Project :

Rain Rain go away

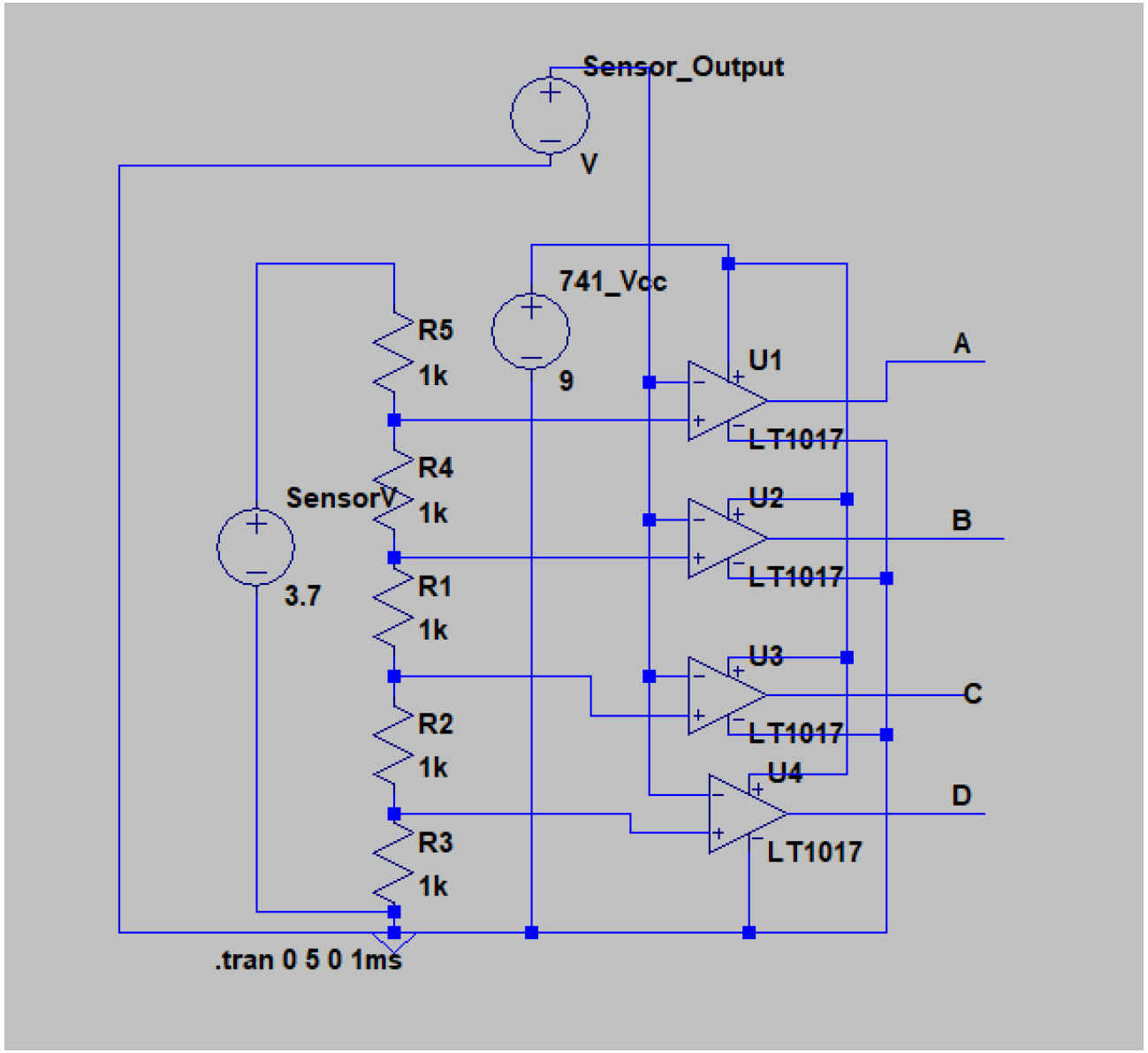
AIM : Design an automated rain detection and wiper control mechanism that activates wiper rotation (0-180 degrees) on detecting rain and displays four discrete controllable speed levels.

https://drive.google.com/drive/folders/1tUdHJNRUtfXS56_pBJ250gK1kQLBFwH8?usp=sharing

(link to some videos of practical implimentation)

