Value	Field	Option	Category	Setting
2007	CONFIG1	23F5	FOSC	INTRC_CLKOUT	Oscillator Selection bits	INTOSC oscillator: CLKOUT function on RA6/OSC2/CLKOUT pin, I/O function on RA7/OSC1/CLKIN
			WDTE	OFF	Watchdog Timer Enable bit	WDT disabled and can be enabled by SWDTEN bit of the WDTCON register
			PWRTE	OFF	Power-up Timer Enable bit	PWRT disabled
			MCLRE	ON	RE3/MCLR pin function select bit	RE3/MCLR pin function is MCLR
			CP	OFF	Code Protection bit	Program memory code protection is disabled
			CPD	OFF	Data Code Protection bit	Data memory code protection is disabled
			BOREN	ON	Brown Out Reset Selection bits	BOR enabled
			IESO	OFF	Internal External Switchover bit	Internal/External Switchover mode is disabled
			FCMEN	OFF	Fail-Safe Clock Monitor Enabled bit	Fail-Safe Clock Monitor is disabled
			LVP	OFF	Low Voltage Programming Enable bit	RB3 pin has digital I/O, HV on MCLR must be used for programming
2008	CONFIG2	3FFF	BOR4V	BOR40V	Brown-out Reset Selection bit	Brown-out Reset set to 4.0V
			WRT	OFF	Flash Program Memory Self Write Enable bits	Write protection off

Config bits generator window

```
#include "p16f887.inc"

; CONFIG1
; __config 0x23F5
 __CONFIG _CONFIG1, _FOSC_INTRC_CLKOUT & _WDTE_OFF & _PWRTE_OFF & _MCLRE_ON & _CP_OFF & _CPD_OFF & _BOREN_ON & _IESO_OFF & _FCMEN_OFF & _LVP_OFF
; CONFIG2
; __config 0x3FFF
 __CONFIG _CONFIG2, _BOR4V_BOR40V & _WRT_OFF
```

Config bits code generated from above and put in assembly code

Configuration directive settings	
In config bits generator	*In code (stored at address 2007)*
FOSC: INTRC_CLKOUT =>	_FOSC_INTRC_CLKOUT
WDTE: OFF =>	_WDTE_OFF
IESO: OFF =>	_IESO_OFF
FCMEN: OFF =>	_FCMEN_OFF
LVP: OFF =>	_LVP_OFF

4a) It is not possible to generate a simple delay loop (without nested) to iterate more than 255 times because 255 (binary = 11111111) is the highest possible number that can be processed in the PIC's registers (8 bits).

5a) The expected frequency to be observed at the CLOCKOUT pin is 1 MHz. It is 1 MHz because internal clock is running at 4 MHz and as it is configured (for INTOSC mode – set in config bits) the CLOCKOUT pin will output the internal oscillator frequency divided by 4. So,

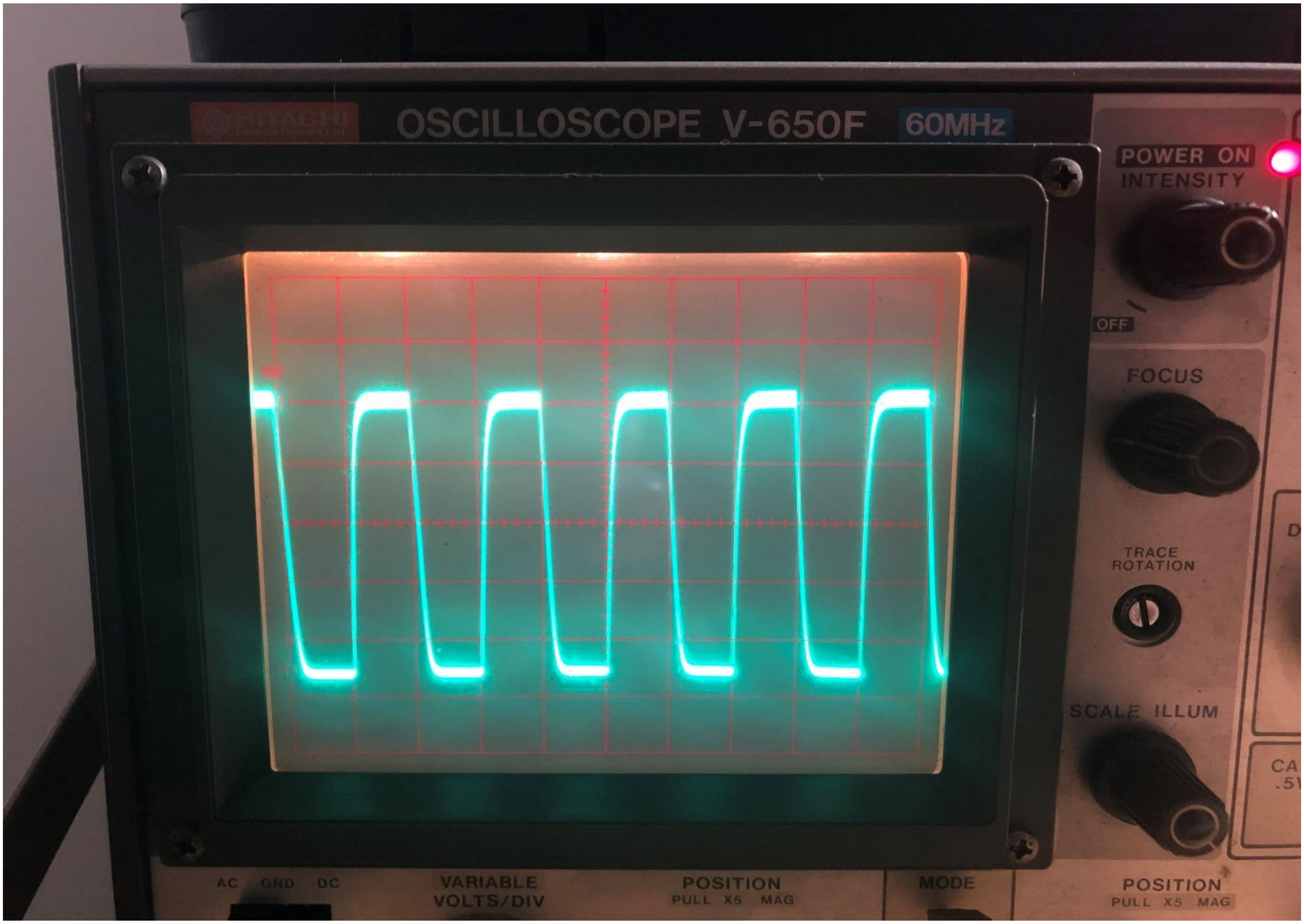
$$CLOCKOUT = \frac{F_{OSC}}{4} = \frac{4\ MHz}{4} = 1\ MHz$$

7. INTOSC – Internal oscillator with FOSC/4 output on OSC2 and I/O on OSC1/CLKIN.

In **INTOSC** mode, OSC1/CLKIN is available for general purpose I/O. OSC2/CLKOUT outputs the selected internal oscillator frequency divided by 4. The CLKOUT signal may be used to provide a clock for external circuitry, synchronization, calibration, test or other application requirements.

