Lab – 6 Datasheet

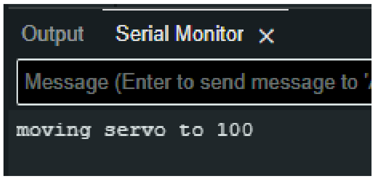
CpE 4010: Sensors, Actuators and Integration

Name: Anindita Deb	KSU ID: 000922115

From procedure 3:

Insert the screenshot of your Serial Monitor showing a selected number (angle) here:

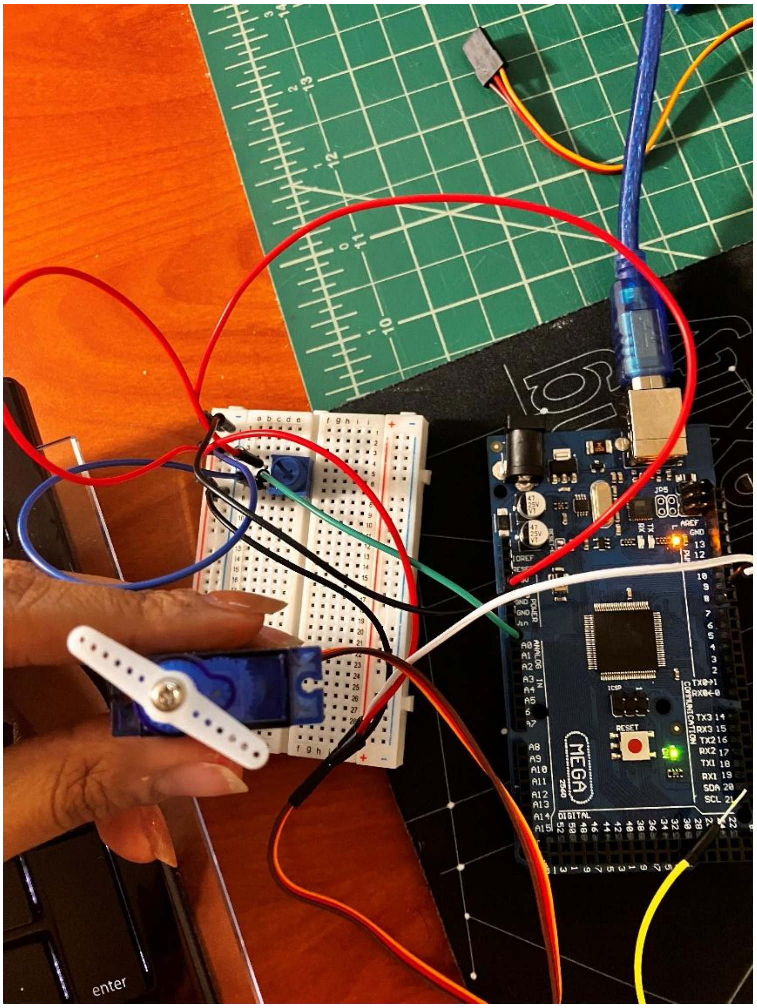
“5”



```
moving servo to 100
```

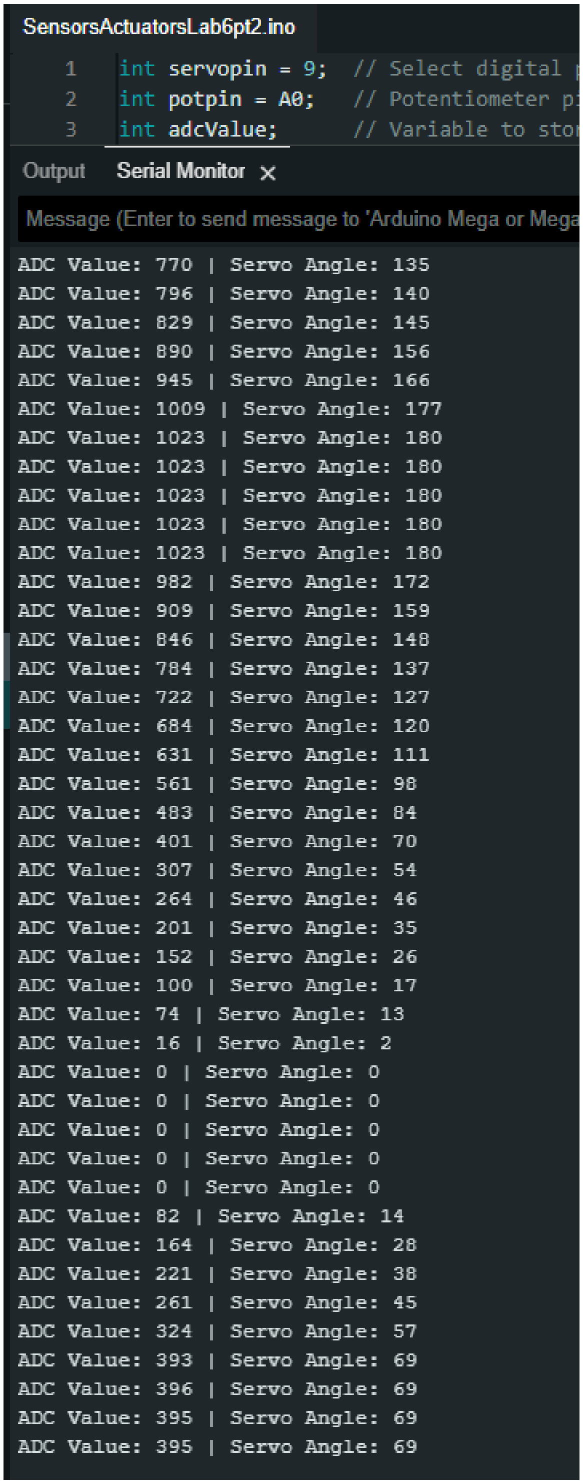
From procedure 4:

Insert picture of your modified circuit showing a servo arm angle associated with a particular potentiometer setting here:



From procedure 5:

Insert a screenshot of your Serial Monitor window showing various ADC output/servo angle values here:

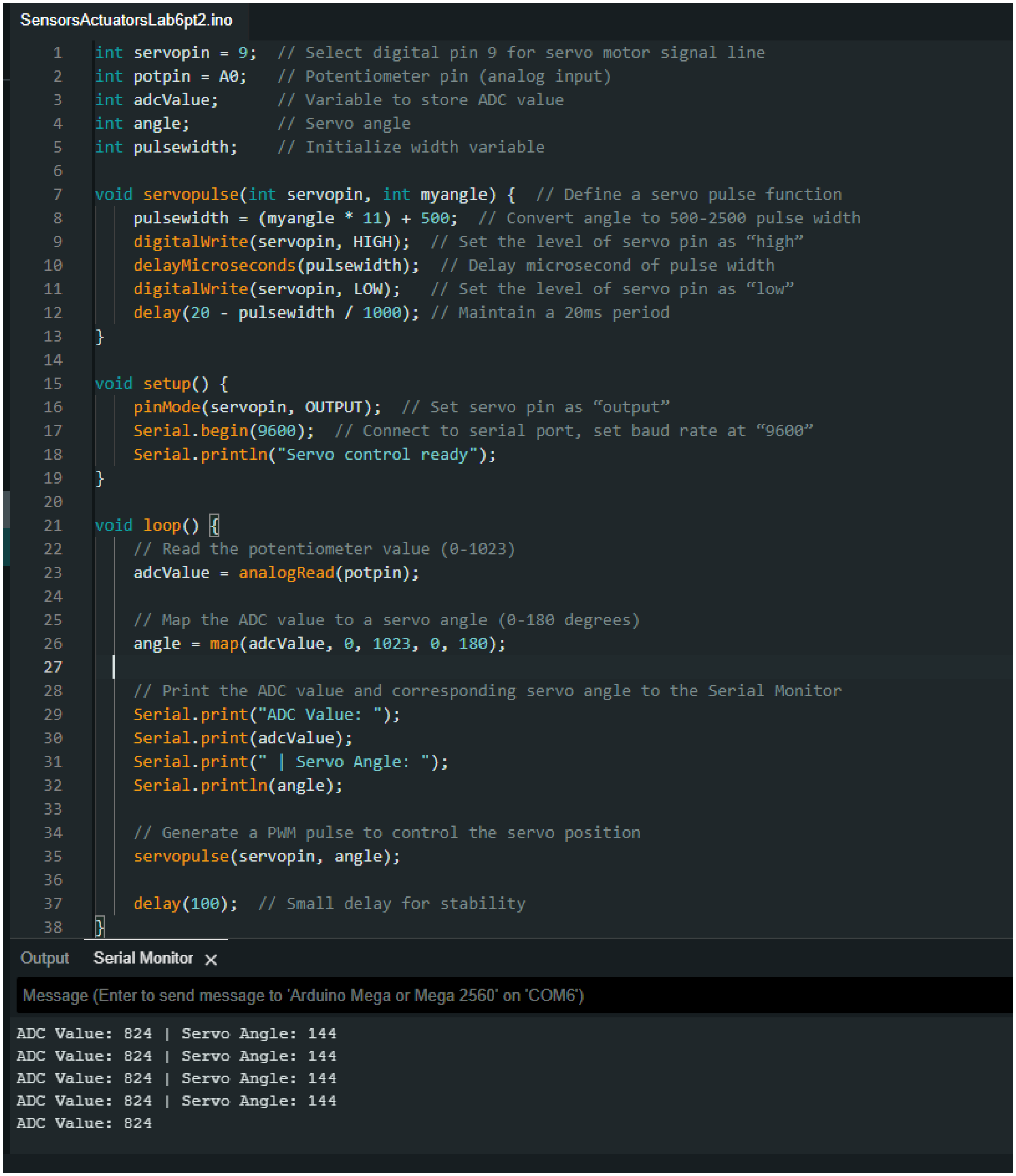

SensorsActuatorsLab6pt2.ino
Output
Serial Monitor
Message (Enter to send message to 'Arduino Mega or Mega

```
int servopin = 9;   // Select digital p
int potpin = A0;    // Potentiometer pi
int adcValue;       // Variable to stor
```

```
ADC Value: 770 | Servo Angle: 135
ADC Value: 796 | Servo Angle: 140
ADC Value: 829 | Servo Angle: 145
ADC Value: 890 | Servo Angle: 156
ADC Value: 945 | Servo Angle: 166
ADC Value: 1009 | Servo Angle: 177
ADC Value: 1023 | Servo Angle: 180
ADC Value: 1023 | Servo Angle: 180
ADC Value: 1023 | Servo Angle: 180
ADC Value: 1023 | Servo Angle: 180
ADC Value: 1023 | Servo Angle: 180
ADC Value: 982 | Servo Angle: 172
ADC Value: 909 | Servo Angle: 159
ADC Value: 846 | Servo Angle: 148
ADC Value: 784 | Servo Angle: 137
ADC Value: 722 | Servo Angle: 127
ADC Value: 684 | Servo Angle: 120
ADC Value: 631 | Servo Angle: 111
ADC Value: 561 | Servo Angle: 98
ADC Value: 483 | Servo Angle: 84
ADC Value: 401 | Servo Angle: 70
ADC Value: 307 | Servo Angle: 54
ADC Value: 264 | Servo Angle: 46
ADC Value: 201 | Servo Angle: 35
ADC Value: 152 | Servo Angle: 26
ADC Value: 100 | Servo Angle: 17
ADC Value: 74 | Servo Angle: 13
ADC Value: 16 | Servo Angle: 2
ADC Value: 0 | Servo Angle: 0
ADC Value: 0 | Servo Angle: 0
ADC Value: 0 | Servo Angle: 0
ADC Value: 0 | Servo Angle: 0
ADC Value: 0 | Servo Angle: 0
ADC Value: 82 | Servo Angle: 14
ADC Value: 164 | Servo Angle: 28
ADC Value: 221 | Servo Angle: 38
ADC Value: 261 | Servo Angle: 45
ADC Value: 324 | Servo Angle: 57
ADC Value: 393 | Servo Angle: 69
ADC Value: 396 | Servo Angle: 69
ADC Value: 395 | Servo Angle: 69
ADC Value: 395 | Servo Angle: 69
```

From procedure 6:

Insert a screenshot of your IDE code window showing your modified source code here:


SensorsActuatorsLab6pt2.ino
Output
Serial Monitor
Message (Enter to send message to 'Arduino Mega or Mega 2560' on 'COM6')

```
int servopin = 9;  // Select digital pin 9 for servo motor signal line
int potpin = A0;   // Potentiometer pin (analog input)
int adcValue;      // Variable to store ADC value
int angle;         // Servo angle
int pulsewidth;    // Initialize width variable

void servopulse(int servopin, int myangle) {  // Define a servo pulse function
    pulsewidth = (myangle * 11) + 500;  // Convert angle to 500-2500 pulse width
    digitalWrite(servopin, HIGH);  // Set the level of servo pin as “high”
    delayMicroseconds(pulsewidth);  // Delay microsecond of pulse width
    digitalWrite(servopin, LOW);  // Set the level of servo pin as “low”
    delay(20 - pulsewidth / 1000); // Maintain a 20ms period
}

void setup() {
    pinMode(servopin, OUTPUT);  // Set servo pin as “output”
    Serial.begin(9600);  // Connect to serial port, set baud rate at “9600”
    Serial.println("Servo control ready");
}

void loop() {
    // Read the potentiometer value (0-1023)
    adcValue = analogRead(potpin);

    // Map the ADC value to a servo angle (0-180 degrees)
    angle = map(adcValue, 0, 1023, 0, 180);

    // Print the ADC value and corresponding servo angle to the Serial Monitor
    Serial.print("ADC Value: ");
    Serial.print(adcValue);
    Serial.print(" | Servo Angle: ");
    Serial.println(angle);

    // Generate a PWM pulse to control the servo position
    servopulse(servopin, angle);

    delay(100);  // Small delay for stability
}
```

```
ADC Value: 824 | Servo Angle: 144
ADC Value: 824 | Servo Angle: 144
ADC Value: 824 | Servo Angle: 144
ADC Value: 824 | Servo Angle: 144
ADC Value: 824
```

Conclusions:

The second part of the lab took the longest to figure out with the potentiometer determining the amount the servo motor would have to move. I realized eventually that I had the wiring wrong for a good portion of it and re did it on a slight larger/ more clear breadboard.