SI	COMPONENT
1.	Resistive rain sensor
2.	Resistors
3.	Op amps
4.	Power supply
5.	LEDs
6.	Motor Driver
7.	DC Motor
8.	Arduino UNO
9.	Limiting switches

10.	Servo motor
11.	Jumper wires
12.	Bread boards
13.	Capacitors

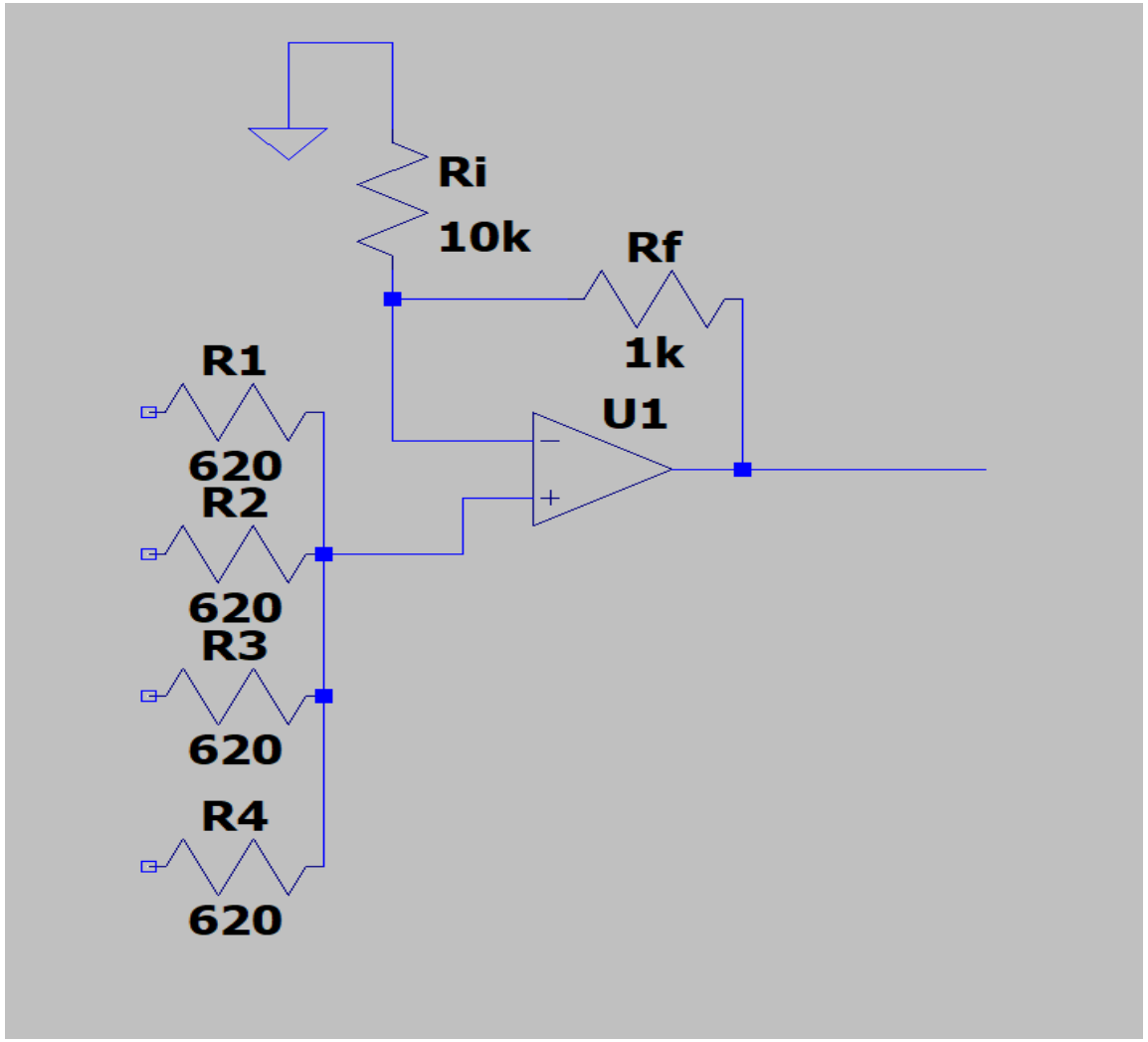


Comparator Network - Schematic

This circuit takes in the Analog output of the sensor and feeds it into the inverting input of Op-amps. These Op-amps are each given a reference voltage from a voltage divider bridge that ranges from 3.7 V to ground (This is the rage of the sensor's analog output). When the sensor output drops below a certain level, the respective Op-amp will output a high voltage.

