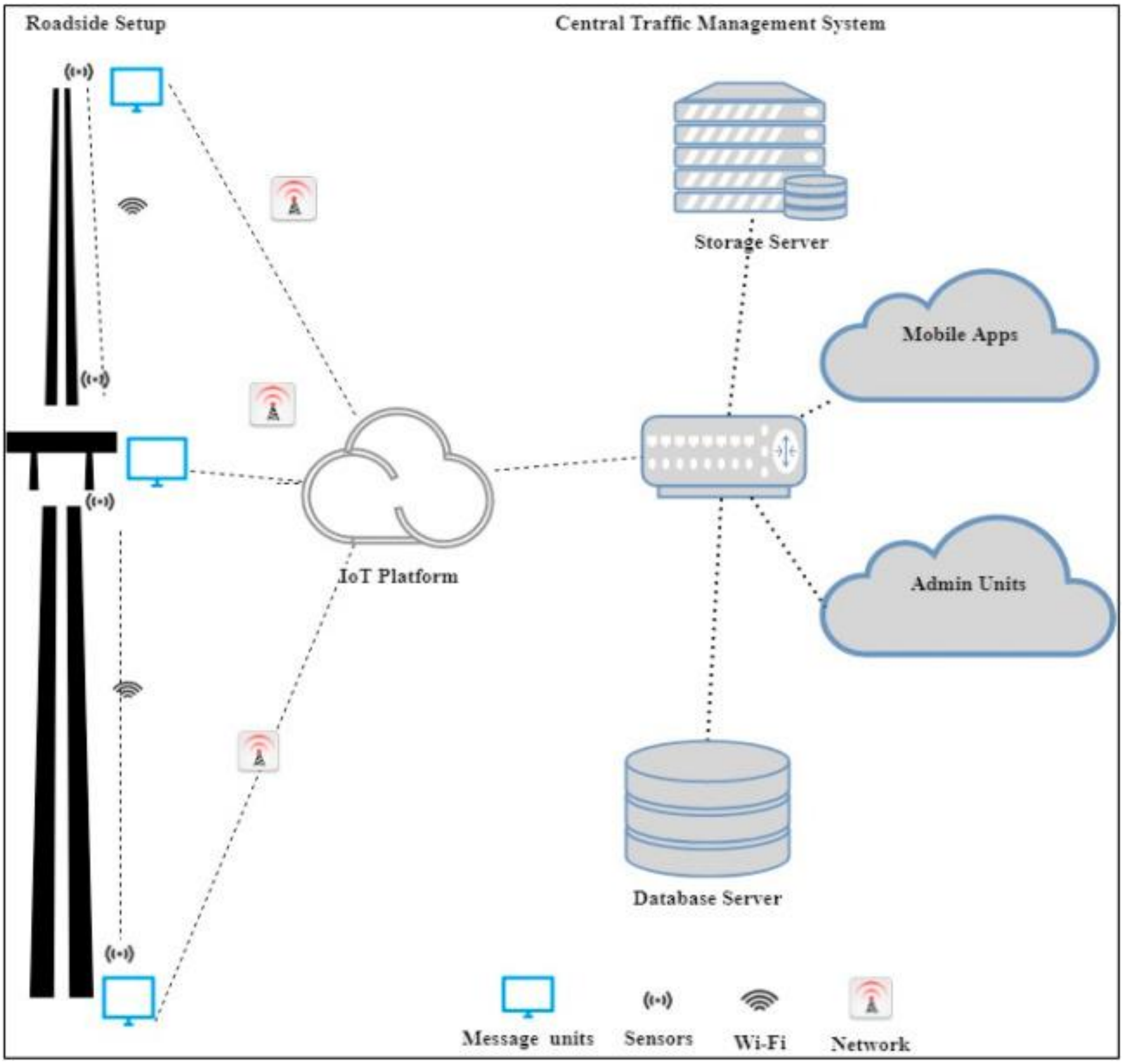
PROJECT DEVELOPMENT PART 1

TRAFFIC MANAGEMENT



TEAM DETAILS

MENTOR	Mrs. M. Maheswari
LEADER	M. Keerthika
MEMBERS	M. Abinaya R. Deepa K. Divya R. Mahalakshmi
PROBLEM DESCRIPTION	We will start to build the IOT simulation for traffic management

