EES 1031 Electronics and Embedded Systems Lab

Academic Year 2024/2025

Practical 06
Pulse Width Modulation (PWM)

Name: GLNN MANJUSRI
Index No: AS20240927

PART A

Activity A-01

Explain how you can produce analog voltages between 0 V to 5V using a potentiometer.
A potentiometer has three pins. When a voltage is supplied from the two end pins, the midde pin (wiper) acts as a voltage divider by changing the resistance length with a rotating contact.

Activity A-02

Explain how you can vary the light intensity of an LED or the speed of a DC motor using a potentiometer.
By connecting the compent to ground through the wiper pin, the voltage supplied to it can be varied from 0v to 5v.

Activity A-03

What is pulse-width modulation (PWM) and how does it work?
PWM works by simulating a lower voltage than supplied by sending a series of pulses. the width of pulse correlates to different voltages. PWM is used to get an analog in-between value from from a binary signal.

Activity A-04

Derive the Root Mean Square (RMS) voltage for a pulse signal.

Activity A-05

What pins can we use to generate PWM signals in the Arduino Uno, and what are the initialized frequency/ period of the PWM signal for each pin?
Pins 3,5,6,9,10 and 11 supports PWM on the arduino uno. They operate at roughly 490Hz, except for pins 5 and 6 which operates at 980Hz.

Activity A-06

Explain how you use the Arduino function "analogWrite()" to generate PWM signals with duty cycles of 0,10,25,50,75,100 %, and calculate the theoretical Vrms for each case (using PWM 3 pin).
analogWrite accepts values from 0 to 255. So by using the formula,
$value = 255 * dutycycle$
the desired duty cycle is generated.

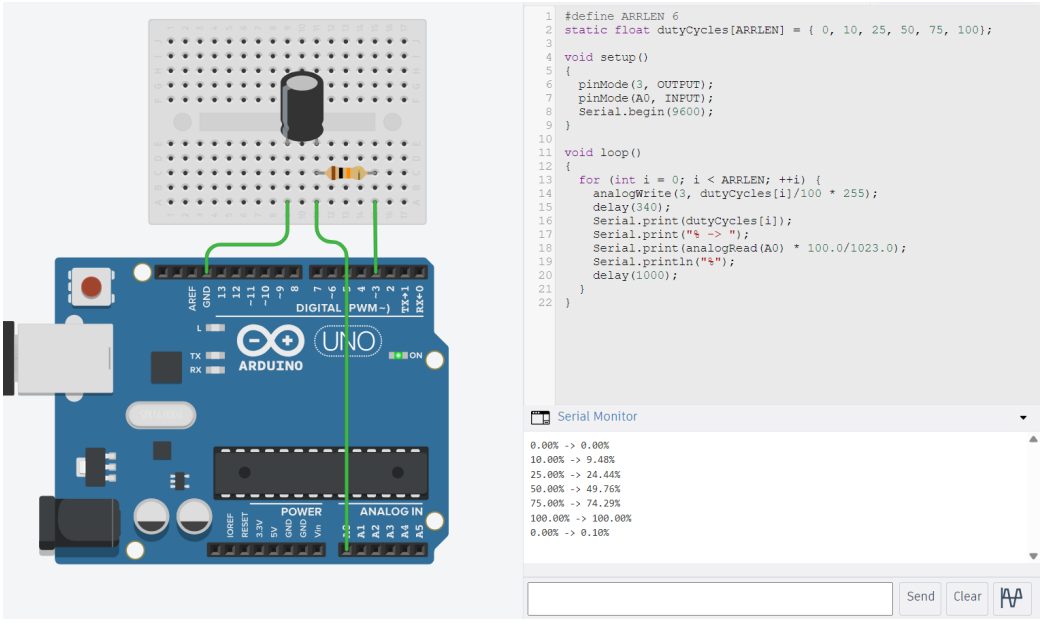
$$ex : analogWrite(3, 0.5 * 255)$$

PART B

Activity B-01

Write an Arduino sketch to generate the above PWM signals and measure the Vrms for each case.