- The outputs of the network (namely, A, B, C and D) are connected to a digital to analog converter circuit. This consists of a non-inverting adder from an op-amp. This will give out different DC signals for different combinations. Such as:
  - 0-0-0-0 2v (no LED glows, no water)
  - 1-0-0-0 around 3.5v (1 LED glows, first speed level)
  - 1-1-0-0 around 4.5v (2 LEDs glow, second speed level)
  - 1-1-1-0 around 5.5 v (3 LEDs glow, third speed level)
  - 1-1-1-1 around 6.5 v (4 LEDs glow, fourth speed level)

- The outputs of the network are connected to the LED dashboard that enables required number of LEDs to glow according to the amount of water sensed.



Non inverting summing amplifier - schematic

GENERATING 4 ANALOG VALUES

In the provided circuit all the values of R1,R2,R3,R4 are of same value R.

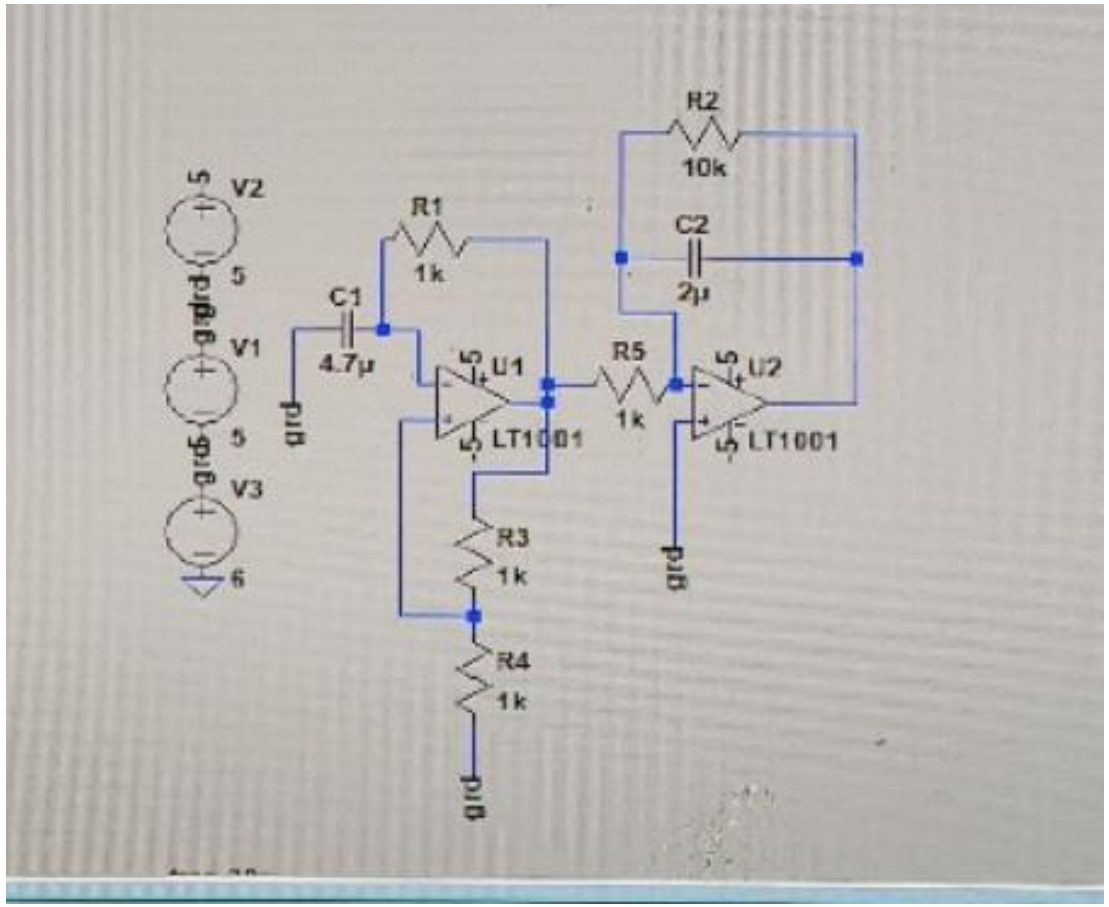
R = 620 ohm

Four Digital inputs are given at V1,V2,V3,V4.

