Michelle Mata

CPE301 – SPRING 2016

Design Assignment 5

**DO NOT REMOVE THIS PAGE DURING SUBMISSION:**

The student understands that all required components should be submitted in complete for grading of this assignment.

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **SUBMISSION ITEM** | **COMPLETED (Y/N)** | **MARKS**  **(/MAX)** |
| 0. | COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS |  |  |
| 1. | INITIAL CODE OF TASK 1 |  |  |
| 2. | INCREMENTAL / DIFFERENTIAL CODE OF TASK 2 |  |  |
| 3. | INCREMENTAL / DIFFERENTIAL CODE OF TASK 3 |  |  |
| 6. | SCHEMATICS |  |  |
| 7. | SCREENSHOTS OF EACH TASK OUTPUT |  |  |
| 8. | SCREENSHOT OF EACH DEMO |  |  |
| 9. | VIDEO LINKS OF EACH DEMO |  |  |
| 10. | GOOGLECODE LINK OF THE DA |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 0. | COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS |  |  |

ATmega328P

DC motor

Servo Motor

Stepper Motor

|  |  |  |  |
| --- | --- | --- | --- |
| 1. | INITIAL CODE OF TASK 1 |  |  |

#define F\_CPU 8000000UL //XTAL = 8MHZ

#include <avr/io.h>

#include <util/delay.h>

int main() {

DDRB = 0xFF; //make PORTB an output

DDRD = 0xFF; //make PORTD an output

DDRB |=(1<<DDB1); // make OC1A an output

TCCR1A |= 1<<COM1A1 |0 << COM1A0 | 0<<COM1B0 | 0 <<FOC1A | 0<<FOC1B | 1 << WGM11 | 0<<WGM10;

TCCR1B |= 0 << ICNC1| 0 <<ICES1 | 1 << WGM13 | 1 << WGM12 | 0<<CS12 | 1<<CS11 | 1 << CS10;

ICR1 = 2499; //F\_CPU / (N \* F\_pwm) - 1

ADCSRA=0x87; //make ADC enable and select ck/128

ADMUX=0xC0; // 2.56V Vref, ADC0 single ended input data will be right-justified

while(1) {

PORTB = PORTB | (1<<4);

\_delay\_ms(1500); //75% duty cycle on

PORTB = PORTB & (~(1<<4));

\_delay\_ms(1500);

ADCSRA |= 1 << ADSC; // start conversion

while((ADCSRA & (1<<ADIF)) ==0); //wait for conversion to finish

OCR1A=(255\*ADC)/1023; // OCR1A is given the input voltage

PORTD=ADCL; //give the low byte to PORT D

PORTB=ADCH; //give the high byte to PORTB

}

return 0 ;

}

|  |  |  |  |
| --- | --- | --- | --- |
| 2. | INITIAL CODE OF TASK 2 |  |  |

#define F\_CPU 8000000UL //XTAL = 8MHZ

#include <avr/io.h>

#include <util/delay.h>

int main() {

DDRB = 0xFF; //make PORTB an output

DDRD = 0xFF; //make PORTD an output

DDRB |=(1<<DDB1); // make OC1A an output

TCCR1A |= 1<<COM1A1 |0 << COM1A0 | 0<<COM1B0 | 0 <<FOC1A | 0<<FOC1B | 1 << WGM11 | 0<<WGM10;

TCCR1B |= 0 << ICNC1| 0 <<ICES1 | 1 << WGM13 | 1 << WGM12 | 0<<CS12 | 1<<CS11 | 1 << CS10;

ICR1 = 2499; //F\_CPU / (N \* F\_pwm) - 1

ADCSRA=0x87; //make ADC enable and select ck/128

ADMUX=0xC0; // 2.56V Vref, ADC0 single ended input data will be right-justified

while(1) {

PORTB = PORTB | (1<<4);

\_delay\_ms(1500); //75% duty cycle on

PORTB = PORTB & (~(1<<4));

\_delay\_ms(1500);

ADCSRA |= 1 << ADSC; // start conversion

while((ADCSRA & (1<<ADIF)) ==0); //wait for conversion to finish

OCR1A=(255\*ADC)/1023; // OCR1A is given the input voltage

PORTD=ADCL; //give the low byte to PORT D

PORTB=ADCH; //give the high byte to PORTB

}

return 0 ;