Explanation found in PIC 16F887 datasheet

Leonardo Fusser (1946995) November 12th, 2020



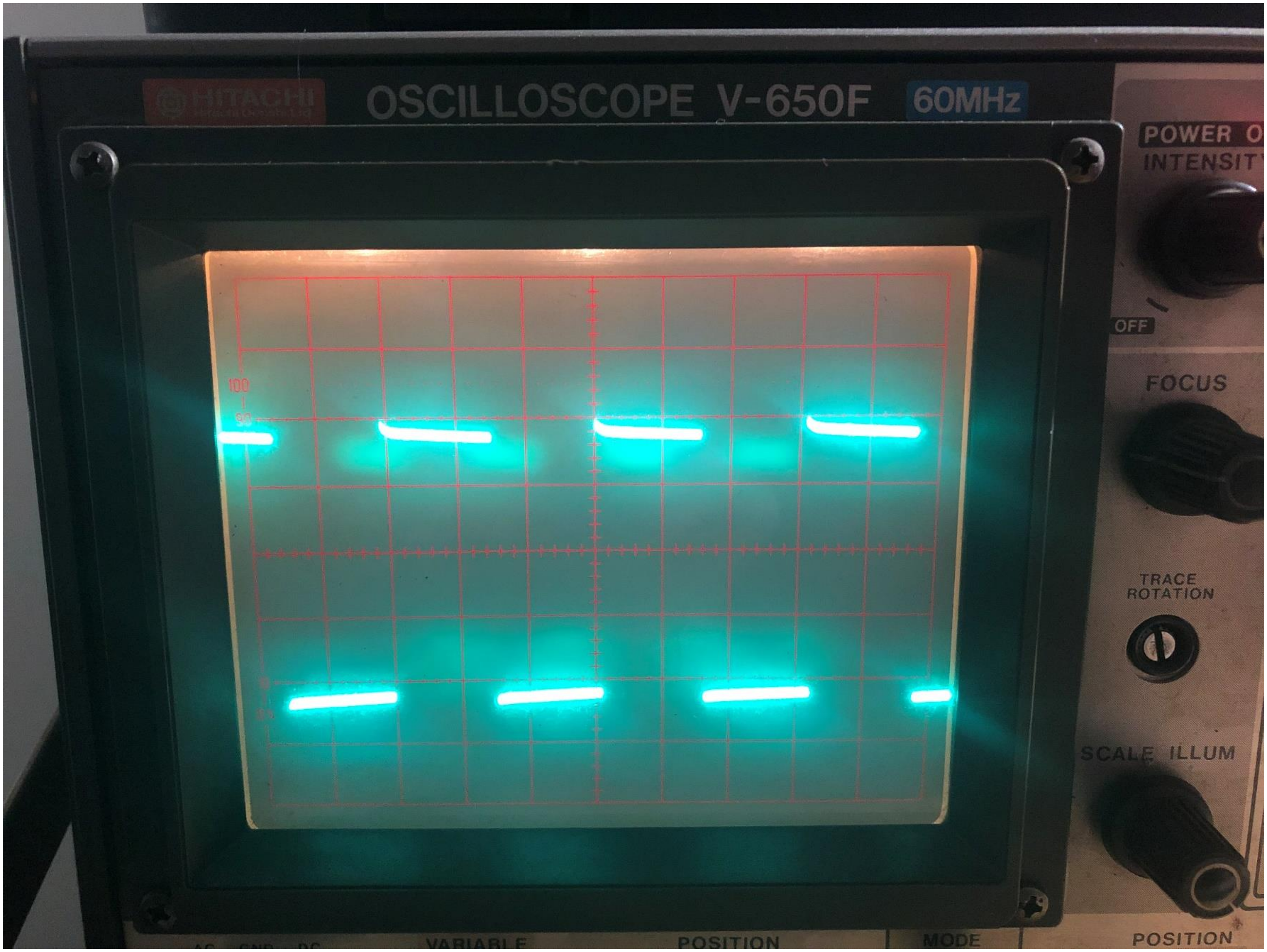
CLOCKOUT pin shown on scope above for Part A

<u>Scope settings</u>: 1 V / div. (&) .5 uS / div.

$$Observed\ CLOCKOUT\ (Part\ A)\ = \frac{1}{T} = \frac{1}{1\ uS} = 1\ MHz$$

b) The expected LED flashing frequency for Part A was calculated to be around 654.45 Hz. This value is based off the assumption that internal clock (Fosc) is running at 4 MHz. Complete calculations and details can be found in “Lab4_calculations_LeonardoFusser.pdf” that is attached to this submission.

c) My LEDs do not flash because the time that they are on or off was calculated to be around 764 uS (as shown in the above calculations). This is relatively a very small number, which means the LEDs will be turning on or off very fast. Because of this, my eyes are not able to see the flashing without looking at the details through an oscilloscope.



LED flashing frequency from RA0 shown on scope above for Part A

Scope settings: 1 V / div. (&) .5 mS / div.

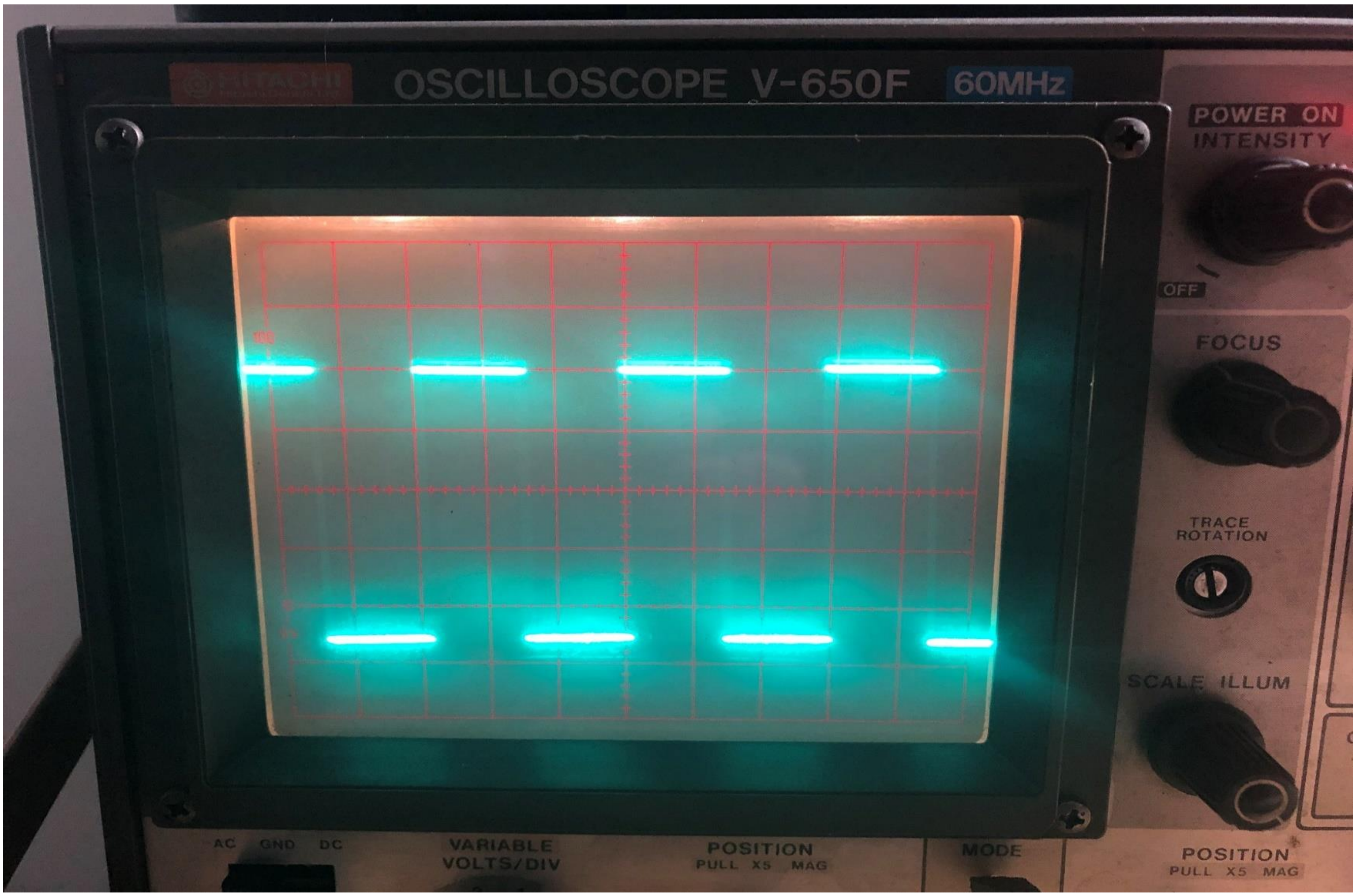
$$RA0\ flashing\ frequency\ (Part\ A)\ = \frac{1}{T} = \frac{1}{1.5\ mS} = 666.66\ Hz$$

PART B:

8a) To set the slowest clock possible, IRCF <2:0> (internal oscillator frequency select bits) of the OSCCON register need to be set to 000 to set the internal clock to 31 kHz.