Op amp as an amplifier formula : $V_{out} = V_{in}(1 + R_f/R_i)$

CASE 1	CASE 2	CASE 3	CASE 4
Digital input :1000 $V_1 = V$(high) {V_2, V_3, V_4 are (low)} $V_{in} = (V(R/3)) / (4R/3)$ $= V/4$ $V_{out} = (V_1/4)(1+(R_f/R_i))$ $=(V_1/4)(1.1)$	Digital input : 1100 $V_1 = V_2$ (high) = V {V_3, V_4 are (low)} $V_{in} = (V)(R/2)/R$ $= V/2$ $V_{out} = (V/2)(1.1)$	Digital input : 1100 $V_1 = V_2 = V_3$ (high) = V {V_4 is (low)} $V_{in} = (V)(R)/(4R/3)$ $= 3V/4$ $V_{out} = (3V/4)(1.1)$	Digital input : 1100 $V_1 = V_2 = V_3 = V_4$(high) = V $V_{in} = (V)$ $V_{out} = (V)(1.1)$

These 4 discrete analog values are fed to the PWM circuit.

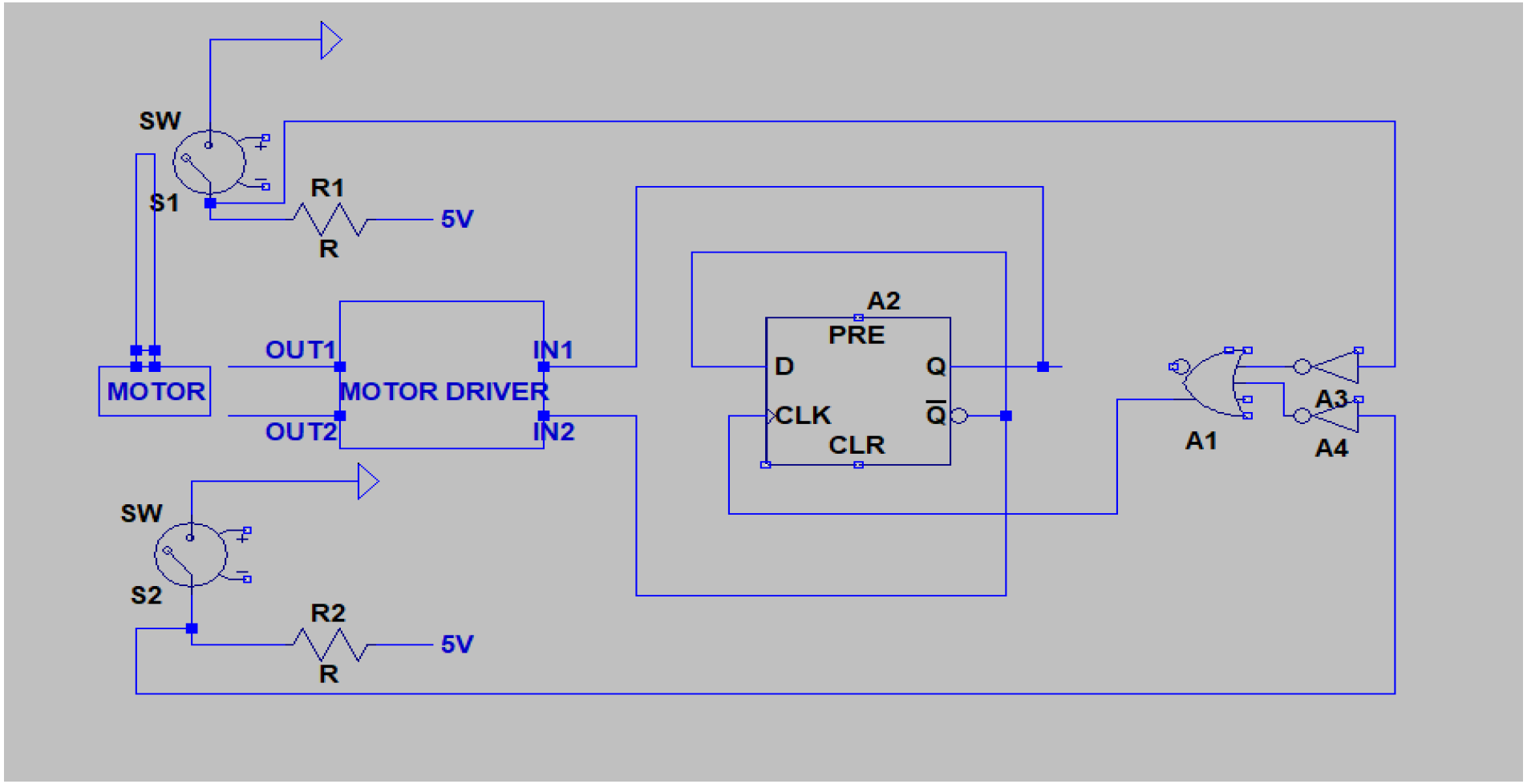


Triangle wave generator - schematic

This circuit consists of two parts. First part uses Opamp ckt to form a square wave signal oscillating with a frequency of about 100 Hz. The sencond part of the circuit forms the integrator using an Opamp that uses the square wave from the first part to give a triangle wave.
The LTspice simulation of the circuit is attached below.

- Further, this Triangle wave is fed into a comparator's inverting input, while the DAC's voltage is fed into the comparator's non-inverting input.
- The resulting output of the comparator is a PWM signal whose duty cycle increases with increased output voltage from the DAC.
- Higher duty cycle causes the motor to rotate at a faster speed. Thus the speed control of the motor through the sensor's output is achieved.

Switching circuit – schematic



- To make the wiper rotate smoothly from 0 to 180 degrees we use the following circuit. The PWM signal is connected to the enable pin of the motor driver used.
- The motor driver also includes two input pins, namely, IN1 and IN2. These pins control the direction of rotation of the motor.
- We use two limiting switches to toggle the supply to IN1 and IN2 to achieve proper wiper rotation.

BONUS PART : Needle pointing to speed mechanism

We have used arduino UNO to control a servo motor that serves as the needle of speed indicator.

CODE :

```
#include <Servo.h>

Servo myservo; // Create a servo object