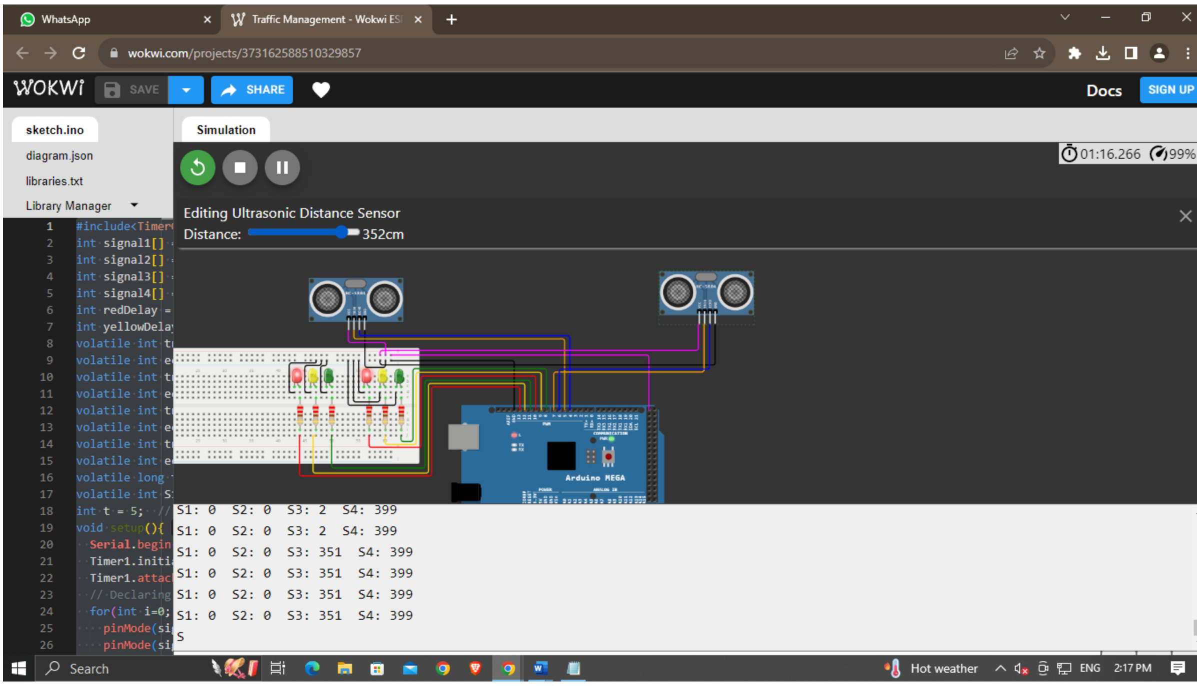
SIMULATION STEPS

STEP 1: Access Wokwi.

- Go to the websites(https://wokwi.com/).

STEP 2: Create a new project.

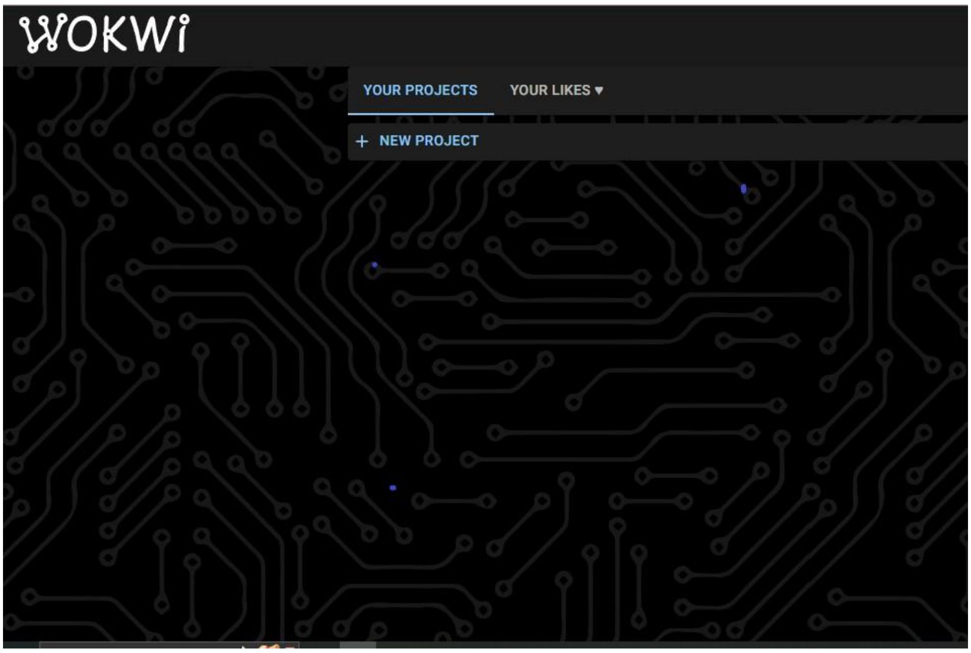
- Click on “Create a new project.”