b) The expected frequency to be observed at the CLOCKOUT pin is 7.7 kHz. It is 7.7 kHz because internal clock is running at 31 kHz and as it is configured (for INTOSC mode – set in config bits) the CLOCKOUT pin will output the internal oscillator frequency divided by 4. So,

$$CLOCKOUT = \frac{F_{OSC}}{4} = \frac{31\ kHz}{4} = 7.7\ kHz$$

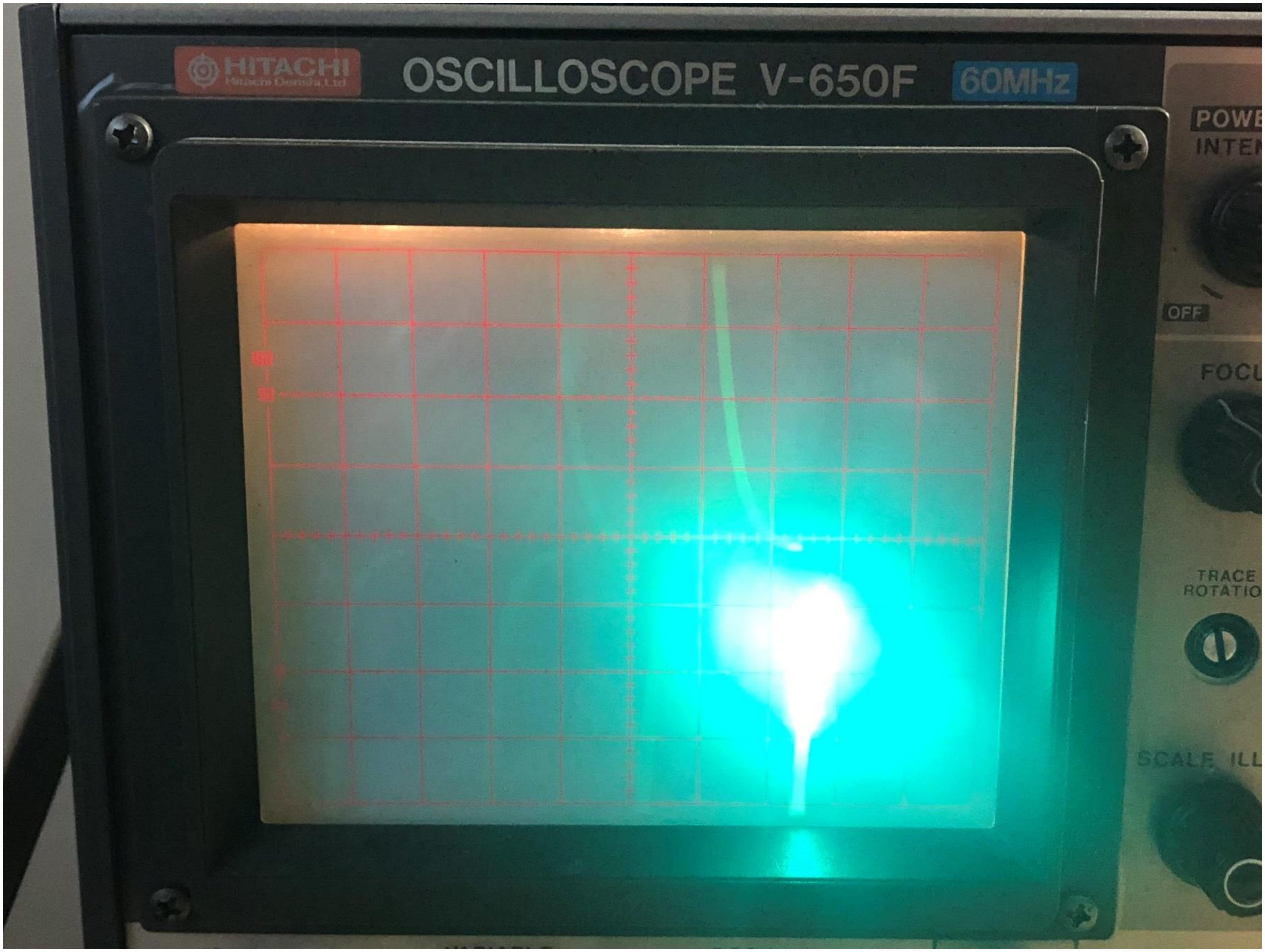


CLOCKOUT pin shown on scope above for Part B

Scope settings: 1 V / div. (&) 50 uS / div.

$$Observed\ CLOCKOUT\ (Part\ B) = \frac{1}{T} = \frac{1}{150\ uS} = 6.7\ kHz$$

b) The expected LED flashing frequency for Part B was calculated to be around 5.07 Hz. This value is based off the assumption that internal clock (Fosc) is running at 31 kHz. Complete calculations and details can be found in “Lab4_calculations_LeonardoFusser.pdf” that is attached to this submission.



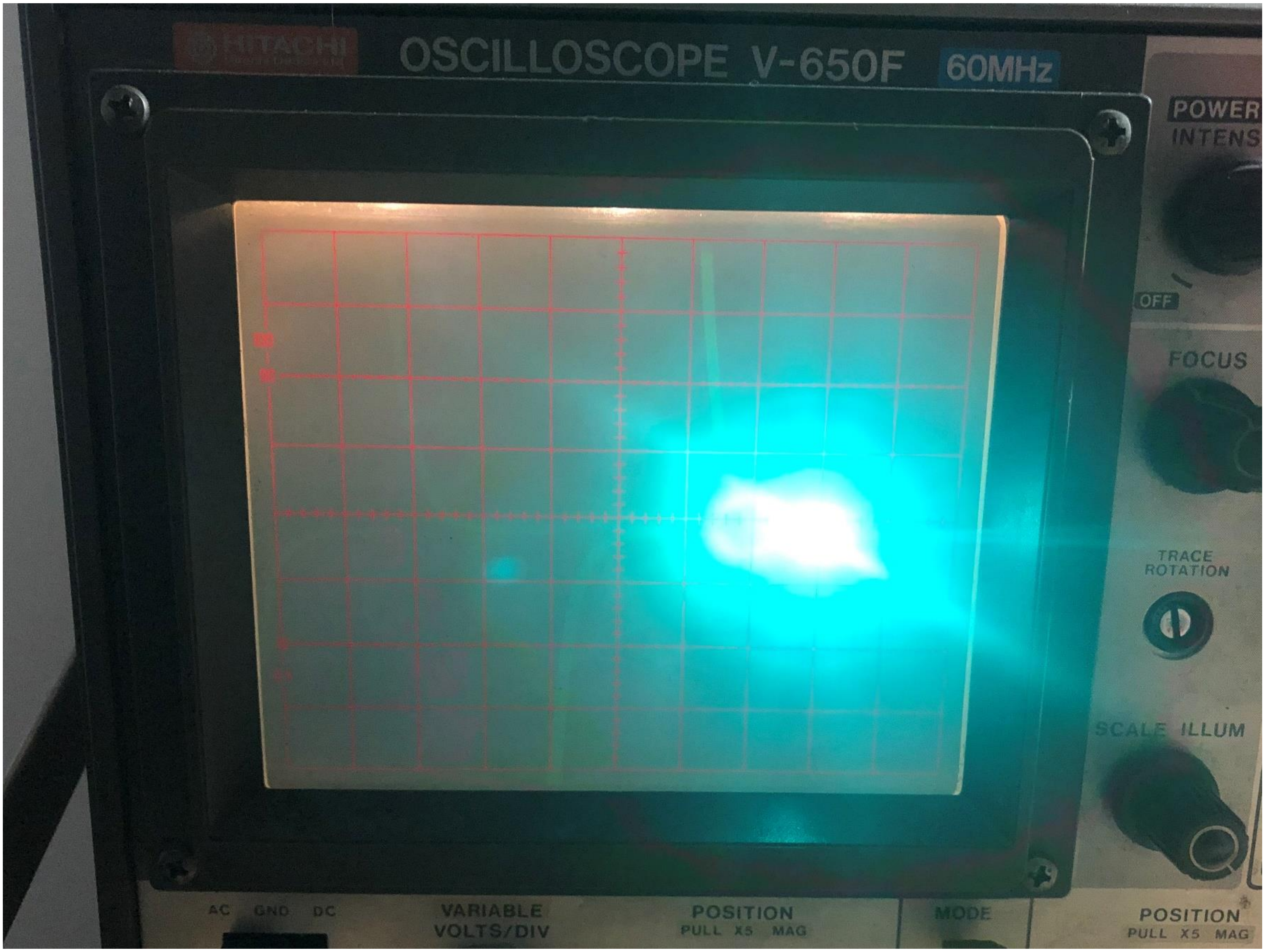
LED flashing frequency from RA0 shown on scope above for Part B Q7-9

Scope settings: 1 V / div. (&) 100 mS / div.

$$RA0\ flashing\ frequency\ (Part\ B\ Q7\ to\ 9) = \frac{1}{T} = \frac{1}{200\ mS} = 5\ Hz$$

9) The LEDs now flash at a slower rate and the flashing is noticeable unlike from before in Part A. This time, the LEDs are staying on for 98.58 mS and are off for 98.58 mS. Internal clock rate is set to 31 kHz (slowest one available). The reason the LEDs stay on and off for the same amount of time is because it is used to achieve 50% duty cycle requirement. Exact details and how I came up with the time the delays are generated for could be found in the “Lab4_calculations_LeonardoFusser.pdf” file.

