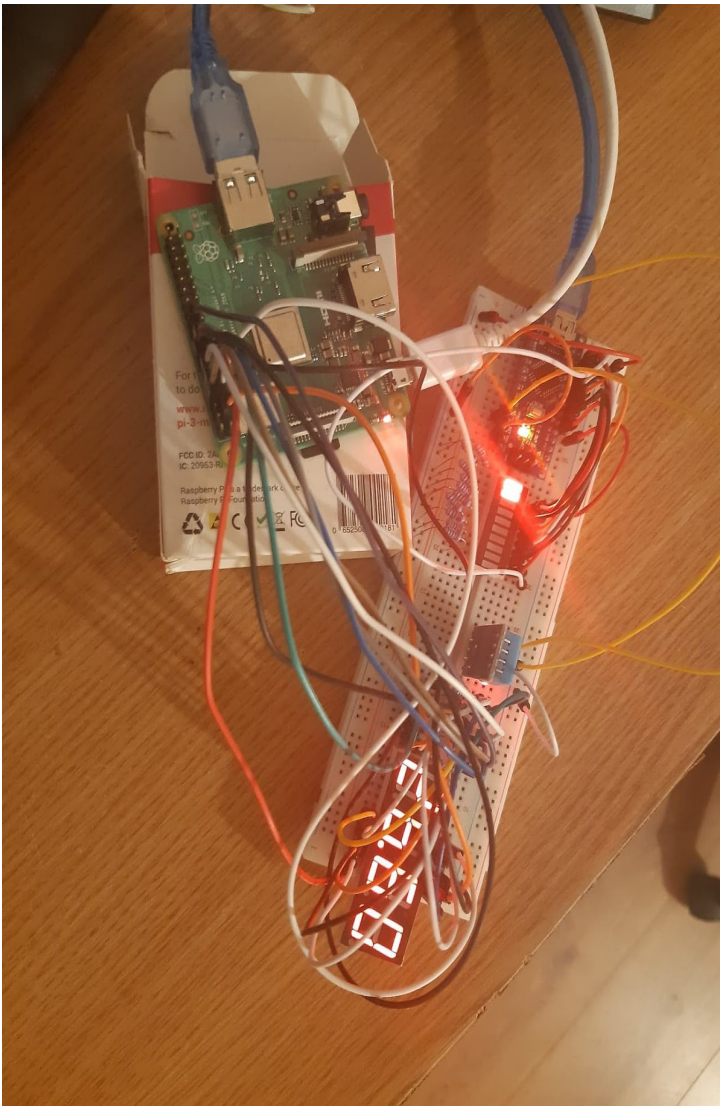


**Nume:** [Scripcaru Marian-Claudiu](#)

**Adresa de email:** [marian.scripcaru23@gmail.com](mailto:marian.scripcaru23@gmail.com)



**Titlul proiectului:** Senzor de temperatura si umiditate DHT11 cu notificare prin email.



**Descriere:** Senzorul DHT11 este conectat la arduino nano și citește temperatura și umiditatea o data la 2 secunde (pentru a nu strica senzorul), și o trimite serial către raspberry pi 3 unde se preia informația și se trimite prin email când este primit mesajul "send\r\n" de la arduino. Este conectata și o bara de 10 leduri care contorizeaza timpul rămas până la trimiterea informației prin email.

**Poveste:** Sistemul a fost gandit pentru a detecta defectiunile dintr-o camera frigorifica.

**Cum functioneaza?**

- Arduino nano citeste temperatura si umiditatea de la senzor si transmite serial informatia;
- Cand s-a incarcat bara de leduri este transmis mesajul "send\r\n" pentru a putea transmite pe email informatia dorita;

**Componente hardware:**

- Raspberry pi 3A+
- Arduino Nano
- Senzor temperatura DHT11
- 1 x Breadboard 830p
- 18 rezistente x 220 ohm
- 1 x bara de leduri cu 10 segmente
- 1 x 4 Digit 7-Segment Display
- 12 fire mama-tata
- 17 fire tata-tata