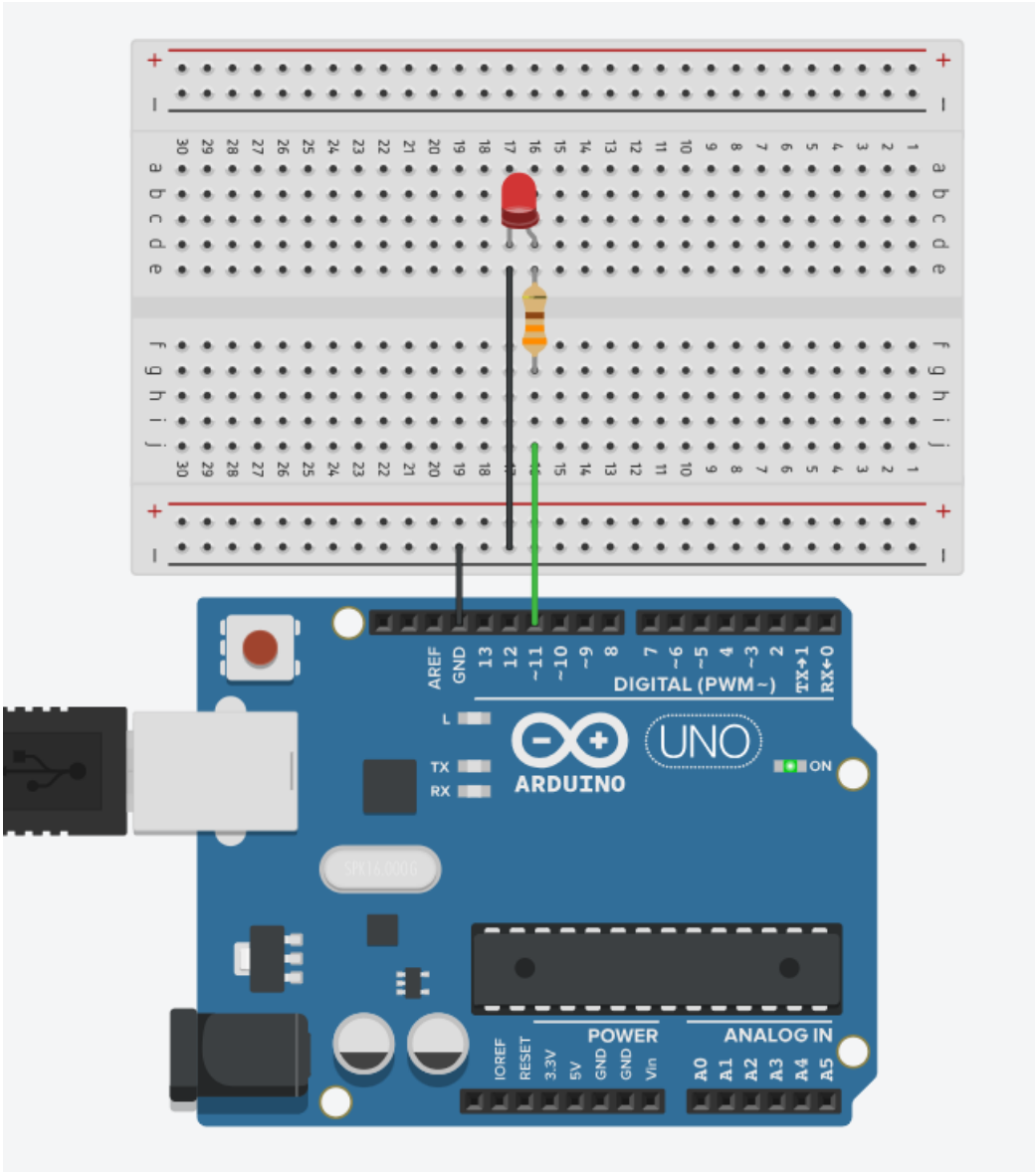
```
#define ARRLEN 6
static float dutyCycles[ARRLEN] = { 0, 10, 25, 50, 75,
   100};

void setup()
{
  pinMode(3, OUTPUT);
  pinMode(A0, INPUT);
  Serial.begin(9600);
}

void loop()
{
  for (int i = 0; i < ARRLEN; ++i) {
    analogWrite(3, dutyCycles[i]/100 * 255);
    delay(340);
    Serial.print(dutyCycles[i]);
    Serial.print("% -> ");
    Serial.print(analogRead(A0) * 100.0/1023.0);
    Serial.println("%");
    delay(1000);
  }
}
```



