8.6)

; 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 DDRC, r16

out PORTC, 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

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