// Define the input pins for the circuit
const int input1 = 2;
const int input2 = 3;
const int input3 = 4;
const int input4 = 5;

// Define the servo angles for specific inputs
const int angle_0000 = 180;   // For input 0000
const int angle_0001 = 150;  // For input 0001
const int angle_0011 = 110;  // For input 0011
const int angle_0111 = 70; // For input 0111
const int angle_1111 = 30; // For input 1111
```

```
void setup() {
  myservo.attach(9); // Attach the servo to pin 9


// Set input pins as inputs
  pinMode(input1, INPUT);
  pinMode(input2, INPUT);
  pinMode(input3, INPUT);
  pinMode(input4, INPUT);

// Initialize servo to 0 degrees
  myservo.write(angle_0000);
}
void loop() {
  // Read the inputs
  int bit1 = digitalRead(input1);
  int bit2 = digitalRead(input2);
  int bit3 = digitalRead(input3);
  int bit4 = digitalRead(input4);

  // Determine the binary value from the inputs
  int binaryValue = (bit4 << 3) | (bit3 << 2) | (bit2 << 1) | bit1;
}
```

```
  // Set the servo angle based on the binary value
  switch (binaryValue) {
    case 0b0000:
      myservo.write(angle_0000);
      break;
    case 0b0001:
      myservo.write(angle_0001);
      break;
    case 0b0011:
      myservo.write(angle_0011);
      break;
case 0b0111:
      myservo.write(angle_0111);
      break;
    case 0b1111:
      myservo.write(angle_1111);
      break;
    default:
      // Optional: handle invalid inputs (do nothing)
      break;
  }
  // Delay for 500 ms
  delay(500);  }
```

BONUS PART : Needle pointing to speed mechanism

Explanation:

- **Purpose:** Controls a servo motor based on 4 digital input pins.
- **Setup (`setup()`)**
  - Attach **servo** to pin 9.
  - Set **pins 2, 3, 4, 5 as inputs**.
  - Initialize servo at **0°**.
- **Loop (`loop()`)**
  - Read **4 input pins**, form a **4-bit binary value**.
  - Use `switch-case` to set the **servo angle** for specific values.
  - **Wait 500ms**, repeat.
- **Servo Angles Mapping:**
  - `0000` → **180°**, `0001` → **150°**, `0011` → **110°**, `0111` → **70°**, `1111` → **0°**.
  - Other inputs do nothing.

Overall circuit – block diagram