const int trigPin = 9;

const int LED =11;

const int echoPin = 10;

long duration;

int distance;

void setup() {

Serial.println("Started");

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

pinMode(LED, OUTPUT);

Serial.begin(4800); // Starts the serial communication

}

void loop() {

// Clears the trigPin

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

// Sets the trigPin on HIGH state for 10 micro seconds

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

// Reads the echoPin, returns the sound wave travel time in microseconds

duration = pulseIn(echoPin, HIGH);

distance = duration \* 0.034 / 2;

Serial.print("Distance: ");

Serial.println(distance);

if(distance<100 && distance>20)

{

digitalWrite(LED, HIGH);

delay(200);

digitalWrite(LED, LOW);

delay(200);

}

if(distance<=20)

{

digitalWrite(LED, HIGH);

delay(50);

digitalWrite(LED, LOW);

delay(50);

}

}

Output:

Distance: 2

Distance: 3

Distance: 6

Distance: 7

Distance: 11

Distance: 15

Distance: 20

Distance: 29

Distance: 28

Distance: 30

Distance: 23

Distance: 22

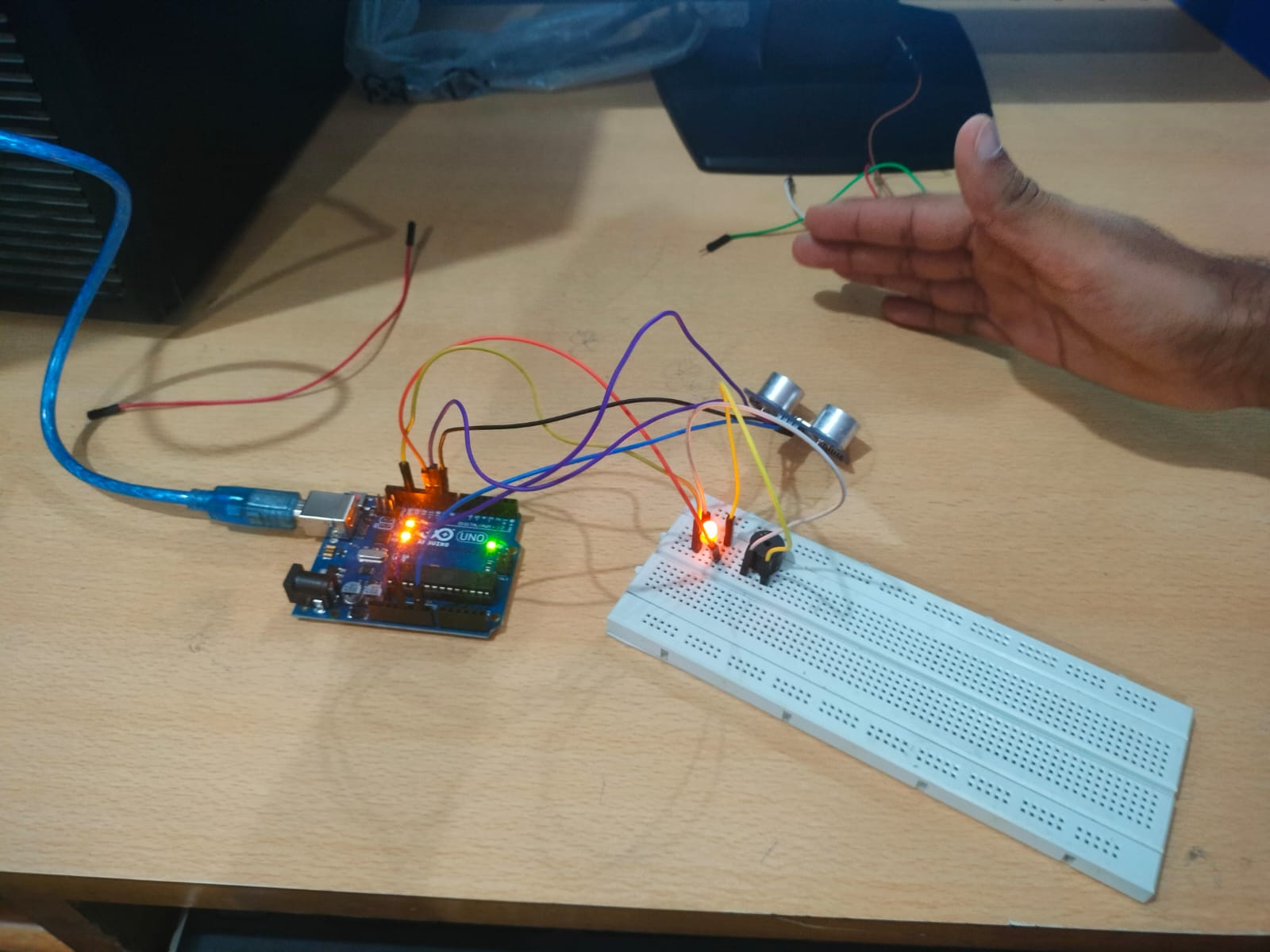
Distance: 28

Distance: 30

Distance: 29

Distance: 32

Distance: 92



Camera

from picamera import PiCamera

from time import sleep

camera=PiCamera()

camera.start\_preview()

#camera.resolution=(600,900)

camera.vflip=True

#camera.hflip=True

sleep(10)

camera.capture('/home/pi/Desktop/test\_d.jpg')

camera.stop\_preview()

Camcorder

from picamera import PiCamera

from time import sleep

camera=PiCamera()

camera.vflip=True

camera.start\_preview()

camera.start\_recording('/home/pi/d\_b.h264')

sleep(5)

camera.resolution=(600,900)

camera.stop\_recording()

camera.stop\_preview()



import Adafruit\_DHT as dht

from urllib.request import urlopen

myAPI = 'A0D4QMPIW4ALZDCG'

ThingsURL = 'https://api.thingspeak.com/update?api\_key=%s' % myAPI

def DHT11\_data():

hum1,temp1 = dht.read\_retry(dht.DHT11, 23)

return hum1,temp1

while True:

hum, temp = DHT11\_data()

if isinstance(hum, float) and isinstance(temp, float):

hum = '%.2f' % hum

temp = '%.2f' % temp

print(hum,temp)

coms = urlopen(ThingsURL + '&field1=%s&field2=%s' % (temp, hum))

print(coms.read())

