ELP305: DESIGN AND SYSTEM LABORATORY

EXPERIMENT 0
INTRODUCTION TO ARDUINO

GROUP NUMBER: 21
EXPERIMENT DONE ON: 08/01/2018
SUBMITTED ON: 09/01/2018
PROFESSOR: Prof. SMK RAHMAN

SUBMITTED BY:
NAME: AVINASH KUMAR(2015MT10319)
NAME: DRASHTI KHATSURIYA(2015MT10598)
NAME: CHARVI NAHAR(2015MT10595)

AIM: To learn about Arduino and its basic applications like:

1. Blinking of a LED
2. Fading of a LED
3. Analog Read
4. PWM Signal
5. Seven Segment Display using Push Button and Arduino

APPARATUS REQUIRED:

1. Arduino
2. LED
3. Resistors
4. Potentiometer
5. Breadboard
6. Jumper Wires

THEORY:

Arduino is a single-board microcontroller. The hardware consists of an open-source hardware board designed around an 8-bit Atmel AVR microcontroller, or a 32-bit Atmel ARM with complementary components to facilitate programming and incorporation into other circuits. Arduino has total 20 I/O pins. There are three pools of memory in the microcontroller used on AVR -based Arduino boards:

1.Flash memory (program space), is where the Arduino sketch is stored.

2.SRAM (static random access memory) is where the sketch creates and manipulates variables when it runs.

3.EEPROM is memory space that programmers can use to store long-term information.

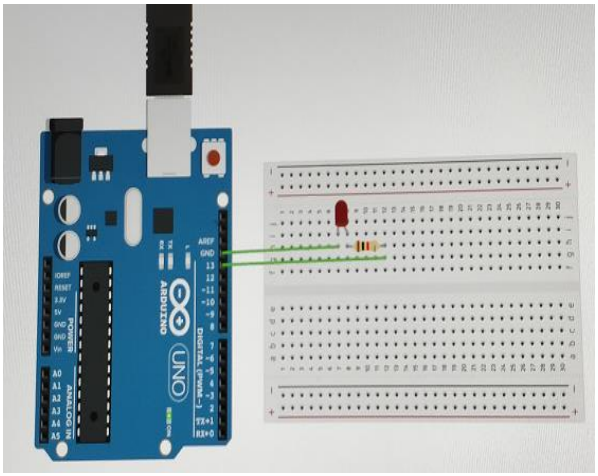
Flash memory and EEPROM memory are non-volatile (the information persists after the power is turned off). SRAM is volatile and will be lost when the power is cycled.

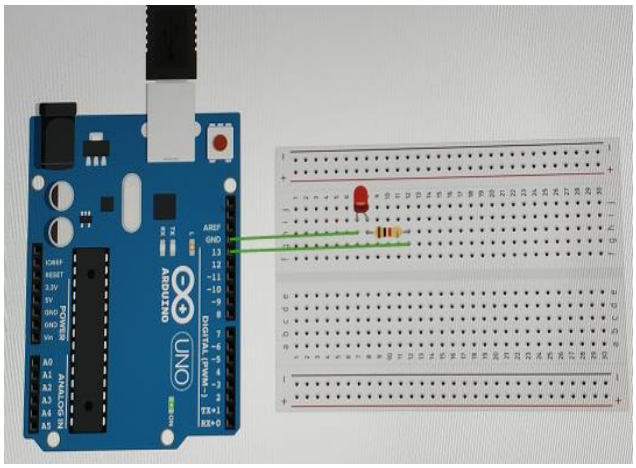
FIG: Arduino UNO

OBSERVATION:

1.BLINKING OF A LED

CIRCUIT DIAGRAM:





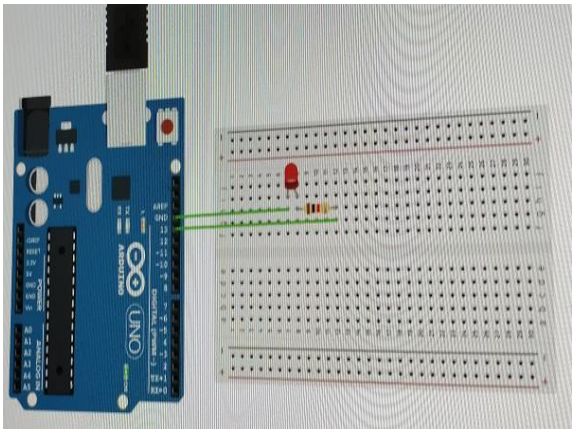
CODE FOR BLINKING A LED:

```
void setup() {

  pinMode(13, OUTPUT);
}


void loop() {
  digitalWrite(LED_BUILTIN, HIGH);
  delay(1000);
  digitalWrite(LED_BUILTIN, LOW);
  delay(1000);
}
```

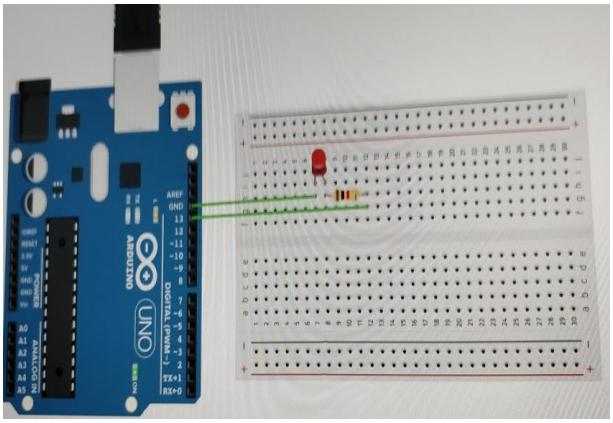
2.FADING OF A LED AND USE OF PWM SINGAL

CIRCUIT DIAGRAM:

AT T = T1



AT T = T2



CODE FOR FADING A LED:

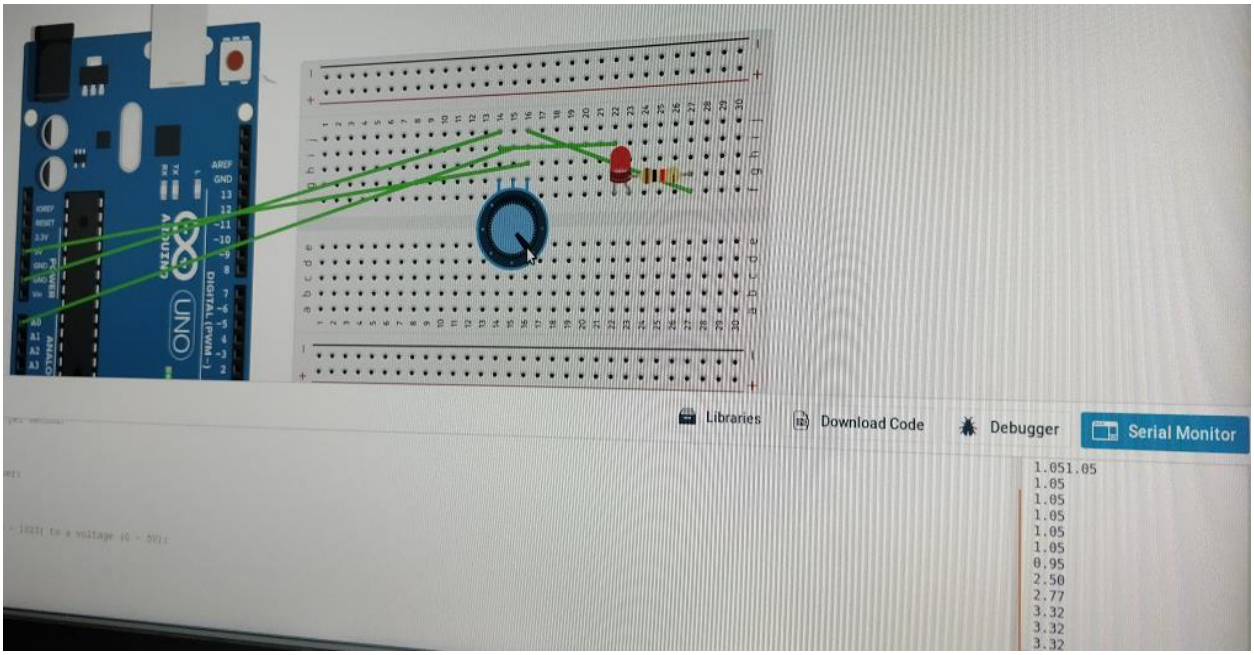
1 (Arduino Uno R3) ⌄ | Upload & Run | Block

```
int led = 3;           // the pin that the LED is attached to
int brightness = 0;    // how bright the LED is
int fadeAmount = 5;    // how many points to fade the LED by

// the setup routine runs once when you press reset:
void setup()  {
  // declare pin 3 to be an output:
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop()  {
  // set the brightness of pin 3:
  analogWrite(led, brightness);

  // change the brightness for next time through the loop:
  brightness = brightness + fadeAmount;

  // reverse the direction of the fading at the ends of the fade:
  if (brightness == 0 || brightness == 255) {
    fadeAmount = -fadeAmount ;
  }
  // wait for 30 milliseconds to see the dimming effect
  delay(30);
}
```