}

|  |  |  |  |
| --- | --- | --- | --- |
| 3. | INITIAL CODE OF TASK 3 |  |  |

#define F\_CPU 8000000UL //XTAL = 8MHZ

#include <avr/io.h>

#include <util/delay.h>

int main() {

DDRB = 0xFF; //make PORTB an output

DDRD = 0xFF; //make PORTD an output

DDRB |=(1<<DDB1); // make OC1A an output

TCCR1A |= 1<<COM1A1 |0 << COM1A0 | 0<<COM1B0 | 0 <<FOC1A | 0<<FOC1B | 1 << WGM11 | 0<<WGM10; //Prescaler=64

TCCR1B |= 0 << ICNC1| 0 <<ICES1 | 1 << WGM13 | 1 << WGM12 | 0<<CS12 | 1<<CS11 | 1 << CS10;

ICR1 = 2499; //F\_CPU / (N \* F\_pwm) - 1

ADCSRA=0x87; //make ADC enable and select ck/128

ADMUX=0xC0; // 2.56V Vref, ADC0 single ended input data will be right-justified

while(1) {

PORTB = PORTB | (1<<4);

\_delay\_ms(1500); //75% duty cycle on

PORTB = PORTB & (~(1<<4));

\_delay\_ms(1500);

ADCSRA |= 1 << ADSC; // start conversion

while((ADCSRA & (1<<ADIF)) ==0); //wait for conversion to finish

OCR1A=(255\*ADC)/1023; // OCR1A is given the input voltage

PORTD=ADCL; //give the low byte to PORT D

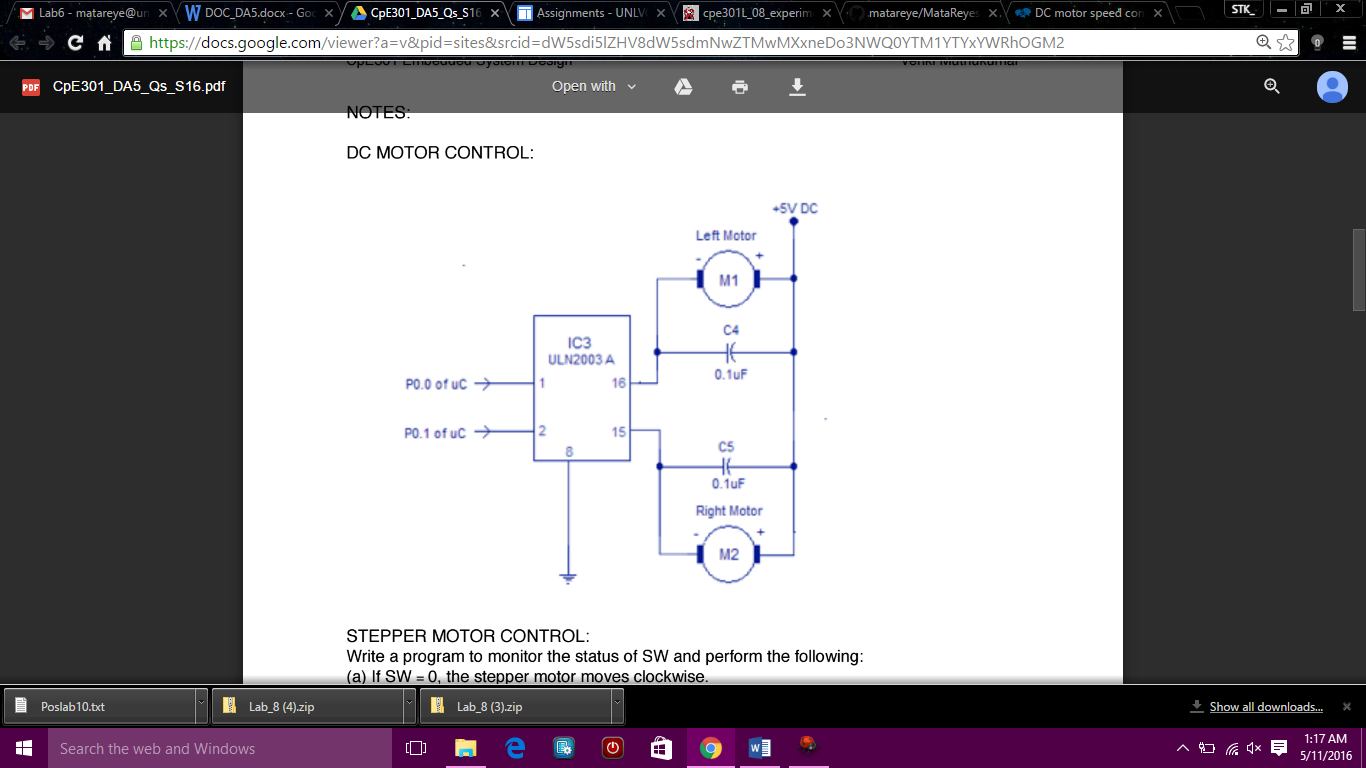
PORTB=ADCH; //give the high byte to PORTB

