**ECE-3226-50:** Lab #8

Pulse Width Modulation/Analog-to-digital conversion

Robert Campbell

**Objective:**

The purpose of this lab is to gain familiarity the pulse width modulation subsystem and analog-to-digital converters on the Atmega32A microcontroller by using an input signal from a temperature signal to change the generated pulse.

**Equipment:**

AVR Studio 7

STK500 Starter Kit

JTAG mkII Debugger

Servo motor

LM34CA Temperature Sensor

**Procedure:**

Part 2 & 3: Generating Variable Pulsewidth Signals

Description: Write a program that generates signals with a pulse-width that can be changed by pressing one of the input buttons (SW0 through SW6) attached to PortA.

The output signal will be to OC0 (PB3).

;

; Lab8\_2.asm

;

; Created: 11/19/2019 8:20:28 AM

; Author : campbellrobert

;

; R17 used for switch input

.org 0x0030

;=========

Init:

; initialize PortA

ldi r16, 0x00 ;PA0-PA7 as inputs

out DDRA, r16

ldi r16, 0xFF

out PORTA, r16 ; enable pull-up resisters

SBI DDRB, 3; set PB3 as output

ldi r16, 8

out ocr0, r16

ldi r16, 0b01101011

out tccr0, r16

InLoop:

in r17, pina

sbrs r17, 0

jmp pw0

sbrs r17, 1

jmp pw1

sbrs r17, 2

jmp pw2

sbrs r17, 3

jmp pw3

sbrs r17, 4

jmp pw4

sbrs r17, 5

jmp pw5

sbrs r17, 6

jmp pw6

rjmp InLoop

end: rjmp end

;Rotation in 30 deg increments

pw0:

ldi r16, 8

out ocr0, r16

ldi r16, 0b01101011

out tccr0, r16

jmp InLoop

pw1:

ldi r16, 12

out ocr0, r16

ldi r16, 0b01101011

out tccr0, r16

jmp InLoop

pw2:

ldi r16, 16

out ocr0, r16

ldi r16, 0b01101011

out tccr0, r16

jmp InLoop

pw3:

ldi r16, 20

out ocr0, r16

ldi r16, 0b01101011

out tccr0, r16

jmp InLoop

pw4:

ldi r16, 24

out ocr0, r16

ldi r16, 0b01101011

out tccr0, r16

jmp InLoop

pw5:

ldi r16, 28

out ocr0, r16

ldi r16, 0b01101011

out tccr0, r16

jmp InLoop

pw6:

ldi r16, 32

out ocr0, r16

ldi r16, 0b01101011

out tccr0, r16

jmp InLoop

Observations: The program largely performed as expected but required calibrating the pulse-width of the signal. The original values were between 15 and 30 in increments of 3, however this cause the servo motor to over-rotate and vibrate. The initial value was calibrated down to 8 and increments for 4 were used to approximate a 30degree rotation of the motor

Part 3: Combining Pulse-Width Modulation with Analog-to-Digital Conversion

Description: Modify the previous program so that it will accept and convert an analog input. It will now use that input in place of the buttons to change the pulse width and display the temperature on the board’s LEDs.

The signal output is connected the same as in Part 2, with the LEDs connected to PortD. Use the ADC0 (PA0) pin as the analog input.

;

; Lab8\_6.asm

;

; Created: 11/19/2019 8:20:28 AM

; Author : campbellrobert

;

; r18 used for conversion values

rjmp Init

.org 0x0030

;=========

Init:

; initialize ADC

ldi r16, 0xE0 ; 2.56Vref, left adjusted, using PA0(ADC0)

out ADMUX, r16

ldi r16, 0xB4 ; Enable ADC, Do not start converting, enable auto-trigger, clear ADIF, disable interrupt, use prescalar 16

out ADCSRA, r16

in r16, SFIOR

ANDI r16, 0x1F ; clear bits 7:5, 000- ---- is what is needed for ADIF Triggered

out SFIOR, R16

;PWM output setupt

SBI DDRB, 3; set PB3 as output

ldi r16, 8

out ocr0, r16

ldi r16, 0b01101011

out tccr0, r16

;LED Setup, using PortC. Initialize to 0xFF for all off (active low)

ldi r16, 0xFF

out DDRD, r16

out PORTD, r16

SBI ADCSRA, 6; Start conversion

InLoop:

SBIS ADCSRA, 4

rjmp InLoop

IN r18, adcl ; Stepsize is 1/4 of the input voltage step size, so the two bits in the low byte do not matter

in r18, adch

mov r17, r18

com r17

out PORTD, r17

cpi r18, 32

brlo pw0

cpi r18, 41

brlo pw1

cpi r18, 51

brlo pw2

cpi r18, 61

brlo pw3

cpi r18, 71

brlo pw4

cpi r18, 81

brlo pw5

cpi r18, 91

brlo pw6

rjmp pw0 ; if all tests fail, set output to 0 deg

end: rjmp end

;Rotation in 30 deg increments

pw0:

ldi r16, 8

out ocr0, r16

ldi r16, 0b01101011

out tccr0, r16

;com r18

;out PORTC, r18

SBI ADCSRA, 4

jmp InLoop

pw1:

ldi r16, 12

out ocr0, r16

ldi r16, 0b01101011

out tccr0, r16

;com r18

;out PORTC, r18

SBI ADCSRA, 4

jmp InLoop

pw2:

ldi r16, 16

out ocr0, r16

ldi r16, 0b01101011

out tccr0, r16

;com r18

;out PORTC, r18

SBI ADCSRA, 4

jmp InLoop

pw3:

ldi r16, 20

out ocr0, r16

ldi r16, 0b01101011

out tccr0, r16

;com r18

;out PORTC, r18

SBI ADCSRA, 4

jmp InLoop

pw4:

ldi r16, 24

out ocr0, r16

ldi r16, 0b01101011

out tccr0, r16

;com r18

;out PORTC, r18

SBI ADCSRA, 4

jmp InLoop

pw5:

ldi r16, 28

out ocr0, r16

ldi r16, 0b01101011

out tccr0, r16

;com r18

;out PORTC, r18

SBI ADCSRA, 4

jmp InLoop

pw6:

ldi r16, 32

out ocr0, r16

ldi r16, 0b01101011

out tccr0, r16

;com r18

;out PORTC, r18

SBI ADCSRA, 4

jmp InLoop

Observations: The initial program did not work as intended, as we initially used PortC for the LED output. For some reason, using PortC cause the display to be unstable with several LEDs flickering constantly and the stable LEDs displaying a pattern that did not make sense. Switching over to PortD corrected this issue.

|  |  |
| --- | --- |
| Temperature Displayed (binary) | Servo Angle in Degrees |
| 0001 1111 | 0 |
| 0010 0000 | 30 |
| 0010 1001 | 60 |
| 0011 0011 | 90 |
| 0011 0101 (highest observed temp) | 90 |

**Discussion/Conclusion:**

The purpose of this experiment was to gain familiarity with pulse-width modulation and converting incoming analog signals in to digital signals.

Generally, there were not any significant issues with the programs that we wrote. In Parts 2 & 3, our initial values were drastically incorrect but were corrected with trial and error. We ended up finding the minimum output that the servo could tolerate stably and an interval that approximated a 30 degree rotation. Unfortunately, that rotation was just an approximation and deviation became more apparent at the larger values (150 looked to be closer to 160). This may also have been due to human error holding the device and protractor stable.

Part 4 should have been completed quickly, but was derailed by troubleshooting the unstable display. The code was reviewed for inconsistencies and eventually verified by peers. We never did find the problem ourselves, but took guidance from the instructor who suggested changing the output to Port D. That resolved the issue and the lab was completed without further issue.

This experiment helped us learn to use the pulse-width modulation subsystem in order to drive a motor as well as accept and make use of an incoming analog signal by way of the analog-to-digital converter subsystem.