10a) The LEDs now flash at an even slower rate than before and are still visible to the human eye. This time, the LEDs are staying on for 197.80 mS and are off for 197.80 mS. This is 2x slower than before. Once again, internal clock rate is set to 31 kHz (slowest one available). The LEDs stay on and off for the same amount of time is because it is used to achieve 50% duty cycle requirement. Exact details and how I came up with the time the delays are generated for could be found in the “Lab4_calculations_LeonardoFusser.pdf” file.



LED flashing frequency from RA0 shown on scope above for Part B Q10

Scope settings: 1 V / div. (&) 100 mS / div.

$$RA0\ flashing\ frequency\ (Part\ B\ Q10)\ = \frac{1}{T} = \frac{1}{400\ mS} = 2.5\ Hz$$

*Note: *complete details and calculations can be found in the "Lab4_calculations_LeonardoFusser.pdf" document.*

PART C:

13a) There is no memory space assigned to the “include” directive because it doesn’t need to store any data in program memory.

b) The hex code at the left column is the assembled machine code and the memory location in which it is placed (once the mnemonics appear).

c) Directive “ORG” is where the assembler loads instructions and data into memory.

d)

```
          Program Memory Usage
           Start         End
      ---------   ---------
       0x000000    0x000020
       0x002007    0x002008
35 out of 8455 program addresses used, program memory utilization is 0%
```

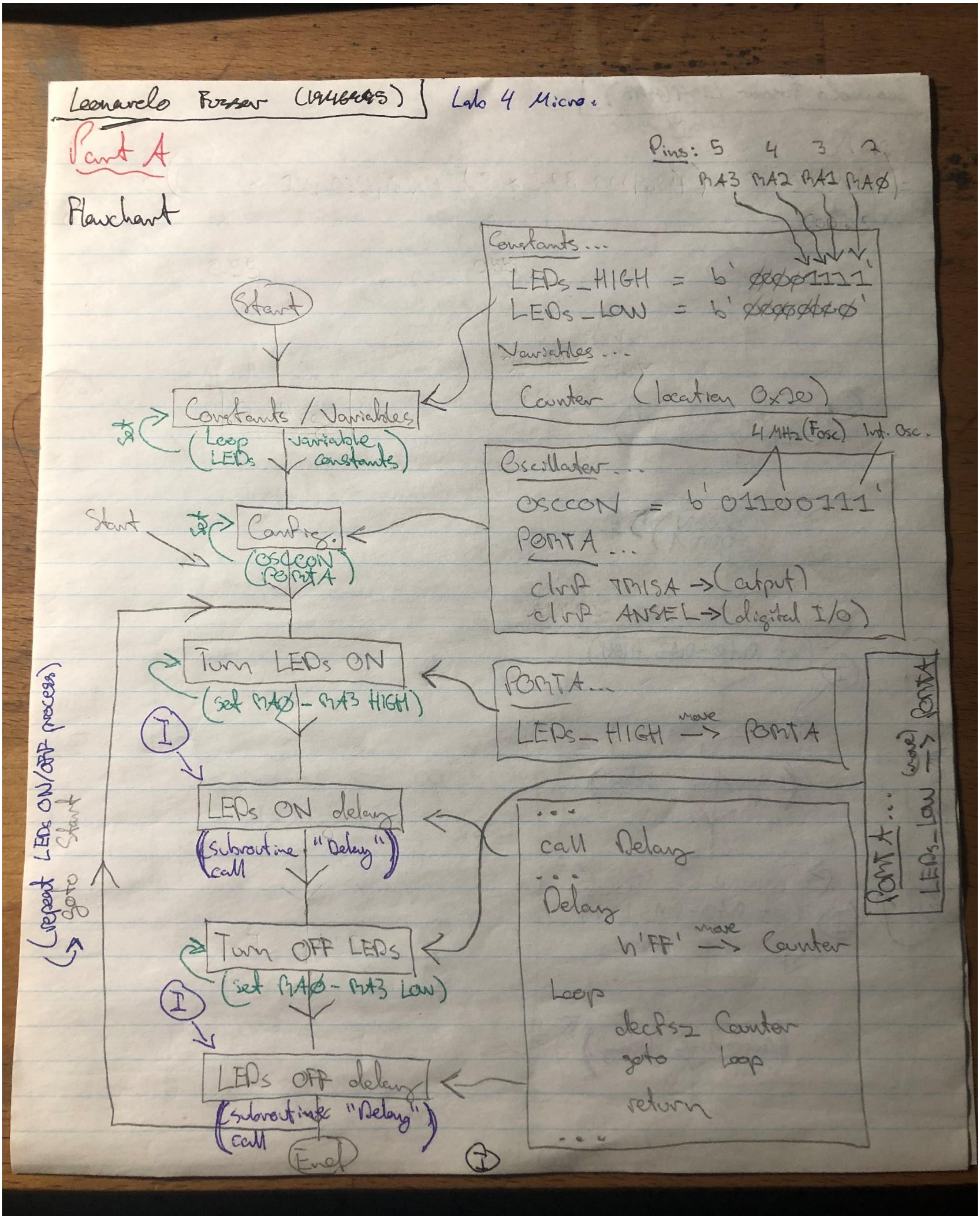
Program memory usage shown above for Part B/C

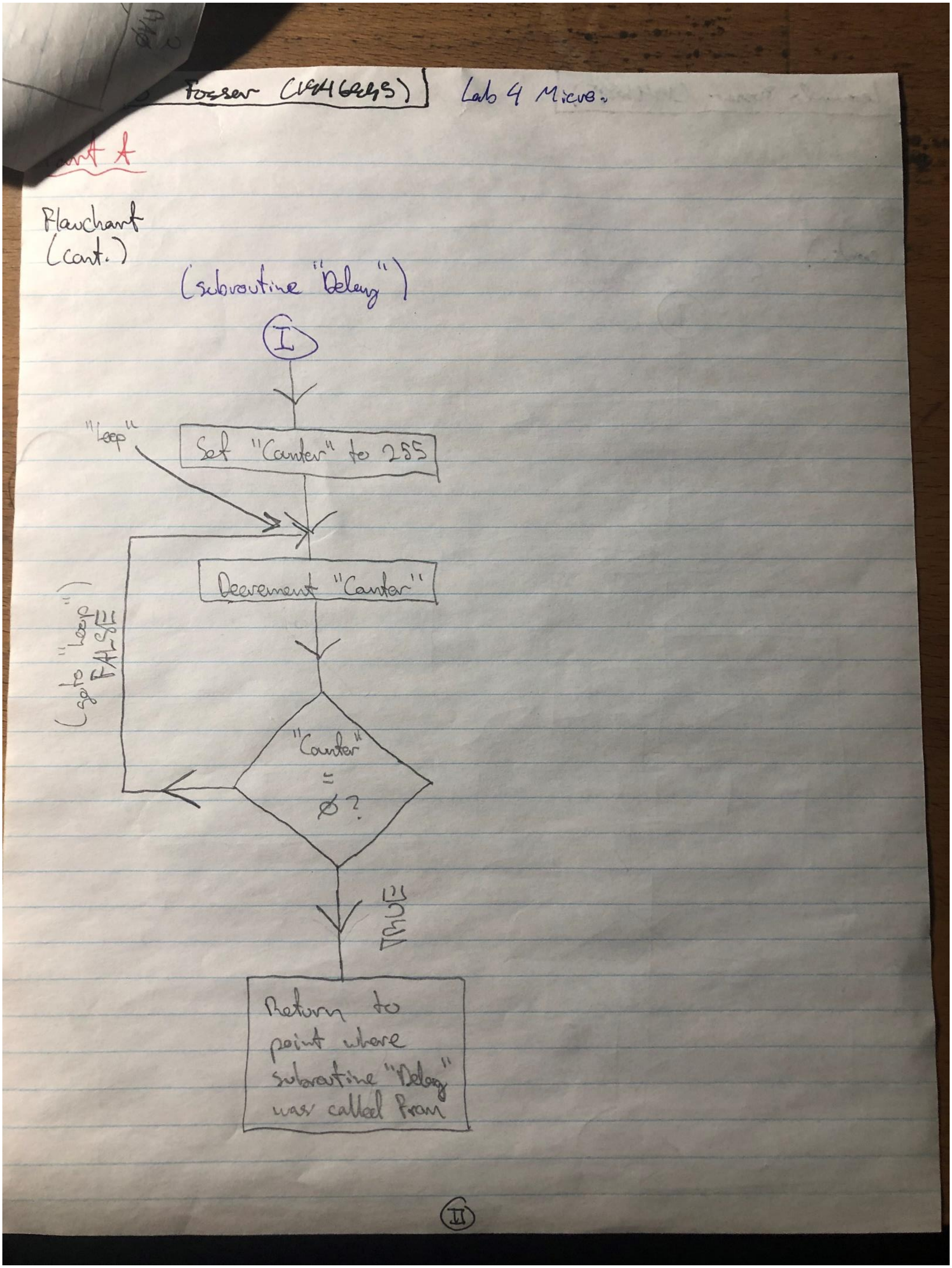
Total size of my source code in program memory is 35 bytes large.