**Aplicatii:**

- Arduino
- Noobs

**Cod:**

**Arduino:**

```
#include "DHT.h"
#define DHTPIN 13
#define DHTTYPE DHT11

int pin=2;
DHT dht(DHTPIN, DHTTYPE, 6);
unsigned long previousMillis = 0;
unsigned long previousMillisSenzor = 0;
const long interval = 3000;
const long intervalSenzorRead = 2000;
bool read = true;
void setup() {
  Serial.begin(9600);
  delay(1000);
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
```

```
pinMode(5, OUTPUT);
pinMode(6, OUTPUT);
pinMode(7, OUTPUT);
pinMode(8, OUTPUT);
pinMode(9, OUTPUT);
pinMode(10, OUTPUT);
pinMode(11, OUTPUT);

Serial.println("DHTxx test!");
dht.begin();

}

void loop() {
  unsigned long currentMillis = millis();
  if(pin==12)
  {
digitalWrite(2 , LOW);
digitalWrite(3 , LOW);
digitalWrite(4 , LOW);
digitalWrite(5 , LOW);
digitalWrite(6 , LOW);
digitalWrite(7 , LOW);
digitalWrite(8 , LOW);
digitalWrite(9 , LOW);
digitalWrite(10 , LOW);
digitalWrite(11 , LOW);

Serial.println("send");
pin=2;
  }
  else
  {
if (currentMillis - previousMillis >= interval)
{
  previousMillis = currentMillis;
  digitalWrite(pin, HIGH);
  delay(3000);
  pin=pin+1;
}
```

```

}
}
if (currentMillis - previousMillisSensor >= intervalSensorRead)
{
previousMillisSensor = currentMillis;
if (read == true)
{
float h = dht.readHumidity();
read = false;
if (isnan(h) ) {
Serial.println("Failed to read from DHT sensor!");
return;
}
Serial.print("H:");
Serial.println(h);
}
else
{
float t = dht.readTemperature();
read = true;
if (isnan(t) ) {
Serial.println("Failed to read from DHT sensor!");
return;
}
Serial.print("T:");
Serial.println(t);
}
}
}

```

### **Raspberry:**

```

import smtplib
import RPi.GPIO as GPIO
import time
import serial
GPIO.setmode(GPIO.BOARD)

#serial
ser= serial.Serial('/dev/ttyUSB0', 9600, timeout=0.001)

```

```

#email
SMTP_SERVER = 'smtp.gmail.com'
SMTP_PORT = 587
GMAIL_USERNAME = 'raspberrypi2020v1@gmail.com'
GMAIL_PASSWORD = 'Raspberrypi20!'

class Emailer:
def sendmail(self, recipient, subject, content):

    #Create Headers
    headers = ["From: " + GMAIL_USERNAME, "Subject: " + subject, "To: " + recipient,
              "MIME-Version: 1.0", "Content-Type: text/html"]
    headers = "\r\n".join(headers)
    #Connect to Gmail Server
    session = smtplib.SMTP(SMTP_SERVER, SMTP_PORT)
    session.ehlo()
    session.starttls()
    session.ehlo()

    #Login to Gmail
    session.login(GMAIL_USERNAME, GMAIL_PASSWORD)

    #Send Email & Exit
    session.sendmail(GMAIL_USERNAME, recipient, headers + "\r\n\r\n" + content)
    session.quit

sender = Emailer()

# GPIO ports for the 7seg pins
segments = (13, 15, 8, 10, 12, 16, 18, 22)
# 7seg_segment_pins (11,7,4,2,1,10,5,3) + 100R inline

for segment in segments:
GPIO.setup(segment, GPIO.OUT)
GPIO.output(segment, 0)

# GPIO ports for the digit 0-3 pins
digits = (7, 5, 3, 11)
# 7seg_digit_pins (12,9,8,6) digits 0-3 respectively

```

```

for digit in digits:
GPIO.setup(digit, GPIO.OUT)
GPIO.output(digit, 1)

num = {' ':(0,0,0,0,0,0,0),
'0':(1,1,1,1,1,1,0),
'1':(0,1,1,0,0,0,0),
'2':(1,1,0,1,1,0,1),
'3':(1,1,1,1,0,0,1),
'4':(0,1,1,0,0,1,1),
'5':(1,0,1,1,0,1,1),
'6':(1,0,1,1,1,1,1),
'7':(1,1,1,0,0,0,0),
'8':(1,1,1,1,1,1,1),
'9':(1,1,1,1,0,1,1)}
temp = 0
humidity = 0
previousTime = 0
timeprint = True
try:
while True:
    encoding = 'utf-8'
    read_serial = ser.readline()
    array_serial = read_serial.decode(encoding).split(':')
    #print(array_serial)
    if array_serial[0] == 'send\r\n':
        print(array_serial[0])
        sendTo = 'marian.scripcaru23@gmail.com'
        emailSubject = "Temperatura si umiditate!"
        emailContent = "Temperatura este: " +str(temp) + "\n" + "Umiditatea este:" + str(humidity)
        sender.sendmail(sendTo, emailSubject, emailContent)
    elif array_serial[0] == 'T':
        temp = float(array_serial[1][0:4])
    elif array_serial[0] == 'H':
        humidity = float(array_serial[1][0:4])
    if abs(time.localtime().tm_sec - previousTime) >= 2:
        previousTime = time.localtime().tm_sec
        if timeprint == True:
            timeprint=False