Editing Ultrasonic Distance Sensor
Distance: 352cm

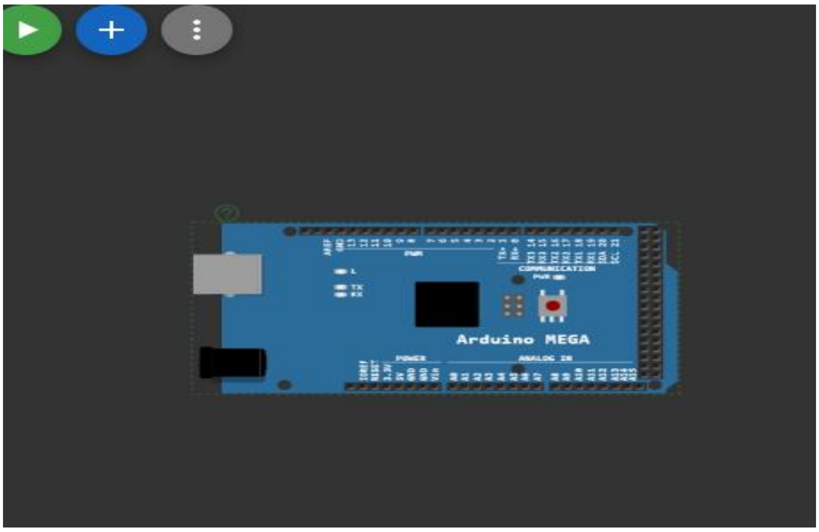
```
#include<Timer
int signal1[]
int signal2[]
int signal3[]
int signal4[]
int redDelay =
int yellowDela
volatile int t
volatile int e
volatile int t
volatile int e
volatile int t
volatile int e
volatile int t
volatile int e
volatile long
volatile int S
int t = 5;  //
void setup(){
  Serial.begin
  Timer1.initi
  Timer1.attac
  // Declaring
  for(int i=0;
    pinMode(si
    pinMode(si
```

```
S1: 0  S2: 0  S3: 2  S4: 399
S1: 0  S2: 0  S3: 2  S4: 399
S1: 0  S2: 0  S3: 351  S4: 399
S1: 0  S2: 0  S3: 351  S4: 399
S1: 0  S2: 0  S3: 351  S4: 399
S1: 0  S2: 0  S3: 351  S4: 399
S
```



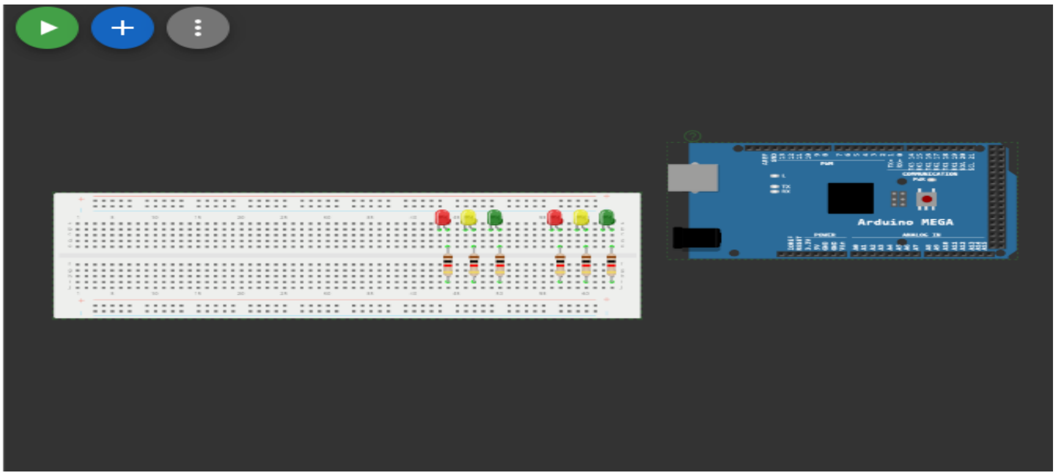
STEP 3: Add component

- In the Components panel, search for a "Arduino MEGA" and drag it onto the virtual breadboard.