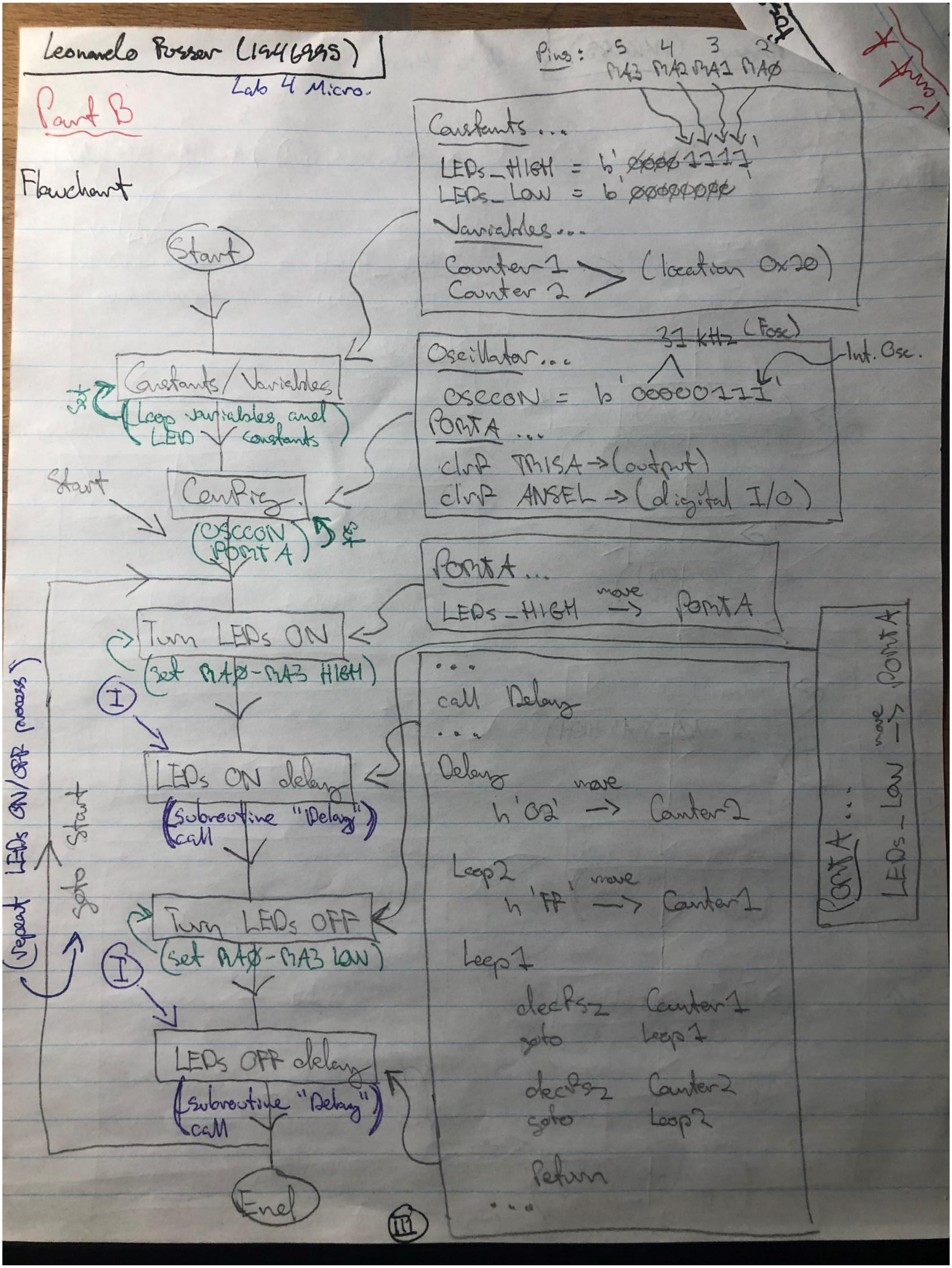
*Note: *the .lst and .map file for Part B/C is attached to this lab report submission.*

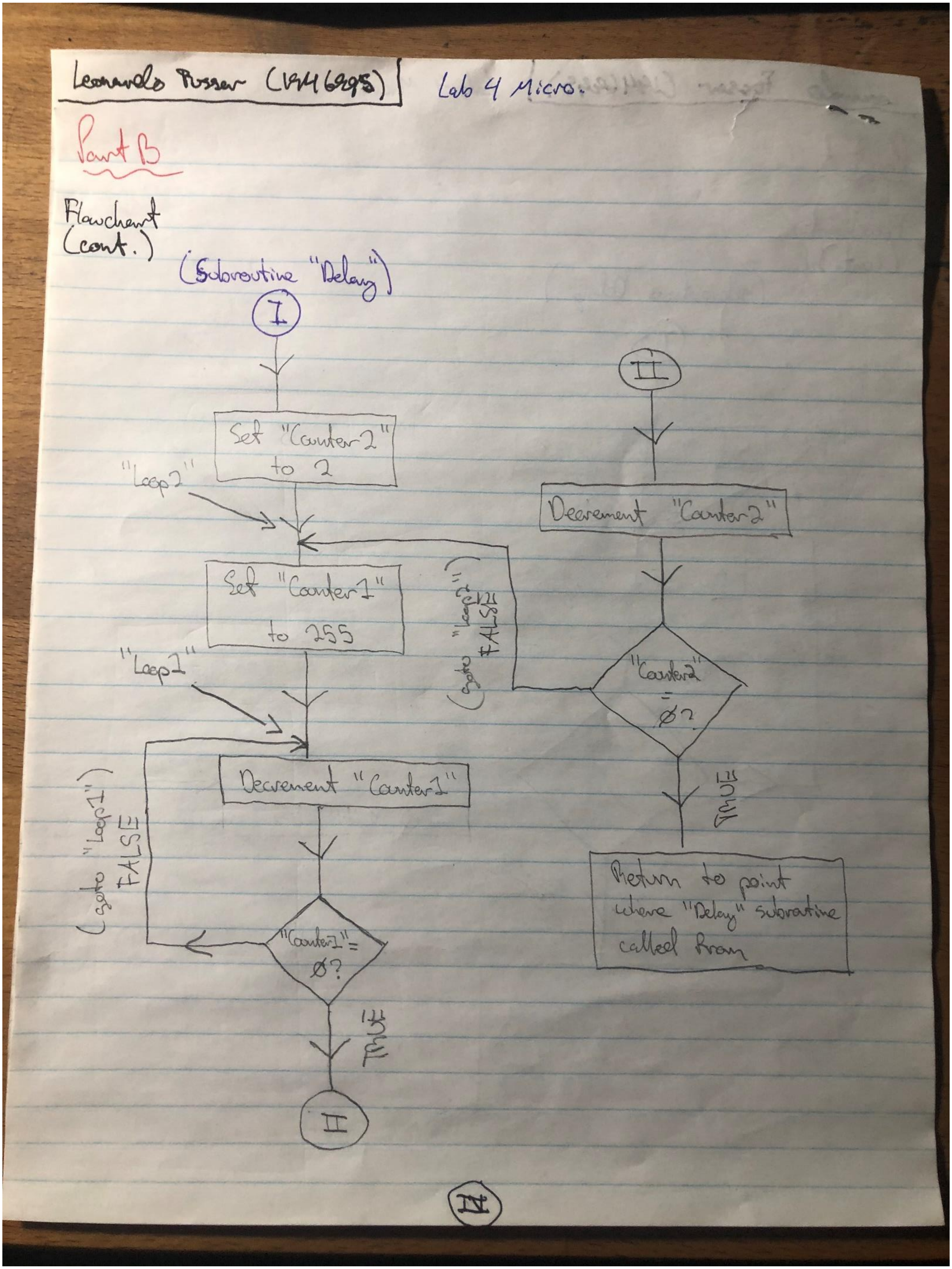
Leonardo Fusser (1946995) November 12th, 2020

