Ga-wun Kim

ENGR 270

Scott Koss

Lab Partners: Rebecca, Kantas

# Computer and Organization

# &

# Microprocessors

# Lab #4

## Introduction

In this lab, we used the knowledge we learned in class to get hands on experience with interrupt. In each experiment, we were asked to make interrupt that acted upon when switches were moved from EDbot.

## Experiment 1

We used the example code; that provided us to implement a system that start with 0.5 sec LED off and on. We observed that each time INT0 occurs, the on and off time increases by one second and each time INT1 occurs, the on and off time decreases by one second. The delay should never be less than 0.5 second or more than 25 seconds.

The below codes are what we did.

**Configured Code:**

```
list p=18F1220 ; Processor type
radix hex ; Default radix for data
config WDT=OFF, LVP=OFF, OSC = INTIO2 ; Disable Watchdog timer, Low V. Prog,
; and RA6 as a clock
#define countOD 0x80
#define countID 0x81
#define lowerlimit 0x82
#define upperlimit 0x83
#include p18f1220.inc ; This header file includes address and
; bit definitions for all SFRs
org 0x000 ; Executes after Reset
GOTO StartL
org 0x008 ; Executes after high priority interrupt
GOTO HPRIO
org 0x018 ; Executes after low priority interrupt
GOTO LPRIO
org 0x20
HPRIO: ; High priority interrupt
CheckUpper1:
CPFSLT upperlimit ; Check if within range
BRA CheckLower1
ADDLW 0xF6 ; If 25 < WREG, Subtract 10
BCF INTCON, INT0IF ; Clear Interrupt 0
RETFIE ; Return from interrupt
CheckLower1:
CPFSGT lowerlimit ; Check if within range
BRA INCR
ADDLW 0xA ; If 5 > WREG, Add 10
BCF INTCON, INT0IF ; Clear Interrupt 0
RETFIE ; Return from interrupt
INCR:
ADDLW .10 ; When interrupt 0 occurs
```

```
BCF INTCON, INT0IF ; Clear Interrupt 0
RETFIE ; Return from interrupt
LPRIO: ; Low priority interrupt
BTFSC INTCON3, INT1IF ; Check for Interrupt 1
BRA Intr1
RETFIE ; Return from interrupt
Intr1: ; Take care of Interrupt 1
CheckUpper2:
CPFSLT upperlimit ; Check if within range
BRA CheckLower2
ADDLW 0xF6 ; If 25 < WREG, Subtract 10
BCF INTCON3, INT1IF ; Clear interrupt 1 flag
RETFIE ; Return from interrupt
CheckLower2:
CPFSGT lowerlimit ; Check if within range
BRA DECR
ADDLW 0xA ; If 5 > WREG, Add 10
BCF INTCON3, INT1IF ; Clear interrupt 1 flag
RETFIE ; Return from interrupt
DECR:
ADDLW 0xF6 ; W (W-10). {note: SUBLW .10 will not
; work}
BCF INTCON3, INT1IF ; Clear interrupt 1 flag
RETFIE ; Return from interrupt
StartL: ; Initialization code to be executed
; during Reset
; Initialize all I/O ports
CLRF PORTA ; Initialize PORTA
CLRF PORTB ; Initialize PORTB
MOVLW 0x7F ; Set all A\D Converter Pins as
MOVWF ADCON1 ; digital I/O pins
MOVLW 0x0D ; Value used to initialize data direction
MOVWF TRISA ; Set Port A direction
MOVLW 0xC7 ; Value used to initialize data direction
MOVWF TRISB ; Set Port B direction
MOVLW 0x00 ; Clear WREG
; Enable NT0 and INT1
BSF INTCON, GIE ; Enable interrupts
BSF INTCON, PEIE ; Enable all interrupts
BSF INTCON, INT0IE ; Enable interrupt 0
BSF INTCON3, INT1IE ; Enable interrupts 1
BCF INTCON3, INT1IP ; INT1 is set to low priority
BSF RCON, IPEN ; Enable priority levels on interrupts
BCF INTCON, INT0IF ; Flags must be cleared to allow an
; interrupt
BCF INTCON3, INT1IF
MOVLW .5
MOVWF lowerlimit
MOVLW .250
```

```
MOVWF upperlimit
MOVLW .5 ; Set starting delay to 0.5 seconds
MainL: ; Main loop
BTG PORTB,5 ; LED Toggle
CALL Delay
BRA MainL
; Delay function - uses the WREG value as the number of 1/10
; of seconds delay period
Delay:
MOVWF countOD
DelayOL:
CLRF countID
DelayIL:
INCF countID
BNZ DelayIL
DECF countOD
BNZ DelayOL
RETURN
end ; Interrupt Example Program
```

### Experiment 2

We typed code for Edbot to blink LED on and off every 0.5 second> When INT0 occurs, EDbot turned clockwise in circles for 2 seconds. When INT1 occurs, EDbot turned counter clockwise for 2 seconds.