- Pulse Width Modulation, or PWM, is a technique for getting analog results with digital means. The average value of voltage (and current) fed to the load is controlled by turning the switch between supply and load on and off at a fast rate. Digital control is used to create a square wave, a signal switched between on and off. This on-off pattern can simulate voltages in between full on (5 Volts) and off (0 Volts) by changing the portion of the time the signal spends on versus the time that the signal spends off. The duration of "on time" is called the pulse width.
- The Arduino UNO it has 6 digital pins that can be used as PWM outputs (3, 5, 6, 9, 10, and 11). The Arduino can send PWM signal with the **analogWrite()** function.
- Here T2 > T1, Showing that during coming down of pulse, intensity got decreases and this we can clearly see in figure above.

3.ANALOG READ

CIRCUIT DIAGRAM:



- Figure contain the values of output changes with value of resistance.

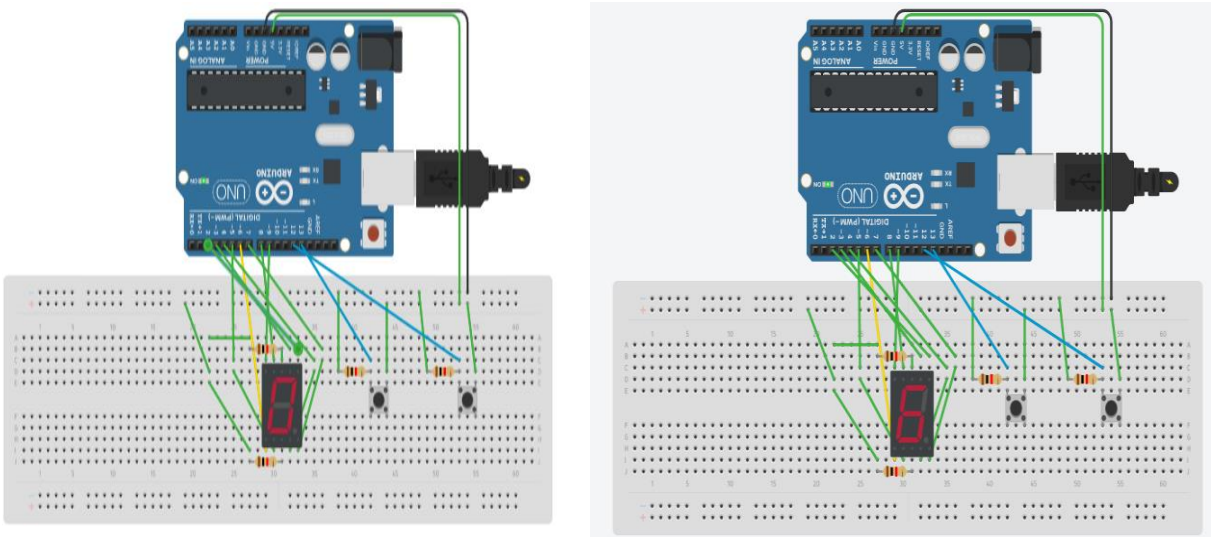
CODE OF ANALOGREAD:

1 (Arduino Uno R3) ⌄ | Upload & Run | Block

```
1  void setup() {
2    // initialize serial communication at 9600 bits per second:
3    Serial.begin(9600);
4  }
5
6  // the loop routine runs over and over again forever:
7  void loop() {
8    // read the input on analog pin 0:
9    int sensorValue = analogRead(A0);
10   // print out the value you read:
11   Serial.println(sensorValue);
12   delay(1);        // delay in between reads for stability
13 }
```

4.SEVEN SEGMENT DISPLAY USING PUSHBUTTON

CIRCUIT DIAGRAM WITH OUTPUTS:



CODE FOR SEVEN SEGMENT:

1 (Arduino Uno R3) ⌄ | Upload & Run

```
//Initialize the 7 segment pins
int A = 3;
int B = 2;
int C = 4;
int D = 5;
int DP = 7;
int E = 6;
int F = 8;
int G = 9;

//Initialize the push buttons pins, push buttons states, and the counter
int switchUpPin = 13;
int switchDownPin = 12;
int counter = 0;
int buttonUpState = 0;
int lastButtonUpState = 0;
int buttonDownState = 0;
int lastButtonDownState = 0;

void setup()
{
  Serial.begin(9600);

  pinMode(A, OUTPUT);
  pinMode(B, OUTPUT);
  pinMode(C, OUTPUT);
  pinMode(D, OUTPUT);
  pinMode(E, OUTPUT);
  pinMode(F, OUTPUT);
  pinMode(G, OUTPUT);
  pinMode(DP, OUTPUT);

  //Start with the deceimal point off
  digitalWrite(DP ,HIGH);
}
void loop()
{
  //Getting the reads from the buttons
  buttonUpState = digitalRead(switchUpPin);
  buttonDownState = digitalRead(switchDownPin);

  //Detecting button press and getting the button status
  //Do this for the button up
  if (buttonUpState != lastButtonUpState)
  {
    if (buttonUpState == HIGH)
    {
      //Reset the counter to -1
      if(counter == 9)
      {
        counter = -1;
      }
      //Increase the counter by 1
      counter++;
      //Print the counter to the console and calling the function
      Serial.println(counter);
      changeNumber(counter);
      //Delaying by 250 ms
      delay(250);
    }
    else
    {
        Serial.println("OFF");
    }
    //Delay to avoid button bouncing
   delay(50);
  }

  //Do this for the button down
  if (buttonDownState != lastButtonDownState)
  {
    if (buttonDownState == HIGH)
```

```
      {
        //Set the counter to 10
        if(counter == 0)
        {
          counter = 10;
        }
        //Decreases the counter by 1
        counter--;
        ////Print the counter to the console and calling the function
        Serial.println(counter);
        changeNumber(counter);
        //Delaying by 250 ms
        delay(250);
      }
      else
      {
          Serial.println("OFF");
      }
      //Delay to avoid button bouncing
     delay(50);
    }
    //Calling the function changeNumber with the arg counter
    changeNumber(counter);
}

//The function to display the numbers
void changeNumber(int buttonPress)
{
  switch (buttonPress)
  {
    //number 0
    case 0:
      digitalWrite(A, LOW);
      digitalWrite(B, LOW);
      digitalWrite(C, LOW);
      digitalWrite(D, LOW);
```

```
        digitalWrite(E, LOW);
        digitalWrite(F, LOW);
        digitalWrite(G, HIGH);
        break;
      //number 1
      case 1:
        digitalWrite(A, HIGH);
        digitalWrite(B, LOW);
        digitalWrite(C, LOW);
        digitalWrite(D, HIGH);
        digitalWrite(E, HIGH);
        digitalWrite(F, HIGH);
        digitalWrite(G, HIGH);
        break;
      //number 2
      case 2:
        digitalWrite(A, LOW);
        digitalWrite(B, LOW);
        digitalWrite(C, HIGH);
        digitalWrite(D, LOW);
        digitalWrite(E, LOW);
        digitalWrite(F, HIGH);
        digitalWrite(G, LOW);
        break;
      //number 3
      case 3:
        digitalWrite(A, LOW);
        digitalWrite(B, LOW);
        digitalWrite(C, LOW);
        digitalWrite(D, LOW);
        digitalWrite(E, HIGH);
        digitalWrite(F, HIGH);
        digitalWrite(G, LOW);
        break;
      //number 4
      case 4:
```

```
        digitalWrite(A, HIGH);
        digitalWrite(B, LOW);
        digitalWrite(C, LOW);
        digitalWrite(D, HIGH);
        digitalWrite(E, HIGH);
        digitalWrite(F, LOW);
        digitalWrite(G, LOW);
        break;
      //number 5
      case 5:
        digitalWrite(A, LOW);
        digitalWrite(B, HIGH);
        digitalWrite(C, LOW);
        digitalWrite(D, LOW);
        digitalWrite(E, HIGH);
        digitalWrite(F, LOW);
        digitalWrite(G, LOW);
        break;
      //number 6
      case 6:
        digitalWrite(A, LOW);
        digitalWrite(B, HIGH);
        digitalWrite(C, LOW);
        digitalWrite(D, LOW);
        digitalWrite(E, LOW);
        digitalWrite(F, LOW);
        digitalWrite(G, LOW);
        break;
      //number 7
      case 7:
        digitalWrite(A, LOW);
        digitalWrite(B, LOW);
        digitalWrite(C, LOW);
        digitalWrite(D, HIGH);
        digitalWrite(E, HIGH);
        digitalWrite(F, HIGH);
        digitalWrite(G, HIGH);
```

```
        //number 8
        case 8:
           digitalWrite(A, LOW);
           digitalWrite(B, LOW);
           digitalWrite(C, LOW);
           digitalWrite(D, LOW);
           digitalWrite(E, LOW);
           digitalWrite(F, LOW);
           digitalWrite(G, LOW);
           break;
        //number 9
        case 9:
           digitalWrite(A, LOW);
           digitalWrite(B, LOW);
           digitalWrite(C, LOW);
           digitalWrite(D, HIGH);
           digitalWrite(E, HIGH);
           digitalWrite(F, LOW);
           digitalWrite(G, LOW);
           break;
    }
}
```