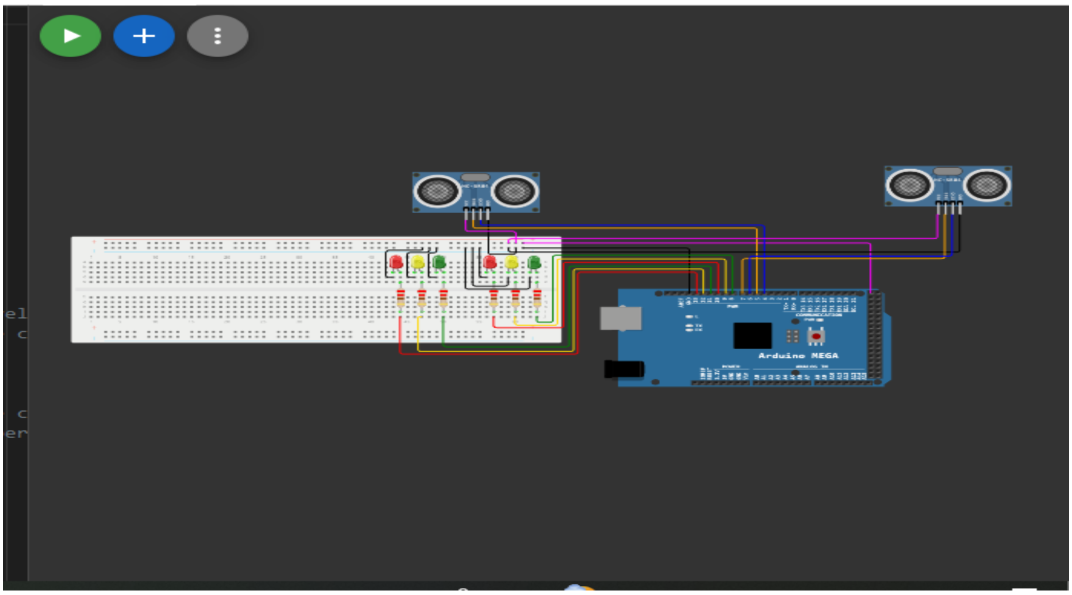
STEP 4: Add an LED and Resistor

- Find and drag an LED and a resistor onto the virtual breadboard



STEP 5: Add a ultrasonic components.

- Find and drag an ultrasonic components onto the virtual breadboard.



Simulation Connection:

- LED1: C Connected in the breadboard at terminal strips(5 hole rows)
- The resistor r1:1 is connected to r2:1below the LED 1:C
- LED2: C Connected in the breadboard at terminal strips(5 hole rows)
- The resistor r2:1 is connected to r2:2below the LED 2:C
- LED3: C Connected in the breadboard at terminal strips(5 hole rows)

- The resistor r3:1 is connected to r3:2below the LED3:C
- LED4: C Connected in the breadboard at

terminal strips(5 hole rows)

- The resistor r4:1 is connected to r4:2below the LED4:C
- LED5: C Connected in the breadboard at terminal strips(5 hole rows)
- The resistor r5:1 is connected to r5:2below the LED5:C
- LED6: C Connected in the breadboard at terminal strips(5 hole rows)
- The resistor r6:1 is connected to r6:2below the LED6:C
- The Arduino mega 10 connected to the resistor r1:2
- The Arduino mega 9 connected to the resistor r2:2
- The Arduino mega 8 connected to the resistor r3:2
- The Arduino mega 13 connected to the resistor r4:2
- The Arduino mega 12 connected to the resistor r5:2
- The Arduino mega 11 connected to the resistor r6:2
- Ultrasonic1:VCC is connected to the breadboard at power rails.

- Ultrasonic1:TRIG is connected to the Arduino mega 5.
- Ultrasonic1ECHO is connected to the Arduino mega 4.
- Ultrasonic1:GND is connected to the breadboard at power rails
- Ultrasonic2:VCC is connected to the breadboard at power rails.
- Ultrasonic2:TRIG is connected to the Arduino mega 7.
- Ultrasonic1ECHO is connected to the Arduino mega 6.
- Ultrasonic2:GND is connected to the breadboard at power rails