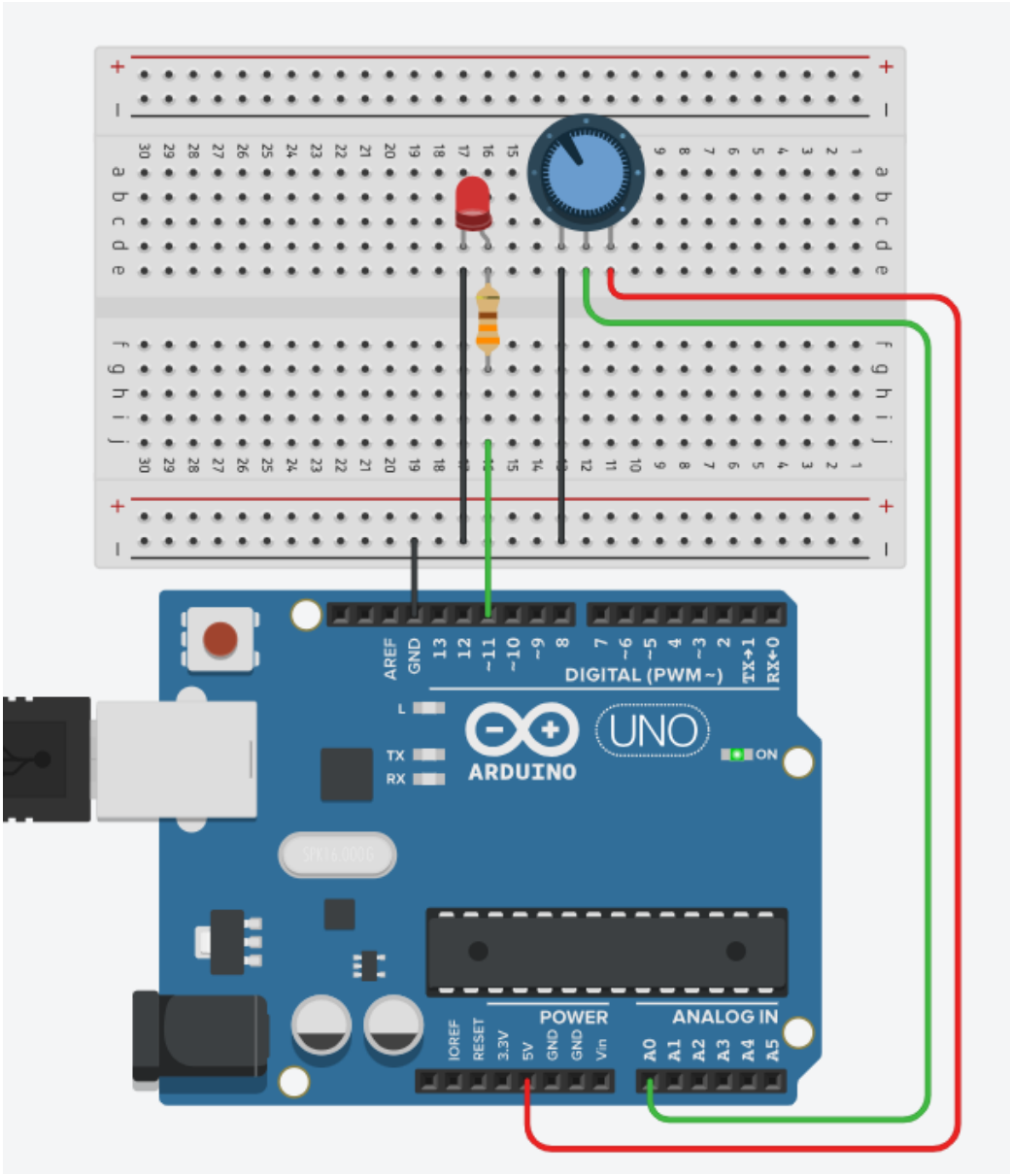
Activity B-02

Write an Arduino sketch to control the light intensity of an LED using the PWM signal.

```
#define PIN 11
#define DT 10

void setup() {
  pinMode(PIN, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  for (float i = 0.0f; i < 256; ++i) {
    analogWrite(PIN, 255 * sin(i / 255 * M_PI));
    delay(DT);
  }
}
```



Then modify the code to vary the light intensity (PWM signal duty cycle) via an analog input coming from the potentiometer.

