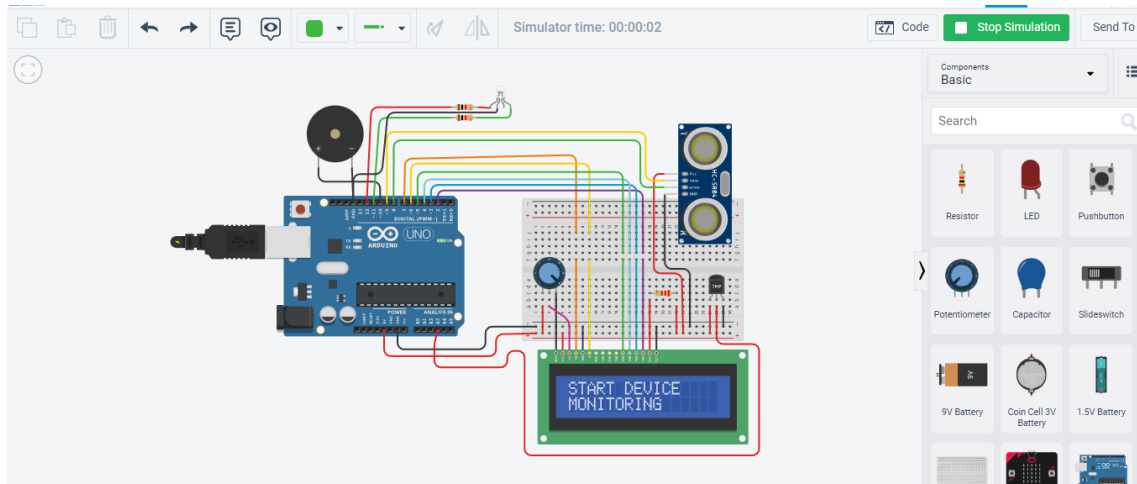


Assignment -1

Design a TinkerCad Circuit for Smart Home



PROGRAM

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(7,6, 5, 4, 3, 2);

#define echoPin 8

#define trigPin 9

long duration;

int distance;

void setup()
{
    pinMode(10,OUTPUT);
    lcd.begin(16, 2);
    lcd.setCursor(0,0);
    lcd.print("KARTHIGA");
```

```
lcd.setCursor(0,1);  
lcd.print("PROJECT 1");  
delay(1000);  
lcd.clear();  
lcd.setCursor(0,0);  
lcd.print("START DEVICE ");  
lcd.setCursor(0,1);  
lcd.print("MONITORING");  
delay(1000);  
lcd.clear();  
pinMode(echoPin,INPUT);  
pinMode(trigPin,OUTPUT);  
pinMode(10,OUTPUT);  
pinMode(11,OUTPUT);  
pinMode(12,OUTPUT);  
Serial.begin(9600);  
}
```

```
void loop()  
{  
  digitalWrite(trigPin, LOW);  
  delayMicroseconds(2);  
  // Sets the trigPin HIGH (ACTIVE) for 10 microseconds  
  digitalWrite(trigPin, HIGH);  
  delayMicroseconds(10);  
  digitalWrite(trigPin, LOW);  
  // Reads the echoPin, returns the sound wave travel time in microseconds  
  duration = pulseIn(echoPin, HIGH);  
  // Calculating the distance
```

```
distance = duration * 0.034 / 2;
```

```
int signal=analogRead(A3);
```

```
float voltage=(signal*5.0)/1024;
```

```
float temC=(voltage-0.5)*100;
```

```
lcd.setCursor(0,0);
```

```
lcd.print("Temperature:");
```

```
lcd.print(temC);//temprature monitor
```

```
Serial.print("Distance: ");
```

```
Serial.print(distance);
```

```
Serial.println(" cm");
```

```
lcd.setCursor(0,1);
```

```
lcd.print("O-distance:");
```

```
lcd.print(distance);
```

```
if (temC>100){
```

```
digitalWrite(12,HIGH);
```

```
digitalWrite(10,HIGH);
```

```
}
```

```
else{
```

```
digitalWrite(12,LOW);
```

```
digitalWrite(10,LOW);
```

```
digitalWrite(11,HIGH);
```

```
}
```

```
if (distance<100){
```

```
digitalWrite(12,HIGH);
```

```
digitalWrite(10,HIGH);
```

```

}
else{
digitalWrite(12,LOW);
digitalWrite(10,LOW);
digitalWrite(11,HIGH);
}
}

```

Assignment 2

Build a python code ,Assume a get temprature and humiddity values and write a condition to continuously detect alarm in case of high temprature.

```
# import standard python modules.
```

```
import time
```

```
# import adafruit dht library.
```

```
import Adafruit_DHT
```

```
# import Adafruit IO REST client.
```

```
from Adafruit_IO import Client, Feed
```

```
# Delay in-between sensor readings, in seconds.
```

```
DHT_READ_TIMEOUT = 5
```

```
# Pin connected to DHT22 data pin
```

```
DHT_DATA_PIN = 26
```

```
# Set to your Adafruit IO key.
```

```
# Remember, your key is a secret,
```

```

# so make sure not to publish it when you publish this code!
ADAFRUIT_IO_KEY = 'YOUR_AIO_KEY'

# Set to your Adafruit IO username.
# (go to https://accounts.adafruit.com to find your username).
ADAFRUIT_IO_USERNAME = 'YOUR_AIO_USERNAME'

# Create an instance of the REST client.
aio = Client(ADAFRUIT_IO_USERNAME, ADAFRUIT_IO_KEY)

# Set up Adafruit IO Feeds.
temperature_feed = aio.feeds('temperature')
humidity_feed = aio.feeds('humidity')

# Set up DHT22 Sensor.
dht22_sensor = Adafruit_DHT.DHT22

while True:
    humidity, temperature = Adafruit_DHT.read_retry(dht22_sensor, DHT_DATA_PIN)
    if humidity is not None and temperature is not None:
        print('Temp={0:0.1f}*C Humidity={1:0.1f}%'.format(temperature, humidity))
        # Send humidity and temperature feeds to Adafruit IO
        temperature = '%.2f'%(temperature)
        humidity = '%.2f'%(humidity)
        aio.send(temperature_feed.key, str(temperature))
        aio.send(humidity_feed.key, str(humidity))
    else:
        print('Failed to get DHT22 Reading, trying again in ', DHT_READ_TIMEOUT, 'seconds')
# Timeout to avoid flooding Adafruit IO

```

```
time.sleep(DHT_READ_TIMEOUT)
```

Asssignment 3

Write a Python code for blinking LED and Traffic lights using Raspberry Pi

Blinking LED

```
import time
import RPi.GPIO as GPIO          ## Import GPIO library
GPIO.setmode(GPIO.BOARD)        ## Use board pin
numbering
GPIO.setup(11, GPIO.OUT)         ## Setup GPIO Pin 11 to
OUT
while True:
    GPIO.output(11,True)         ## Turn on Led
    time.sleep(1)                ## Wait for one second
    GPIO.output(11,False)       ## Turn off Led
    time.sleep(1)                ## Wait for one second
```

Traffic Lights

```
from gpiozero import LED

red = LED(22)
amber = LED(27)
green = LED(17)

red.blink(1, 1)
amber.blink(2, 2)
green.blink(3, 3)
```