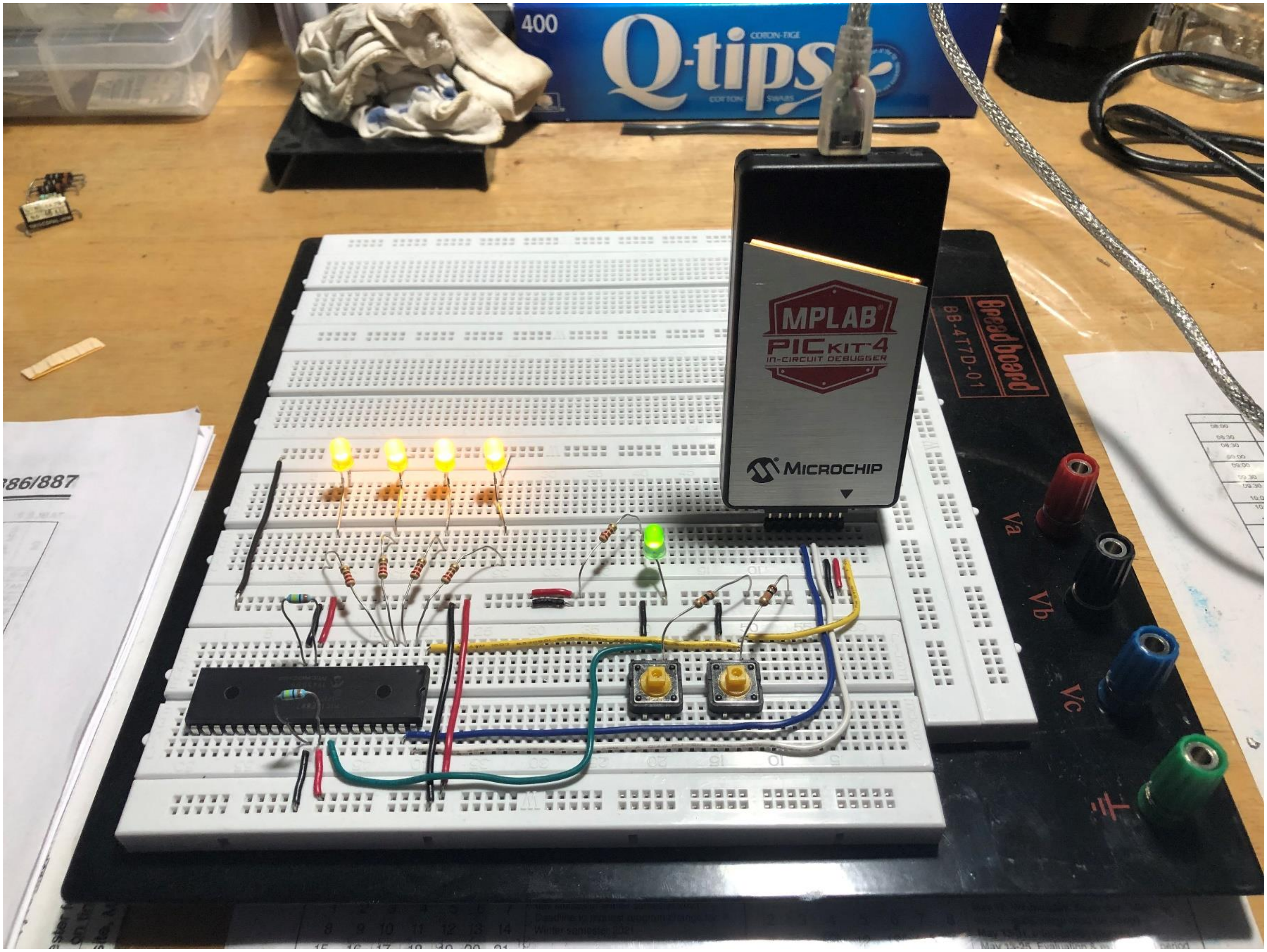
Original design:



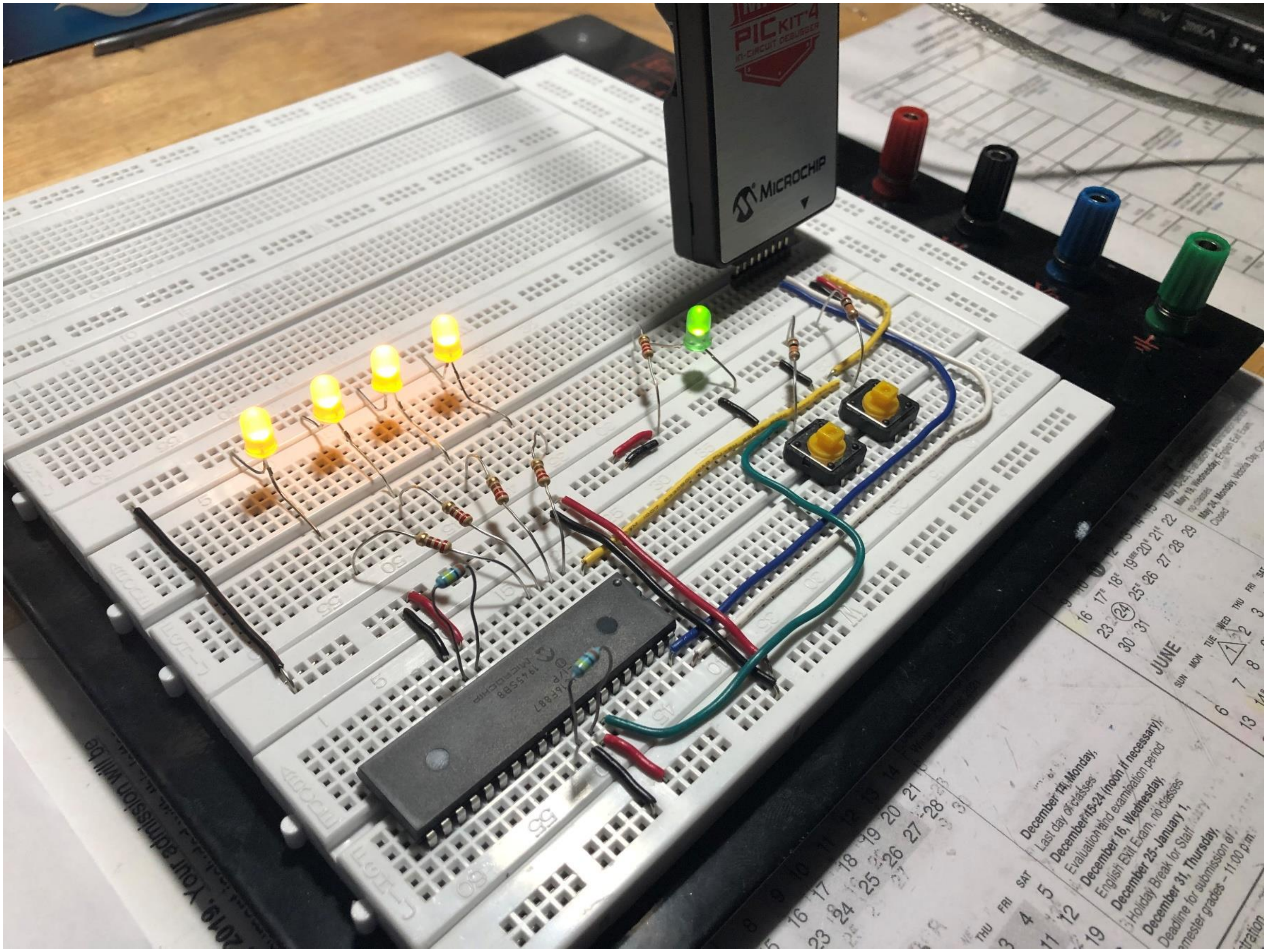

Fusser (1946995)
Lab 4 Micro.
Flowchart (cont.)
(subroutine "Delay")
I
Set "Counter" to 255
"Loop"
Decrement "Counter"
(goto "Loop")
"FALSE"
"Counter" = 0?
"TRUE"
Return to point where subroutine "Delay" was called from
II