```

```

else:
    timeprint=True
if timeprint == True:
    n = temp*100
    # print(n)
else:
    n = humidity*100
    # print("humid:"+n)

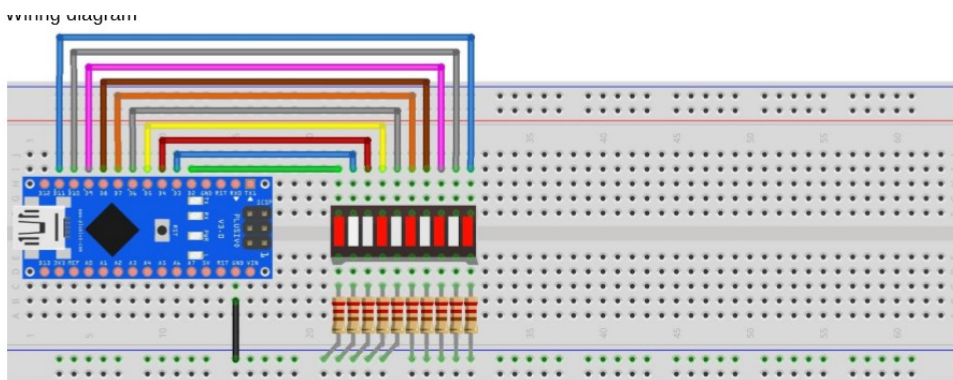
s = str(n).rjust(4)
for digit in range(4):
    for loop in range(0,7):
        GPIO.output(segments[loop], num[s[digit]][loop])
        if digit==1:
            GPIO.output(22, 1)
        else:
            GPIO.output(22, 0)
        GPIO.output(digits[digit], 0)
        time.sleep(0.001)
        GPIO.output(digits[digit], 1)
finally:
    GPIO.cleanup()

```

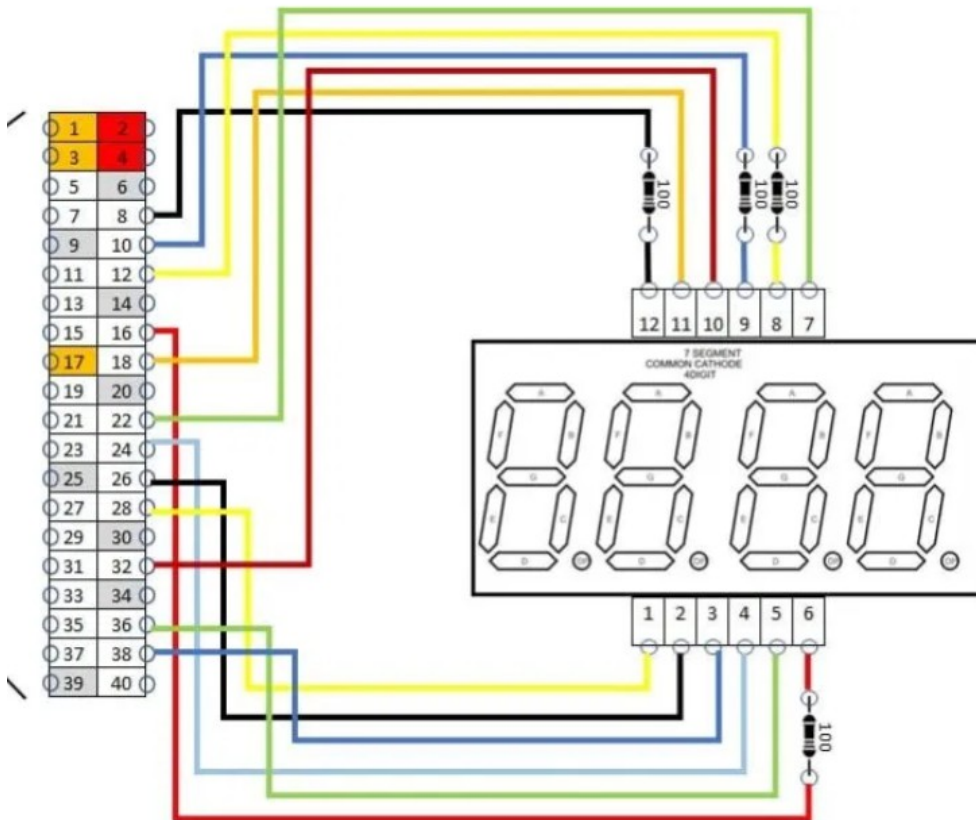
**Videoclip:** <https://youtu.be/suYMIX5DoI8>

**Schema:**

Bara leduri:



4 digit display:



Iar senzorul DHT11 este conectat la vcc si gnd la arduino nano iar pinul de data este conectat la pinul D13 a placutei. Comunicarea dintre cele 2 este realizata prin cablu USB - USB Tip A.

**Echipa 5**