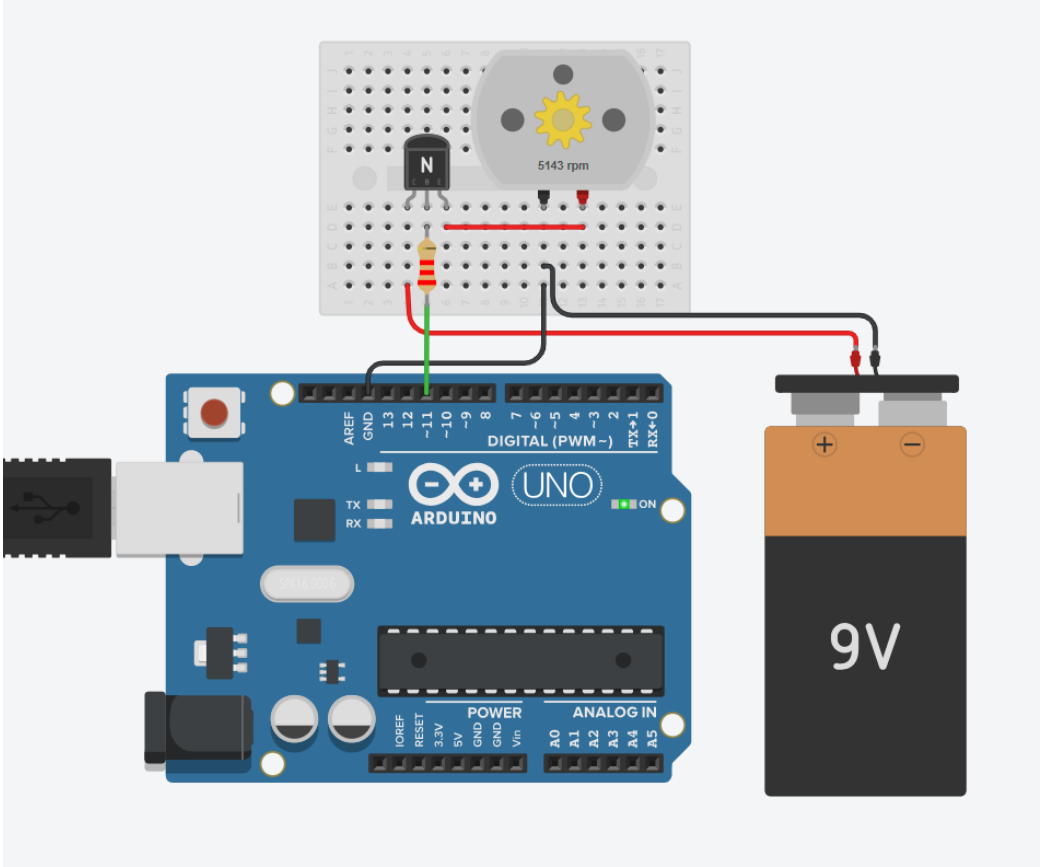



```
#define IN  A0
#define OUT 11

void setup() {
  pinMode(OUT, OUTPUT);
  pinMode(IN, INPUT);
}

void loop() {
```

```
  analogWrite(OUT, analogRead(IN) >> 2); // OUT =
     (2^8/2^10)*IN
}
```

Activity B-03

Construct a circuit to control the speed of a DC motor using a PWM signal and explain how the circuit works.