Leonardo Fusser (1946995)
Lab 4 Micro.
Part B
Flowchart
Start
Constants / Variables
(Loop variables and LED constants)
set
Start
Config.
(OSCCON PORT A)
set
Turn LEDs ON
(set RA0-RA3 HIGH)
I
LEDs ON delay
(subroutine "Delay" call)
Turn LEDs OFF
(set RA0-RA3 LOW)
I
LEDs OFF delay
(subroutine "Delay" call)
End
goto Start
(repeat LEDs ON/OFF process)
III
Pins: 5 4 3 2
RA3 RA2 RA1 RA0
Constants...
LEDs_HIGH = b'00001111'
LEDs_LOW = b'00000000'
Variables...
Counter 1
Counter 2
(location 0x20)
31 kHz (Fosc)
Int. Osc.
Oscillator...
OSCCON = b'00000111'
PORTA...
clrf TRISA → (output)
clrf ANSEL → (digital I/O)
PORTA...
LEDs_HIGH move → PORTA
PORTA...
LEDs_LOW move → PORTA
...
call Delay
...
Delay
h'02' move → Counter2
Loop2
h'FF' move → Counter1
Loop1
decfsz Counter1
goto Loop1
decfsz Counter2
goto Loop2
Return
...

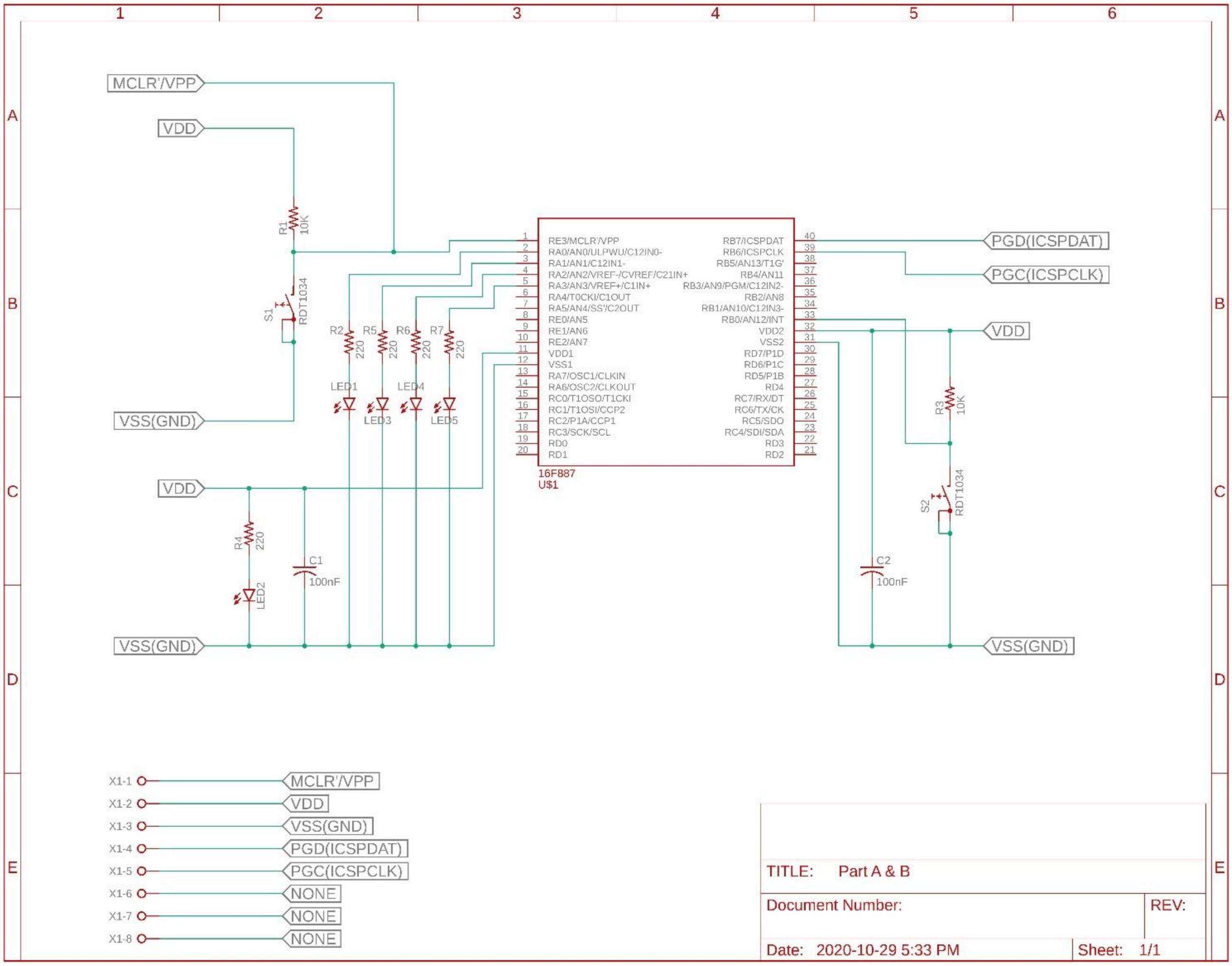

Leonardo Fusser (1946995)
Lab 4 Micro.
Part B
Flowchart (cont.)
(Subroutine "Delay")
I
Set "Counter2" to 2
"Loop2"
Set "Counter1" to 255
"Loop1"
Decrement "Counter1"
"Counter1" = Ø?
FALSE (goto "Loop1")
TRUE
II
II
Decrement "Counter2"
"Counter2" = Ø?
FALSE (goto "Loop2")
TRUE
Return to point where "Delay" subroutine called from
IV



Final prototype circuit for both Part A and Part B. Blinking LEDs are connected to pins 2 – 5 (RA0 – RA3) on PIC 16F887. Each LED has appropriate current limiting resistor (200 Ω). Green LED is power-up LED. Push-button right under power-up LED not used for this lab. Push-button next to it (to the right) is reset button for PIC 16F887.



Final prototype circuit for both Part A and Part B. Blinking LEDs are connected to pins 2 – 5 (RA0 – RA3) on PIC 16F887. Each LED has appropriate current limiting resistor (200 Ω). Green LED is power-up LED. Push-button right under power-up LED not used for this lab. Push-button next to it (to the right) is reset button for PIC 16F887.



*<u>Note</u>: *a copy of this schematic is included with this lab submission. It is titled "Lab4_LeonardoFusser_Schematic – Part A-B.pdf"*

Leonardo Fusser (1946995) November 12th, 2020

Part A code (I)

```
;*************************************************************************************************************
;*Leonardo Fusser (1946995)                                                                                 *
;*Microcontroller & Microprocessor systems, Lab4 - Part A & Part B Q7-9, Day Yann Fong                     *
;*Program to flash 4 LEDs from pins 2-5 at 50% duty cycle on PIC16F887.                                     *
;*************************************************************************************************************

;[PIC config]

#include "p16f887.inc"

; CONFIG1
; __config 0x23F5
 __CONFIG _CONFIG1, _FOSC_INTRC_CLKOUT & _WDTE_OFF & _PWRTE_OFF & _MCLRE_ON & _CP_OFF & _CPD_OFF & _BOREN_ON & _IESO_OFF & _FCMEN_OFF & _LVP_OFF
; CONFIG2
; __config 0x3FFF
 __CONFIG _CONFIG2, _BOR4V_BOR40V & _WRT_OFF

;[Start of program]

;*****************************************************
;[Constant declarations]

LEDs_HIGH    equ b'00001111'   ;Value when LEDs are ON.
LEDs_LOW     equ b'00000000'   ;Value when LEDs are OFF.
;*****************************************************

    ;*********************************************************************************************
    ;[Variable declarations]

    cblock   0x20            ;Stores variables at address 0x20 in GPR.
    Counter                  ;Variable for software delay (in subroutine "Delay") - loop counter.
    endc
    ;*********************************************************************************************

    org     0                ;Reset vector.
    goto    Main             ;Goto "Main".

Main

    ;*************************************************************************************************************
    ;[Config for internal oscillator]
    ;(Used for debugging purposes for Part A and used to select lowest possible internal frequency for Part B, Q7-9)

    ;banksel OSCCON          ;(Oscillator control register)
    ;movlw   b'00000111'     ;Sets internal clock rate to 31 kHz (IRCF<2:0> = 000) Fosc = 31 kHz.
                             ;(SCS is set for internal oscillator for system clock, LTS is stable, HTS is stable, OSTS
                             ; is set for internal oscillator HFINTOSC or LFINTOSC)
    ;movwf   OSCCON
    ;*************************************************************************************************************

    ;*************************************************************************************************************
    ;[Config for PORTA]

    ;TRISA0-TRISA7
    banksel TRISA            ;(Register that determines if pin(s) are configured as Input or Output)
    clrf    TRISA            ;Sets RA0, RA1, RA2, RA3 to be an output.

    ;ANS0-ANS7
    banksel ANSEL            ;(Register that determines if pin(s) are configured as Anolog or Digital I/O)
    clrf    ANSEL            ;Sets RA0, RA1, RA2, RA3 to be a digital I/O.
    ;*************************************************************************************************************
```

