Midterm Requirement for the System Integration and Architecture 2 subject.

Instructions: considering your last semester project and grouping, you are expected to submit the following.

a. A cover page of your project.
b. Final Design (Presentable Image) of your previous semester project.
c. The lines of code being used to develop the project.
d. Every single line of your code must be explained and has a comment to identify the usage.
e. Line of codes must be in sequence from top process to bottom process.
f. Print out and submit on or before the Midterm Exam.

Examples on how to explain and put a comment for every line of code.

Sample 1.

This is the simplest form of input with only two possible states: on or off. This example reads a simple switch or pushbutton connected to pin2. When the switch is closed the input pin will read HIGH and turn on an LED.

```
int ledPin = 13;            // output pin for the LED
int inPin = 2;              // input pin (for a switch)

void setup()
{
  pinMode(ledPin, OUTPUT);  // declare LED as output
  pinMode(inPin, INPUT);    // declare switch as input
}

void loop()
{
  if (digitalRead(inPin) == HIGH) // check if input is HIGH
  {
    digitalWrite(ledPin, HIGH);   // turns the LED on
    delay(1000);                  // pause for 1 second
    digitalWrite(ledPin, LOW);    // turns the LED off
    delay(1000);                  // pause for 1 second
  }
}
```

Sample 2.

Using a potentiometer and one of the Arduino's analog-to-digital conversion (ADC) pins it is possible to read analog values from 0-1024. The following example uses a potentiometer to control an LED's rate of blinking.

```
int potPin = 0;    // input pin for the potentiometer
int ledPin = 13;   // output pin for the LED

void setup()
{
  pinMode(ledPin, OUTPUT);  // declare ledPin as OUTPUT
}

void loop()
{
  digitalWrite(ledPin, HIGH);    // turns ledPin on
  delay(analogRead(potPin));     // pause program
  digitalWrite(ledPin, LOW);     // turns ledPin off
  delay(analogRead(potPin));     // pause program
}
```

Sample 3.

Variable resistors include CdS light sensors, thermistors, flex sensors, and so on.
This example makes use of a function to read the analog value and set a delay time.
This controls the speed at which an LED brightens and dims.

```
int ledPin    =  9;    // PWM pin for the LED
int analogPin =  0;    // variable resistor on analog pin 0

void setup(){}        // no setup needed

void loop()
{
  for (int i=0; i<=255; i++)  // ascending value for i
  {
    analogWrite(ledPin, i);   // sets brightess level to i
    delay(delayVal());        // gets time value and pauses
  }
  for (int i=255; i>=0; i--)  // descending value for i
  {
    analogWrite(ledPin, i);   // sets brightess level to i
    delay(delayVal());        // gets time value and pauses
  }
}

int delayVal()
{
  int v;                      // create temporary variable
  v  = analogRead(analogPin); // read analog value
  v /= 8;                     // convert 0-1024 to 0-128
  return v;                   // returns final value
}
```

Sample 4.

Hobby servos are a type of self-contained motor that can move in a 180º arc. All that is needed is a pulse sent every 20ms. This example uses a servoPulse function to move the servo from 10º -170º and back again.

```
int servoPin = 2;   // servo connected to digital pin 2
int myAngle;        // angle of the servo roughly 0-180
int pulseWidth;     // servoPulse function variable

void setup()
{
  pinMode(servoPin, OUTPUT);   // sets pin 2 as output
}

void servoPulse(int servoPin, int myAngle)
{
  pulseWidth = (myAngle * 10) + 600;  // determines delay
  digitalWrite(servoPin, HIGH);       // set servo high
  delayMicroseconds(pulseWidth);      // microsecond pause
  digitalWrite(servoPin, LOW);        // set servo low
}

void loop()
{
  // servo starts at 10 deg and rotates to 170 deg
  for (myAngle=10; myAngle<=170; myAngle++)
  {
    servoPulse(servoPin, myAngle);    // send pin and angle
    delay(20);                        // refresh cycle
  }
  // servo starts at 170 deg and rotates to 10 deg
  for (myAngle=170; myAngle>=10; myAngle--)
  {
    servoPulse(servoPin, myAngle);    // send pin and angle
    delay(20);                        // refresh cycle
  }
}
```