**Pseudo Code:**

```
Step 1: Initialize input/output pins on the PICmicro (18F1220)
Step 2: Enable interrupts 0 and 1
Step 3: Set INT0 as high priority at address 0x08
Step 4: Set INT1 as low priority at address 0x18
Step 5: Enable motors and drive forward for approximately 0.5 seconds
Step 6: Reverse motors and drive backwards for approximately 0.5 seconds
Step 7: Repeat steps 5 and 6 indefinitely until INT0, INT1, or Reset are enabled
Step 8: If INT0 is enabled, drive left motor forward and right motor backwards
Step 9: Clear interrupt flag and return from interrupt
Step 10: If INT1 is enabled, drive right motor forward and left motor backwards
Step 11: Clear interrupt flag and return from interrupt
```

**Configured code:**

```
list p=18F1220 ; Processor type
radix hex ; Default radix for data
config WDT=OFF, LVP=OFF, OSC = INTIO2 ; Disable Watchdog timer, Low V. Prog,
; and RA6 as a clock
#define countOD 0x80
#define countID 0x81
```

```
#include p18f1220.inc ; This header file includes address and
; bit definitions for all SFRs
org 0x000 ; Executes after Reset
GOTO MainLoop
org 0x008 ; Executes after high priority interrupt
GOTO HPRIO
org 0x018 ; Executes after low priority interrupt
GOTO LPRIO
org 0x20
HPRIO: ; High priority interrupt
BSF PORTA, 7 ; Drive left motor forward
BCF PORTA, 6 ; Drive right motor backward
MOVLW .10
Call Delay
BCF INTCON, INT0IF ; Clear Interrupt 0
RETFIE ; Return from interrupt
LPRIO: ; Low priority interrupt
BTFSC INTCON3, INT1IF ; Check for Interrupt 1
BRA Intr1
RETFIE ; Return from interrupt
Intr1: ; Take care of Interrupt 1
BSF PORTA, 6 ; Drive right motor forward
BCF PORTA, 7 ; Drive left motor backward
MOVLW .10
Call Delay
BCF INTCON3, INT1IF ; Clear interrupt 1 flag
RETFIE ; Return from interrupt
MainLoop: ; Initialization code to be executed
; during Reset
CLRF PORTA ; Initialize PORTA
CLRF PORTB ; Initialize PORTB
MOVLW 0x7F ; Set all A\D Converter Pins as
MOVWF ADCON1 ; digital I/O pins
MOVLW 0x0D ; Value used to initialize data direction
MOVWF TRISA ; Set Port A direction
MOVLW 0xC7 ; Value used to initialize data direction
MOVWF TRISB ; Set Port B direction
MOVLW 0x00 ; Clear WREG
BSF INTCON, GIE ; Enable interrupts
BSF INTCON, PEIE ; Enable all interrupts
BSF INTCON, INT0IE ; Enable interrupt 0
BSF INTCON3, INT1IE ; Enable interrupts 1
BCF INTCON3, INT1IP ; INT1 is set to low priority
BSF RCON, IPEN ; Enable priority levels on interrupts
BCF INTCON, INT0IF ; Flags must be cleared to allow an
; interrupt
BCF INTCON3, INT1IF
Loop1: ; Main loop
BSF PORTB, 3 ; Enable left motor
BSF PORTA, 7 ; Drive left motor forward
BSF PORTB, 4 ; Enable right motor
BSF PORTA, 6 ; Drive right motor forward
MOVLW .5
Call Delay
BCF PORTA, 7 ; Drive left motor backward
BCF PORTA, 6 ; Drive right motor backward
MOVLW .5
Call Delay
BRA Loop1
Delay:
MOVWF countOD
DelayOL:
CLRF countID
DelayIL:
INCF countID
BNZ DelayIL
DECF countOD
BNZ DelayOL
RETURN
end ; End of program
```

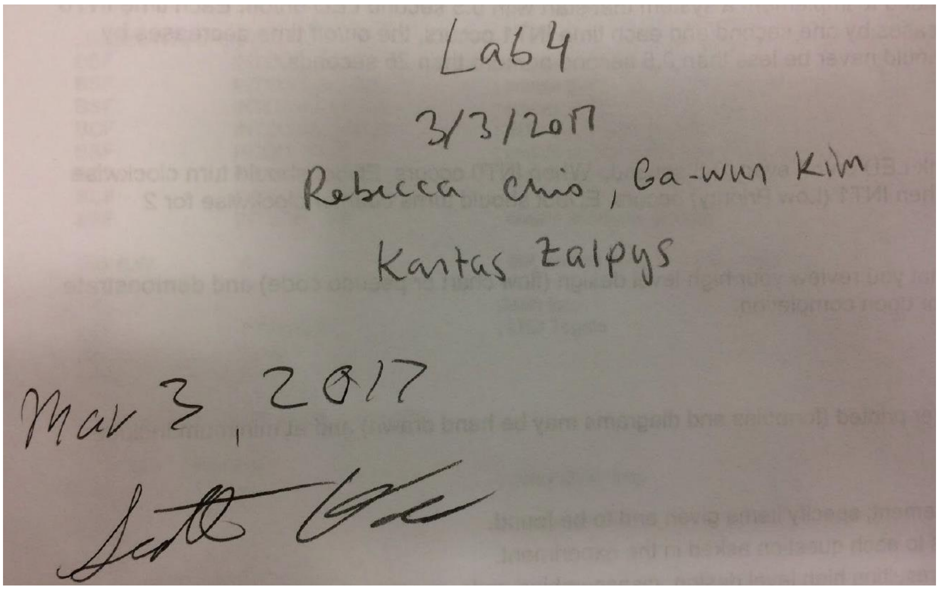
**Learn from this lab**

- **We learned what is interrupt and**
- **We learned how to make interrupt code and use interrupt to EDbot.**

**Conclusion (New experiment)**

This lab was pretty interesting experiments and experience; because I confirmed the interrupt conception in actual lab. Actually, I could not understand what interrupt is, exactly, but, from this lab, I could make sense interrupt.

**Signature for certificate.**



Lab4

3/3/2017

Rebecca Chio, Ga-wun Kim

Kastas Zalpys

Mar 3, 2017

Scott