Leonardo Fusser (1946995) November 12th, 2020

Part A code (II)

```
;***************************************************************************************************************************
;(Begin LED ON/OFF procedure)
Start

    ;RA0 - RA7
    banksel PORTA           ;(Register that determines if pin(s) are configured to produce an active LOW or active HIGH)
    movlw   LEDs_HIGH       ;Sets RA0, RA1, RA2 and RA3 to turn ON.
    movwf   PORTA           ;" "

    call    Delay           ;Goto subroutine (software delay) - to keep LEDs ON for a certain amount of time - 50% ON**!

    banksel PORTA           ;(Register that determines if pin(s) are configured to produce an active LOW or active HIGH)
    movlw   LEDs_LOW        ;Sets RA0, RA1, RA2 and RA3 to turn OFF.
    movwf   PORTA           ;" "

    call    Delay           ;Goto subroutine (software delay) - to keep LEDs OFF for a certain amount of time - 50% OFF**!

    goto    Start           ;Repeat LED ON/OFF procedure (go back to "Start" and start procedure over).

;(Subroutine - software delay)
Delay

    movlw   h'FF'           ;255 in hex.
    movwf   Counter         ;Load 255 into loop counter (used for loop below).

;(Loop that creates delay for LEDs)
Loop

    decfsz  Counter         ;Decrement "Counter" until value is 0.
    goto    Loop            ;Goto "Loop" in order to keep decrementing "Counter".
                            ;(Note: this instruction is SKIPPED if "Counter" is 0, creating a NOP instruction)

    return                  ;Return to point where subroutine was originally called from.
;***************************************************************************************************************************

    End                     ;End

    ;[End of program]

;**Subroutine "Delay" creates a software delay for LEDs to remain ON or OFF. The time it takes to keep the LEDs on or
; OFF is dependent on what Fosc is set at. For example, if Fosc is set for 4 MHz, the delay generated will keep the LEDs ON or OFF
; for 764 uS. Another example, if Fosc is set for 31 kHz, the delay generated will keep the LEDs ON or OFF for 98.58 mS.
; If Fosc is set to another value other than 4 MHz or 31 kHz, calculations will need to be done in order to determine how
; long the subroutine "Delay" will generate a delay for the LEDs to remain ON or OFF.
```

Leonardo Fusser (1946995) November 12th, 2020

Part B code (I)

```
;************************************************************************************************
;*Leonardo Fusser (1946995)                                                                     *
;*Microcontroller & Microprocessor systems, Lab4 - Part B Q10, Day Yann Fong                    *
;*Program to flash 4 LEDs from pins 2-5 at 50% duty cycle with longer delay on PIC16F887.       *
;************************************************************************************************

;[PIC config]

#include "p16f887.inc"

; CONFIG1
; __config 0x23F5
 __CONFIG _CONFIG1, _FOSC_INTRC_CLKOUT & _WDTE_OFF & _PWRTE_OFF & _MCLRE_ON & _CP_OFF & _CPD_OFF & _BOREN_ON & _IESO_OFF & _FCMEN_OFF & _LVP_OFF
; CONFIG2
; __config 0x3FFF
 __CONFIG _CONFIG2, _BOR4V_BOR40V & _WRT_OFF

;[Start of program]

;****************************************************
;[Constant declarations]

LEDs_HIGH    equ b'00001111'   ;Value when LEDs are ON.
LEDs_LOW     equ b'00000000'   ;Value when LEDs are OFF.
;****************************************************

    ;************************************************************************************************
    ;[Variable declarations]

    cblock   0x20             ;Stores variables at address 0x20 in GPR.
    Counter1                  ;Variable for software delay (in subroutine "Delay") - inner loop counter.
    Counter2                  ;Variable for software delay (in subroutine "Delay") - outer loop counter (nested loop).
    endc
    ;************************************************************************************************

    org     0                 ;Reset vector.
    goto    Main              ;Goto "Main".


Main

    ;************************************************************************************************
    ;[Config for internal oscillator]

    banksel OSCCON            ;(Oscillator control register)
    movlw   b'00000111'       ;Sets internal clock rate to 31 kHz (IRCF<2:0> = 000) Fosc = 31 kHz.
                              ;(SCS is set for internal oscillator for system clock, LTS is stable, HTS is stable, OSTS
                              ; is set for internal oscillator HFINTOSC or LFINTOSC)
    movwf   OSCCON
    ;************************************************************************************************

    ;************************************************************************************************
    ;[Config for PORTA]

    ;TRISA0-TRISA7
    banksel TRISA             ;(Register that determines if pin(s) are configured as Input or Output)
    clrf    TRISA             ;Sets RA0, RA1, RA2, RA3 to be an output.

    ;ANS0-ANS7
    banksel ANSEL             ;(Register that determines if pin(s) are configured as Anolog or Digital I/O)
    clrf    ANSEL             ;Sets RA0, RA1, RA2, RA3 to be a digital I/O.
    ;************************************************************************************************
```

Leonardo Fusser (1946995) November 12th, 2020

Part B code (II)

```
;****************************************************************************************************************************
;(Begin LED ON/OFF procedure)
Start

    ;RA0 - RA7
    banksel PORTA           ;(Register that determines if pin(s) are configured to produce an active LOW or active HIGH)
    movlw   LEDs_HIGH       ;Sets RA0, RA1, RA2 and RA3 to turn ON.
    movwf   PORTA           ;" "

    call    Delay           ;Goto subroutine (software delay) - to keep LEDs ON for a certain amount of time - 50% ON**!

    banksel PORTA           ;(Register that determines if pin(s) are configured to produce an active LOW or active HIGH)
    movlw   LEDs_LOW        ;Sets RA0, RA1, RA2 and RA3 to turn OFF.
    movwf   PORTA           ;" "

    call    Delay           ;Goto subroutine (software delay) - to keep LEDs OFF for a certain amount of time - 50% OFF**!

    goto    Start           ;Repeat LED ON/OFF procedure (go back to "Start" and start procedure over).

;(Subroutine - software delay)
Delay

    movlw   h'02'           ;2 in hex.
    movwf   Counter2        ;Load 2 into outer loop counter (used for outer loop below).

;(Loop that creates 2nd extended delay for LEDs - nested loop)
Loop2

    movlw   h'FF'           ;255 in hex.
    movwf   Counter1        ;Load 255 into inner loop counter (used for inner loop below).

;(Loop that creates 1st delay for LEDs)
Loop1

    decfsz  Counter1        ;Decrement "Counter1" until value is 0.
    goto    Loop1           ;Goto "Loop1" in order to keep decrementing "Counter1".
                            ;(Note: this instruction is SKIPPED if "Counter1" is 0, creating a NOP instruction)

    decfsz  Counter2        ;Decrement "Counter2" until value is 0.
    goto    Loop2           ;Goto "Loop2" in order to keep decrementing "Counter2".
                            ;(Note: this instruction is SKIPPED if "Counter2" is 0, creating a NOP instruction)

    return                  ;Return to point where subroutine was originally called from.
;*************************************************************************************************************

    End                     ;End

    ;[End of program]

;**Subroutine "Delay" creates a software delay for LEDs to remain ON or OFF. The time it takes to keep the LEDs on or
; OFF is dependent on what Fosc is set at. For example, if Fosc is set for 31 kHz, the delay generated will keep the LEDs ON or OFF
; for 197.80 mS. If Fosc is set to another value other than 31 kHz, calculations will need to be done in order to determine how
; long the subroutine "Delay" will generate a delay for the LEDs to remain ON or OFF. This software delay is based of off the
; nested loop technique.
```

*<u>Note</u>: *a copy of both .asm source files for both parts can be found with this lab report submission.*