}

return 0 ;

}

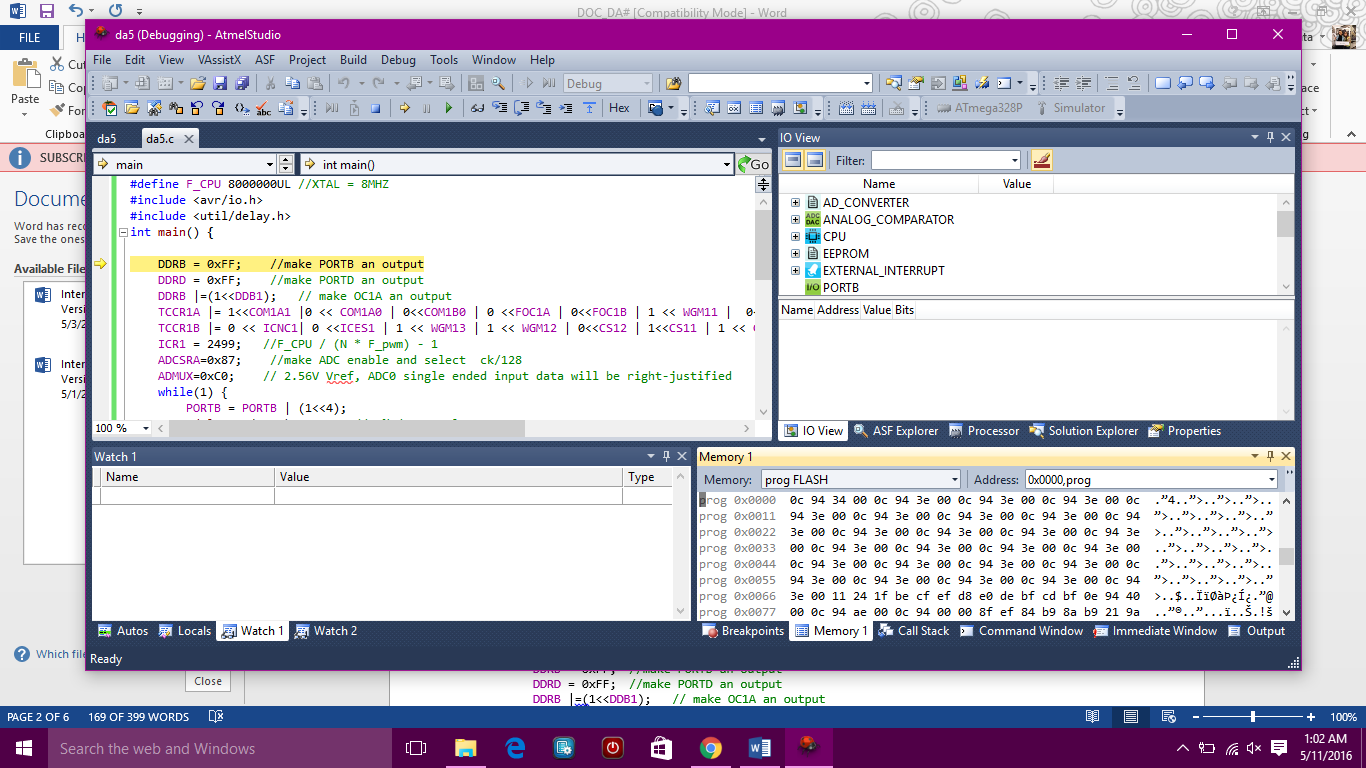
|  |  |  |  |
| --- | --- | --- | --- |
| 6. | SCHEMATICS |  |  |