STEP 6: CODE

```
#include<TimerOne.h>
```

```
int signal1[] = {23, 25, 27};
int signal2[] = {46, 48, 50};
int signal3[] = {13, 12, 11};
int signal4[] = {10, 9, 8};
int redDelay = 5000;
int yellowDelay = 2000;
volatile int triggerpin1 = 31;
volatile int echopin1 = 29;
volatile int triggerpin2 = 44;
volatile int echopin2 = 42;
volatile int triggerpin3 = 7;
volatile int echopin3 = 6;
volatile int triggerpin4 = 5;
volatile int echopin4 = 4;
volatile long time;                    // Variable for storing the time traveled
volatile int S1, S2, S3, S4;           // Variables for storing the distance covered
int t = 5;  // distance under which it will look for vehicles.
void setup(){
  Serial.begin(115200);
  Timer1.initialize(100000);  //Begin using the timer. This function must be called first. "microseconds" is the period of time the timer takes.
  Timer1.attachInterrupt(softInterr); //Run a function each time the timer period finishes.
  // Declaring LED pins as output
  for(int i=0; i<3; i++){
    pinMode(signal1[i], OUTPUT);
    pinMode(signal2[i], OUTPUT);
    pinMode(signal3[i], OUTPUT);
    pinMode(signal4[i], OUTPUT);
  }
  // Declaring ultrasonic sensor pins as output
  pinMode(triggerpin1, OUTPUT);
  pinMode(echopin1, INPUT);
  pinMode(triggerpin2, OUTPUT);
  pinMode(echopin2, INPUT);
  pinMode(triggerpin3, OUTPUT);
  pinMode(echopin3, INPUT);
  pinMode(triggerpin4, OUTPUT);
  pinMode(echopin4, INPUT);
}
void loop()
{
  // If there are vehicles at signal 1
  if(S1<t)
  {
    signal1Function();
  }
  // If there are vehicles at signal 2
  if(S2<t)
  {
    signal2Function();
  }
  // If there are vehicles at signal 3
  if(S3<t)
  {
    signal3Function();
  }
  // If there are vehicles at signal 4
  if(S4<t)
  {
    signal4Function();
  }
}
// This is interrupt function and it will run each time the timer period finishes. The timer period is set at 100 milli seconds.
void softInterr()
{
  // Reading from first ultrasonic sensor
  digitalWrite(triggerpin1, LOW);
  delayMicroseconds(2);
  digitalWrite(triggerpin1, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerpin1, LOW);
  time = pulseIn(echopin1, HIGH);
  S1= time*0.034/2;
  // Reading from second ultrasonic sensor
  digitalWrite(triggerpin2, LOW);
  delayMicroseconds(2);
  digitalWrite(triggerpin2, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerpin2, LOW);
  time = pulseIn(echopin2, HIGH);
  S2= time*0.034/2;
  // Reading from third ultrasonic sensor
  digitalWrite(triggerpin3, LOW);
  delayMicroseconds(2);
  digitalWrite(triggerpin3, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerpin3, LOW);
  time = pulseIn(echopin3, HIGH);
  S3= time*0.034/2;
  // Reading from fourth ultrasonic sensor
  digitalWrite(triggerpin4, LOW);
  delayMicroseconds(2);
  digitalWrite(triggerpin4, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerpin4, LOW);
  time = pulseIn(echopin4, HIGH);
  S4= time*0.034/2;
  // Print distance values on serial monitor for debugging
  Serial.print("S1: ");
  Serial.print(S1);
  Serial.print("  S2: ");
  Serial.print(S2);
  Serial.print("  S3: ");
  Serial.print(S3);
  Serial.print("  S4: ");
  Serial.println(S4);
}
void signal1Function()
```

