

THE CITY COLLEGE OF NEW YORK
Department of Electrical Engineering
EE425 Computer Engineering Laboratory

Experiment 2 – Pulse Train Generation

Objective: The experiment is designed to exhibit some of the capabilities of generating pulses at the ports of the PIC18F4520.

Specific Tasks:

Use the *P0_template.asm* file as a starting point in order to complete the following tasks.

1. Write an assembly program that generates a pulse train at the **RA2** bit of **PORTA** for each of the following cases.
 - a. $T_H = T_L = 0.1\text{ms}$
 - b. $T_H = 0.1\text{ms}, T_L = 0.3\text{ms}$
2. Write an assembly program that generates a pulse train at the **RA2** bit of **PORTA** for each of the following cases.
 - a. $T_H = T_L = 0.2\text{ms}$
 - b. $T_H = 0.3\text{ms}, T_L = 0.2\text{ms}$
 - c. $T_H = 0.25\text{ms}, T_L = 0.35\text{ms}$

By the end of this assignment you should have written five (5) different *.asm* files.

Notes:

Please observe the following items while working through this assignment.

1. Do not use the **Prescaler** option.
2. Do not modify any line of code of the **LoopTime** subroutine in the *P0_template.asm* file.
3. Do not call the **LoopTime** subroutine more than once.
4. Do not create a second subroutine similar to **LoopTime**.

Guidelines:

There are four distinct phases for the pulse train generation:

1. Configure the pins of the PORT as outputs.
2. Send your pulse to the configured pins using appropriate commands (review the following commands: *btg*, *bsf*, *bcf*).
3. Use conditional logic and a “delay” mechanism to control the duty cycle of the pulse trains.
4. Loop around indefinitely to generate a continuous pulse train.

To read or write anything from a port, it is necessary to configure the port pins accordingly, using the *Special Function Register* (SFR) *TRISx*, where *x* is the name of the port (e.g. *TRISB* configures the pins of port B).

In order to configure a pin as an output, put a 0 in the corresponding bit of *TRISx*. Similarly, to configure a pin as an input, put a 1 in the corresponding bit.

Example: to configure pin 4 of port B as an output and the others as inputs:

```
MOVLFB'11101111', TRISB
```

where *MOVLFB* is a macro defined as follows:

```
MOVLFB macro    literal,    dest
               movlw    literal
               movwf    dest
               endm
```