**Internal clock & Software delay loop**

*Lab #3*

**Phillip May,** 1639294

**ABSTRACT**

*In this lab I programmed to the PIC16F887 to take longer to complete a given amount of code..*

Experiment Performed on **5 October 2017**

Report Submitted on **12 October 2017**



Department of Computer

Engineering Technology

**INTRODUCTION TO MICROPROCESSORS (247-302)**

*Day Yann Fong*

Table of Contents

[Introduction 3](#_Toc495560299)

[Procedure 3](#_Toc495560300)

[Results and Discussion 3](#_Toc495560301)

[Conclusion 5](#_Toc495560302)

[References 5](#_Toc495560303)

# Introduction

---In this lab I will about how to program the PIC16F887 and how to make the circuit required to use it. We will set up and test the internal clock, basic digital input, basic digital output and branching programs.

# Procedure

The procedure is provided in the lab instructions.

# Results and Discussion

Part A: Flashing LEDs

1. Revise circuit to include 4 LEDS on RA0-3

3. Setup your code with the following configuration settings:

• To use the default internal clock, with clock out.

• Disable watchdog timer.

In order to set the clock to internal with clock out, you set the oscillator control selection bits to INTRC\_CLKOUT. To disable the watchdog timer you once again go to configuration bits and choose to disable it. After setting all your configuration bits generate code and put it at the top of your code in the configuration section.

4. Write your code to flash all the 4 LEDs at the same time, at 50% duty cycle. Use a simple delay loop (loop for 255 times) to generate the delay between turning on and off the LEDs.

The code is included with the lab.

1. Is that possible to generate a simple delay loop (without nested) of more than 255 times?

Not possible on a PIC16F887 due to the limitation of the data bus being 8 bit. Because the data bus is 8 bits it can only handle a number up 2^8-1 in a branch instruction. It would require a nested loop more than 255 times. An idea that did occur to me and that I never tested was to have the loop exit on a certain bit being 0 and increment by a number in such a way that the register will overflow several times before that bit ever becomes. This is probably impractical and will result in code that is difficult to read.

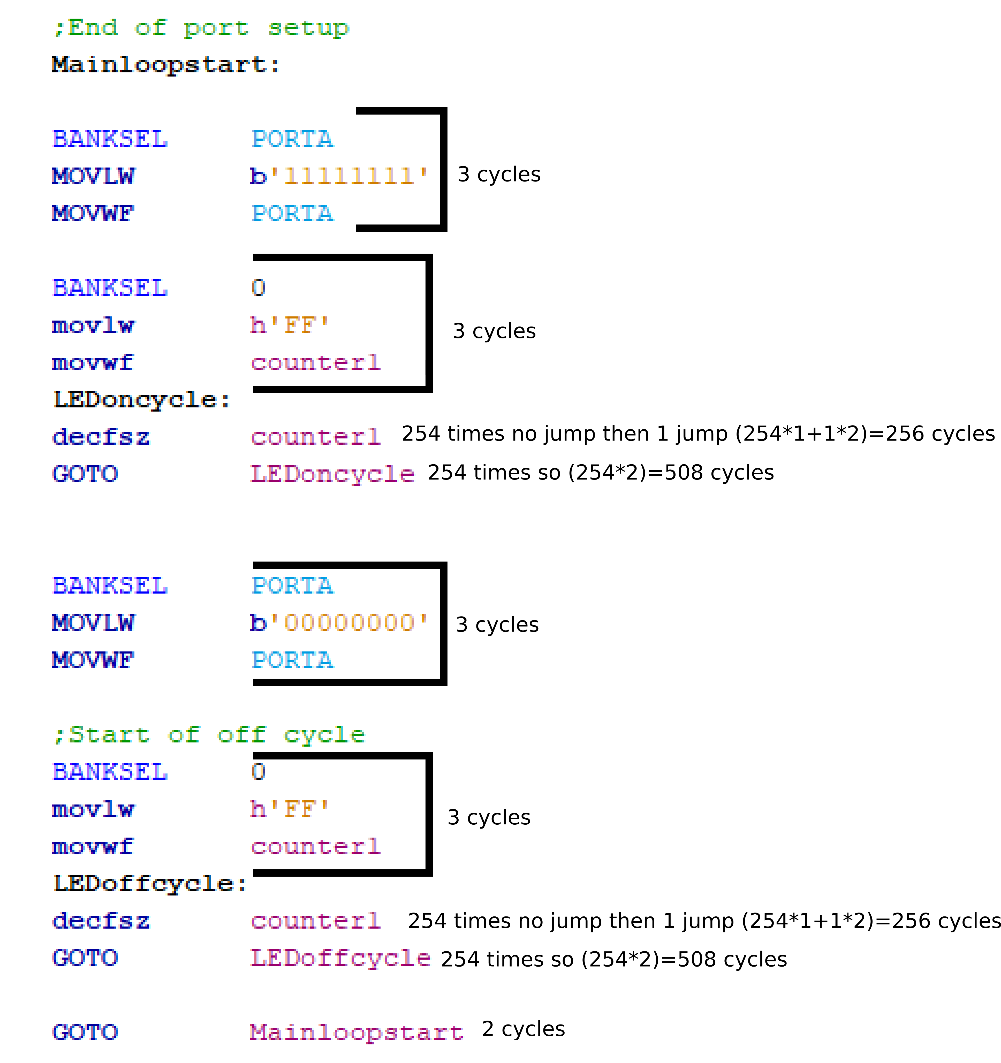
5. Compile and download your code to your circuit.

a) What is the frequency of the waveform observed at clock out pin? Is this what you are expecting?