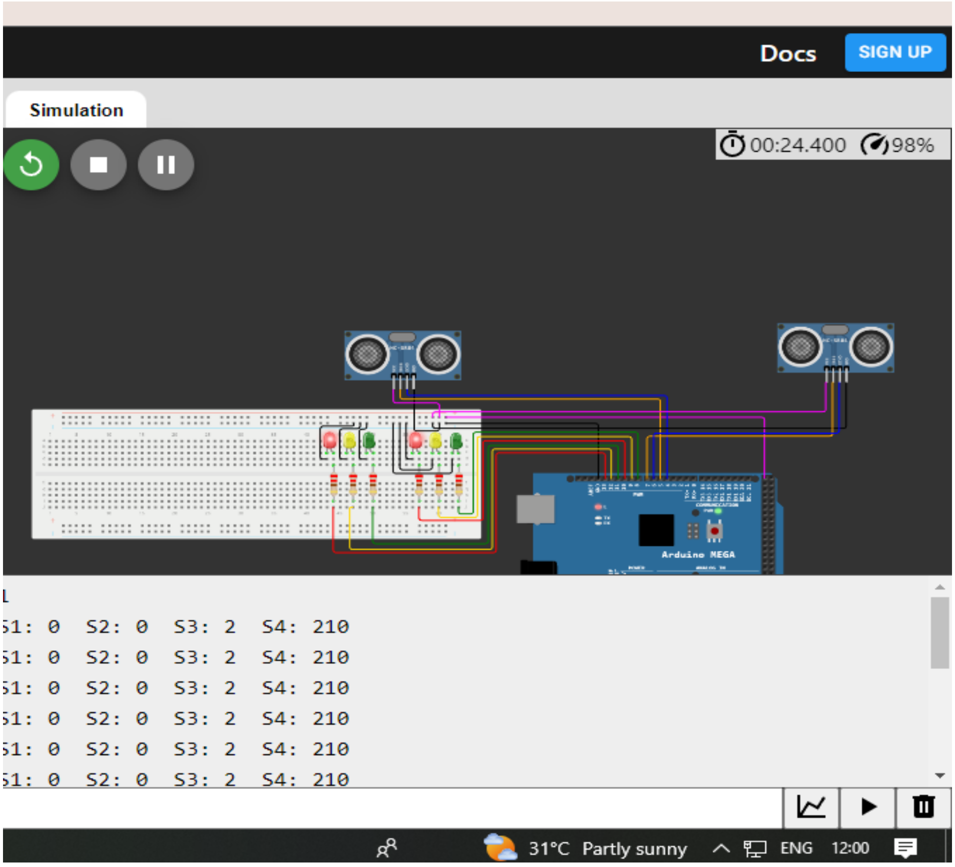
```
{
  Serial.println("1");
  low();
  // Make RED LED LOW and make Green HIGH for 5 seconds
  digitalWrite(signal1[0], LOW);
  digitalWrite(signal1[2], HIGH);
  delay(redDelay);
  // if there are vehicels at other signals
  if(S2<t || S3<t || S4<t)
  {
    // Make Green LED LOW and make yellow LED HIGH for 2 seconds
    digitalWrite(signal1[2], LOW);
    digitalWrite(signal1[1], HIGH);
    delay(yellowDelay);
  }
}
void signal2Function()
{
  Serial.println("2");
  low();
  digitalWrite(signal2[0], LOW);
  digitalWrite(signal2[2], HIGH);
  delay(redDelay);

  if(S1<t || S3<t || S4<t)
  {
    digitalWrite(signal2[2], LOW);
    digitalWrite(signal2[1], HIGH);
    delay(yellowDelay);
  }
}
void signal3Function()
{
  Serial.println("3");
  low();
  digitalWrite(signal3[0], LOW);
  digitalWrite(signal3[2], HIGH);
  delay(redDelay);
  if(S1<t || S2<t || S4<t)
  {
    digitalWrite(signal3[2], LOW);
    digitalWrite(signal3[1], HIGH);
    delay(yellowDelay);
  }
}
void signal4Function()
{
  Serial.println("4");
  low();
  digitalWrite(signal4[0], LOW);
  digitalWrite(signal4[2], HIGH);
  delay(redDelay);
  if(S1<t || S2<t || S3<t)
  {
    digitalWrite(signal4[2], LOW);
    digitalWrite(signal4[1], HIGH);
    delay(yellowDelay);
  }
}
// Function to make all LED's LOW except RED one's.
void low()
{
  for(int i=1; i<3; i++)
  {
    digitalWrite(signal1[i], LOW);
    digitalWrite(signal2[i], LOW);
    digitalWrite(signal3[i], LOW);
    digitalWrite(signal4[i], LOW);
  }
  for(int i=0; i<1; i++)
  {
    digitalWrite(signal1[i], HIGH);
    digitalWrite(signal2[i], HIGH);
    digitalWrite(signal3[i], HIGH);
    digitalWrite(signal4[i], HIGH);
  }
}
```