|  |  |  |  |
| --- | --- | --- | --- |
| 7. | SCREENSHOTS OF EACH TASK OUTPUT |  |  |

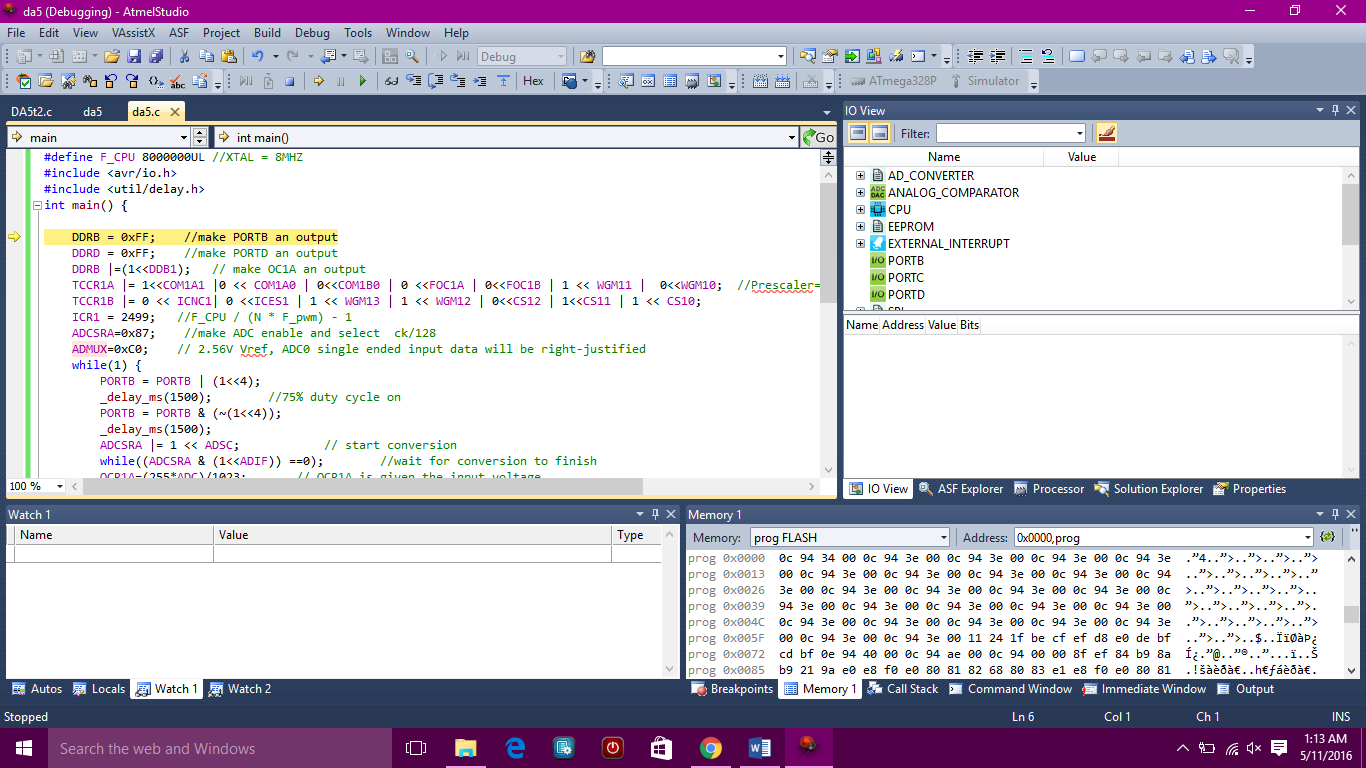
TASK 1/A:

Write an AVR C program to control the speed of the DC Motor using a potentiomenter connected to any of the analog-in port