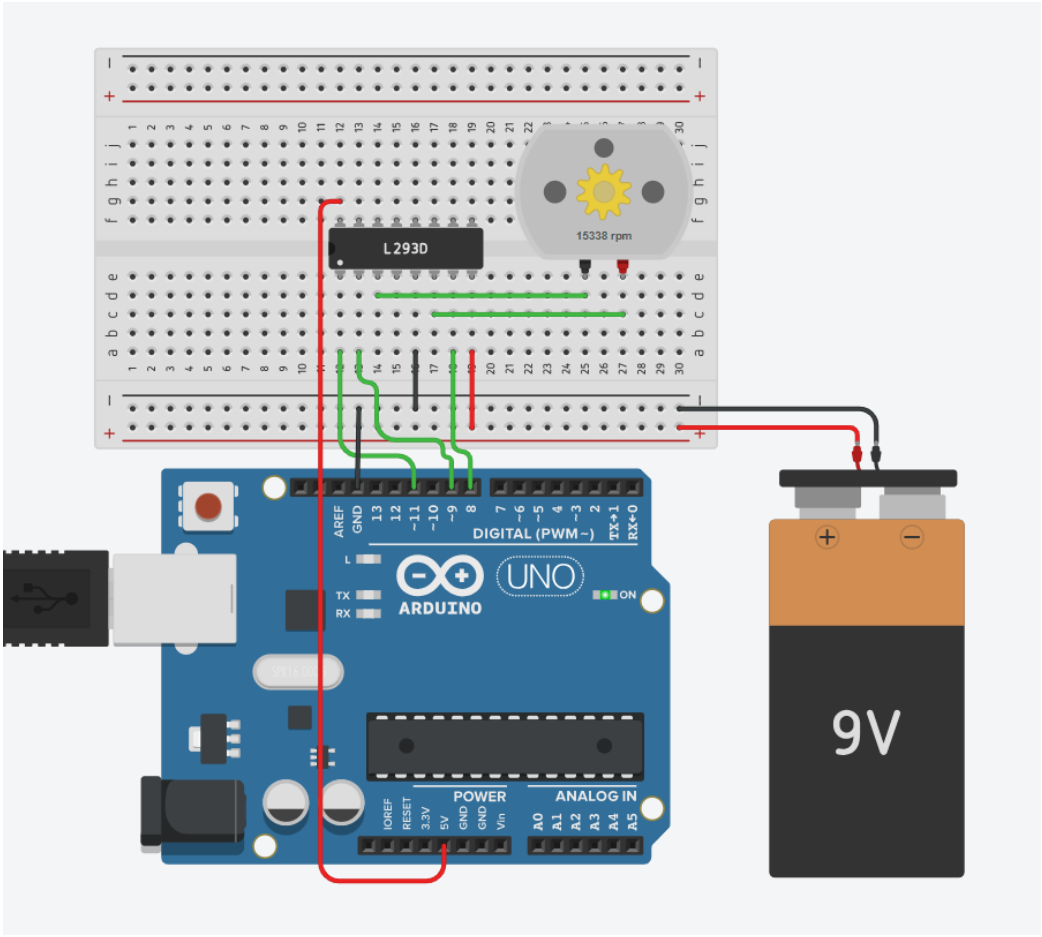

```
#define EN 11
#define DT 10
#define INTERVAL 1000

void setup() {
  pinMode(EN, OUTPUT);
}

void loop() {
```

```
10    for (int i = 1; i < 256; ++i) {
11      analogWrite(EN, i);
12      delay(DT);
13    }
14
15    for (int i = 254; i >= 0; --i) {
16      analogWrite(EN, i);
17      delay(DT);
18    }
19
20    delay(INTERVAL);
21  }
```

Do not provide the power to the motor directly from the microcontroller. Why?

Activity B-04

Now, construct a circuit to control the DC motor speed and the direction (H bridge). Explain how the circuit works. Write an Arduino sketch to demonstrate the DC motor controller.




```
#define EN 11
#define CLKWS  8
#define ACLKWS 9
#define DT 10
#define INTERVAL 1000

void setup() {
  pinMode(EN,      OUTPUT);
  pinMode(CLKWS,   OUTPUT);
  pinMode(ACLKWS,  OUTPUT);
}

void loop() {
  digitalWrite(CLKWS,   HIGH);
  digitalWrite(ACLKWS,  LOW);

  for (int i = 1; i < 256; ++i) {
```

```
    analogWrite(EN, i);
    delay(DT);
  }
  for (int i = 254; i >= 0; --i) {
    analogWrite(EN, i);
    delay(DT);
  }
  delay(INTERVAL);

  digitalWrite(CLKWS,  LOW);
  digitalWrite(ACLKWS, HIGH);

  for (int i = 1; i < 256; ++i) {
    analogWrite(EN, i);
    delay(DT);
  }
  for (int i = 254; i >= 0; --i) {
    analogWrite(EN, i);
    delay(DT);
  }
  delay(INTERVAL);
}
```