STEP 7: Simulation

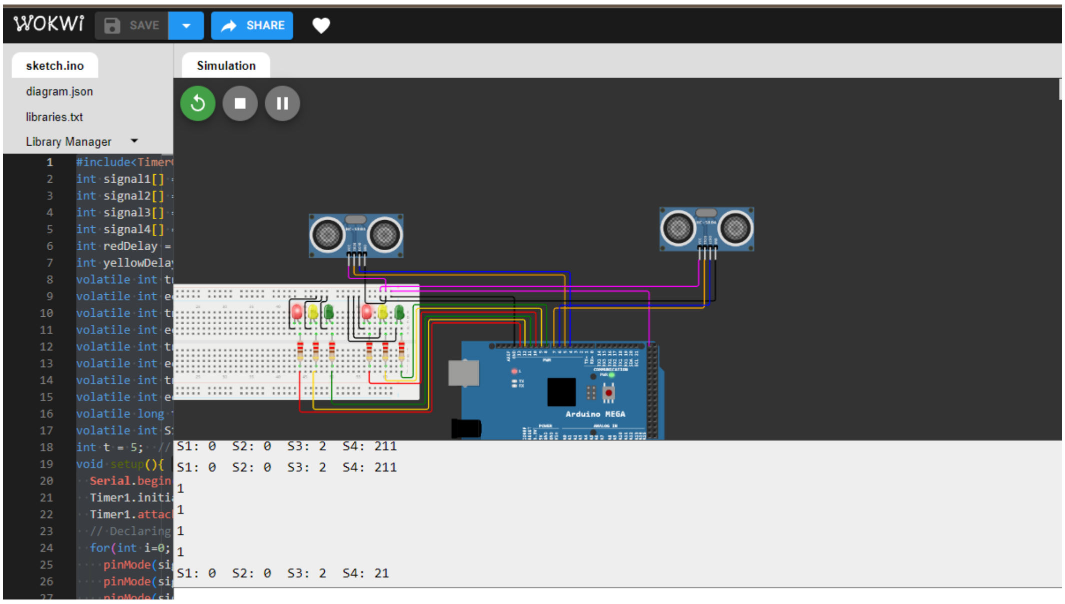
- Click the "Simulate" button to start the

Simulation

- The LED should start blinking according to the code

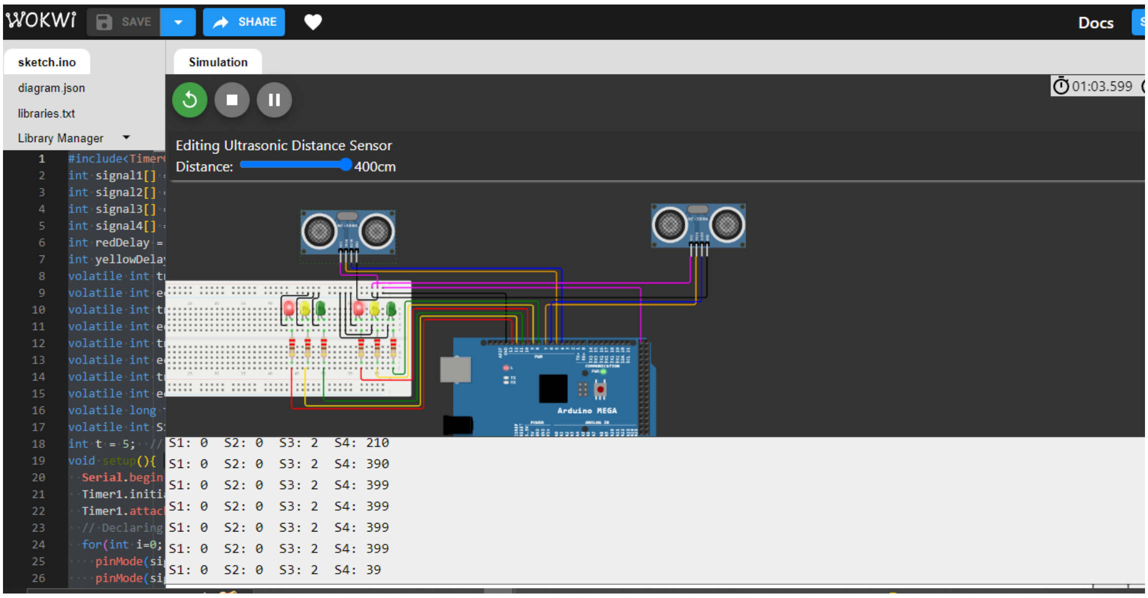


```
L
S1: 0  S2: 0  S3: 2  S4: 210
S1: 0  S2: 0  S3: 2  S4: 210
S1: 0  S2: 0  S3: 2  S4: 210
S1: 0  S2: 0  S3: 2  S4: 210
S1: 0  S2: 0  S3: 2  S4: 210
S1: 0  S2: 0  S3: 2  S4: 210
```



```
#include<Timer
int signal1[]
int signal2[]
int signal3[]
int signal4[]
int redDelay =
int yellowDelay
volatile int t
volatile int e
volatile int t
volatile int e
volatile int t
volatile int e
volatile int t
volatile int e
volatile long
volatile int S
int t = 5;  //
void setup(){
  Serial.begin
  Timer1.initi
  Timer1.attac
  // Declaring
  for(int i=0;
    pinMode(si
    pinMode(si
```