Given that the datasheet said clkout is (Frequency oscillator)/4 output, this is exactly what I expected. The default internal oscillator speed is 4Mhz so a clkout of 1Mhz of a period of 1 micro second if expected.

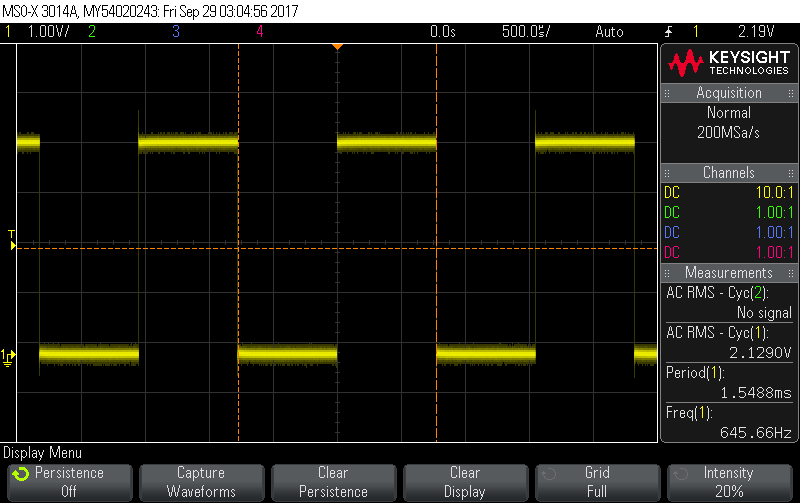
1. What is the LED flashing frequency generated by your code? Does this match the calculation of delay loop based on your code?

Calculation



4Mhz clock so 1 micro second instruction cycle.

Total instruction cycles is 3+3+256+508+3+3+256+508+2=1542uS or 1.542mS. This is within 99.5% accuracy of what the oscilloscope measured.



c) Do your LEDs flash? If no, why?

The oscillator clearly shows the voltage is being set and unset so the LED must be turning on and off the problem is that I probably can’t see it as it is flashing too fast.

Part B: Flashing LEDs at slower speeds

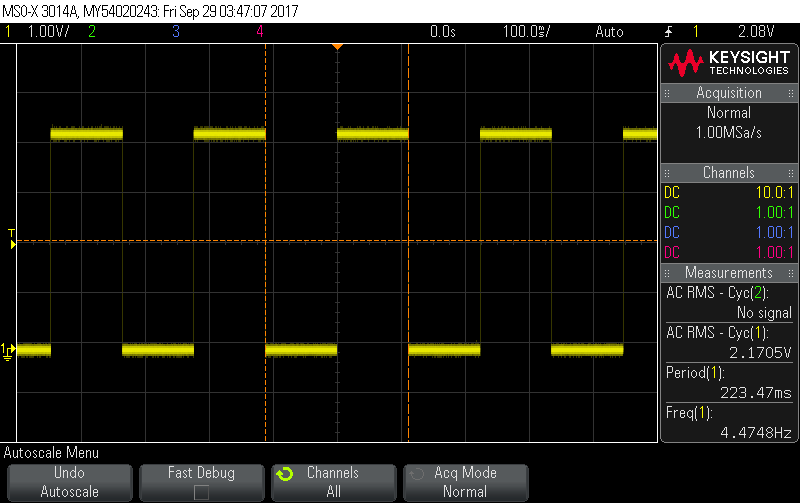
8. Based on your code in part A, reduce the flashing speed of your LED by modify your code to select the slowest internal clock available.

a) What are the settings needed to select the slowest internal clock?

Bits 5,6 and 7 of OSCCON need to be set to zero, I looked at the registers default values and modified only bits I needed while keeping the others as default but an andwf would be more optimal if the values are non-default.

b) What would be the expected clock out frequency and LED flashing frequency? Show your calculation.

The new instruction cycle would be 31Khz/4=7.75Khz. The instruction period would 1/7.75Khz = 129.032uS. From my previous calculations there where 1542 instructions so 129.032uS\*1542=198.9677mS. Which is reasonably close to what I measured



10. To further increase the delay I put 255 loop inside a nested loop that repeats twice so as to approximately double the delay, the code is included.

# Conclusion

In this lab I learn how to use software delays on the PIC16F887 in order to stop code from being executed instantly. I also learned about using nested loops on the PIC16F887. While the lab went according to plan there are a few possible optimizations such as using ANDWF instead of setting entire registers.

# References

|  |  |
| --- | --- |
| [1] | Microchip Technology Inc., "MPASM Assembler, MPLINK Object Linker, MPLIB Object Librarian User's Guide," Microchip Technology Inc., 2013. [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/33014L.pdf. [Accessed 20 9 2017]. |