RESULT

We have implemented the circuits successfully and got the result. Our output is coming correct in each section of experiment.

CONCLUSION

We have successfully implemented the circuits and got the result very correct in each section. First, we read about the arduino and its components. Then we did small experiments to understand the concepts. We have learnt a lot about the Arduino. We have also learnt how to write the code in arduino and some of the important functions like: AnalogRead(), Analogwrite(), delay(), pinMode(), etc. We have also implemented seven segment using pushbutton and arduino. Not only that we also made circuit for both to count increment and decrement of pushing the button. We also learnt about concepts of PWM(pulse width Modulation) and about serial monitors. We took help from arduino tutorials and many links which we have listed below.

CHALLENGES FACED

Because of prior knowledge about Arduino, we have not faced much problems. But some of the concepts skipped from mind because of not working on arduino since last four months. Now, we refreshed the whole concept and not found much diffculties during programming. But we took little help from webpages to see the functions and we finally implemented the whole experiment successfully.

REFERENCES

1. https://en.wikipedia.org/wiki/Arduino
2. https://www.arduino.cc/en/Tutorial/PWM
3. https://www.arduino.cc/en/Tutorial/AnalogInputPins
4. https://www.arduino.cc/en/Tutorial/ReadAnalogVoltage
5. https://www.arduino.cc/en/Tutorial/AnalogReadSerial
6. https://eelabs.wordpress.com/2014/04/14/schematic-and-inputoutput-pins-explain-for-arduino-uno-r3/
7. https://www.allaboutcircuits.com/technical-articles/understanding-arduino-uno-hardware-design/