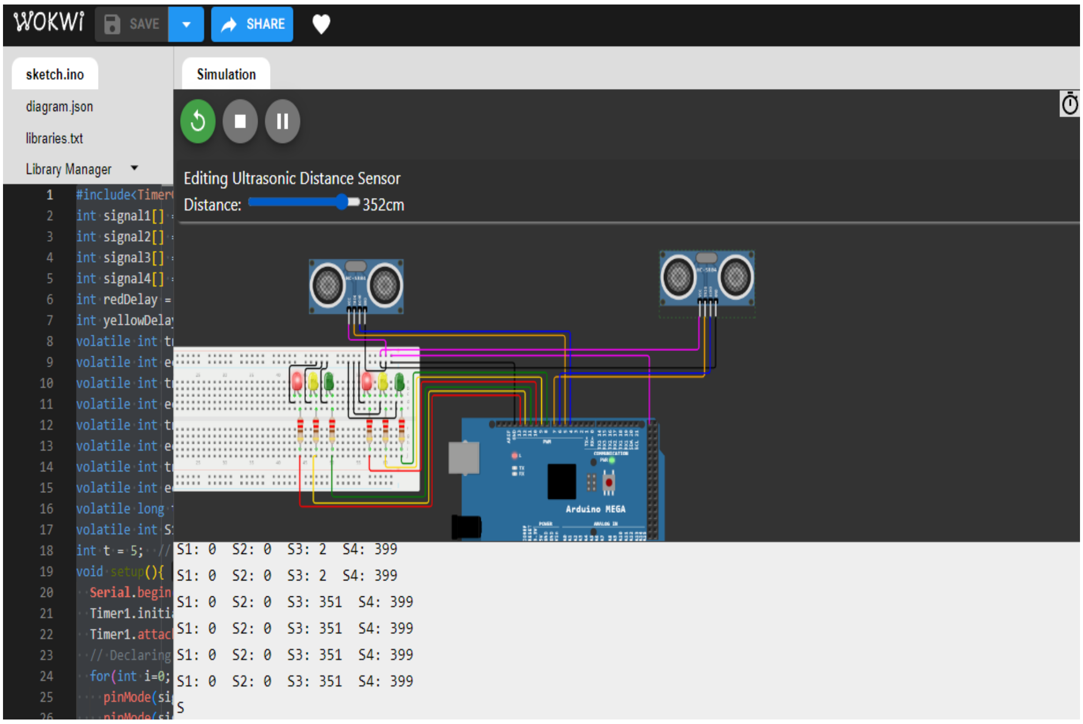
```
S1: 0   S2: 0   S3: 2   S4: 211
S1: 0   S2: 0   S3: 2   S4: 211
1
1
1
1
S1: 0   S2: 0   S3: 2   S4: 21
```



Editing Ultrasonic Distance Sensor
Distance: 400cm

```
#include<Timer
int signal1[]
int signal2[]
int signal3[]
int signal4[]
int redDelay =
int yellowDelay
volatile int t
volatile int e
volatile int t
volatile int e
volatile int t
volatile int e
volatile int t
volatile int e
volatile long
volatile int S
int t = 5;  //
void setup(){
  Serial.begin
  Timer1.initi
  Timer1.attac
  // Declaring
  for(int i=0;
    pinMode(si
    pinMode(si
```

```
S1: 0   S2: 0   S3: 2   S4: 210
S1: 0   S2: 0   S3: 2   S4: 390
S1: 0   S2: 0   S3: 2   S4: 399
S1: 0   S2: 0   S3: 2   S4: 399
S1: 0   S2: 0   S3: 2   S4: 399
S1: 0   S2: 0   S3: 2   S4: 399
S1: 0   S2: 0   S3: 2   S4: 39
```



Editing Ultrasonic Distance Sensor
Distance: 352cm

```
#include<Timer
int signal1[]
int signal2[]
int signal3[]
int signal4[]
int redDelay =
int yellowDela
volatile int t
volatile int e
volatile int t
volatile int e
volatile int t
volatile int e
volatile int t
volatile int e
volatile long
volatile int S
int t = 5;  //
void setup(){
  Serial.begin
  Timer1.initi
  Timer1.attac
  // Declaring
  for(int i=0;
    pinMode(si
    pinMode(si
```

```
S1: 0  S2: 0  S3: 2  S4: 399
S1: 0  S2: 0  S3: 2  S4: 399
S1: 0  S2: 0  S3: 351  S4: 399
S1: 0  S2: 0  S3: 351  S4: 399
S1: 0  S2: 0  S3: 351  S4: 399
S1: 0  S2: 0  S3: 351  S4: 399
S
```