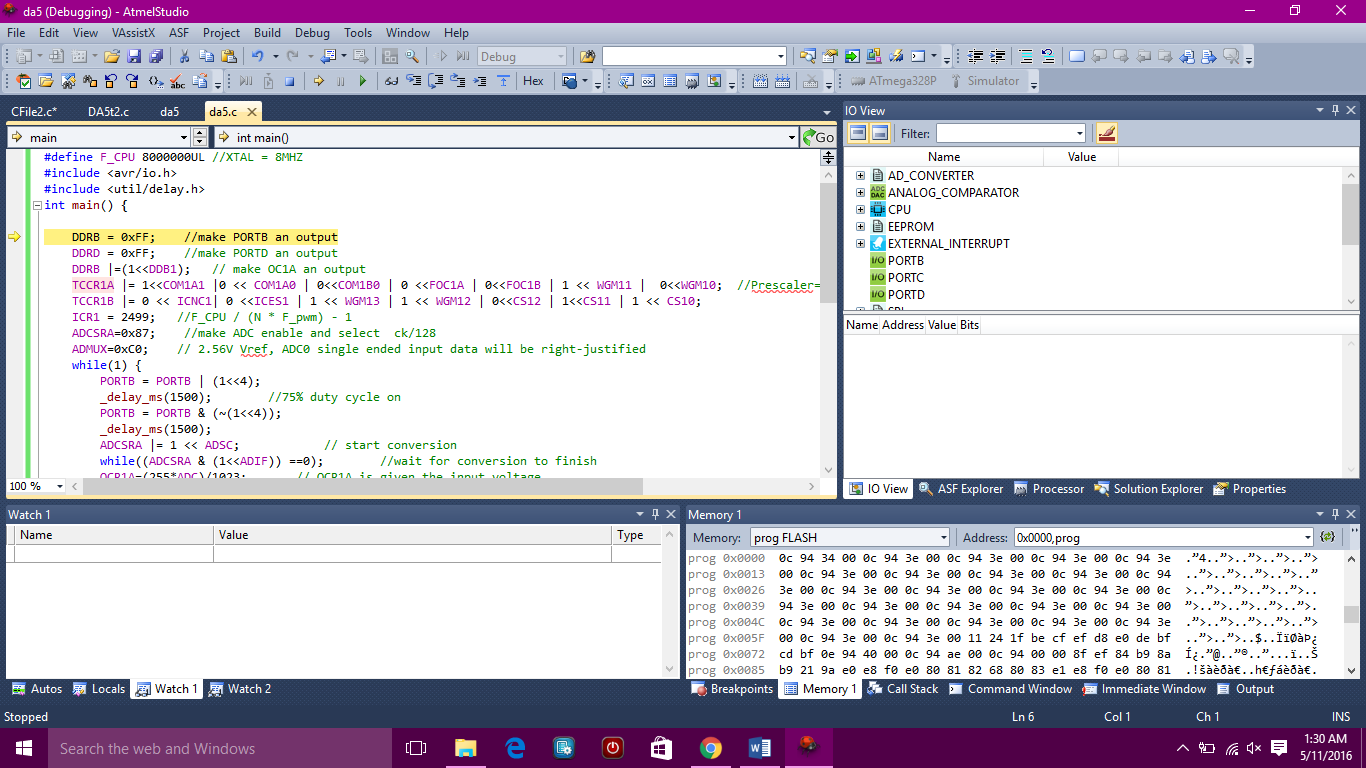
TASK 2:

Write an AVR C program to control the speed of the Stepper Motor using a potentiometer connected to any of the analog-in port



TASK 3:

Write an AVR C program to control the speed of the Servo Motor using a potentiometer connected to any of the analog-in port



|  |  |  |  |
| --- | --- | --- | --- |
| 8. | SCREENSHOT OF EACH DEMO |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 9. | VIDEO LINKS OF EACH DEMO |  |  |
| http:// @youtube | | | |
| 10. | GOOGLECODE LINK OF THE DA |  |  |
| https://github.com/matareye/MataReyesCPE301s16/tree/master | | | |

**Student Academic Misconduct Policy**

<http://studentconduct.unlv.edu/misconduct/policy.html>

“This assignment submission is my own, original work”.

Michelle Mata