TEMPERATURE CONTROLLED FAN WITH WEB APP

EMBEDDED PROGRAMMING PROJECT REPORT

Avick Sinha (202005812)
Section 1
Jabed Hossain (202002512)
Section 1
Mohammed Alauddin (202000512)
Section 1
Shuarthu Tapash Barua (202000112)
Section 1

Index:

1. Introduction:

The project is made using the microcontroller Arduino Uno, DHT11 sensor and 5V DC fan. Arduino Uno will be used to collect data from the Digital Temperature sensor and relay it to the website through serial communication. Codes are written in Arduino and JavaScript to integrate the hardware and software. The outcome will result a Fan that can be controlled from anywhere with data of the room being constantly updated on the website for a set threshold. The website infrastructure is created using Node-Red. A real-life use case could be at factory, where if the temperature of the room has exceeded a certain threshold value the fans will be automatically turned on until the room has been cooled down below the temperature threshold.

2. Methodology:

2.1. Hardware:

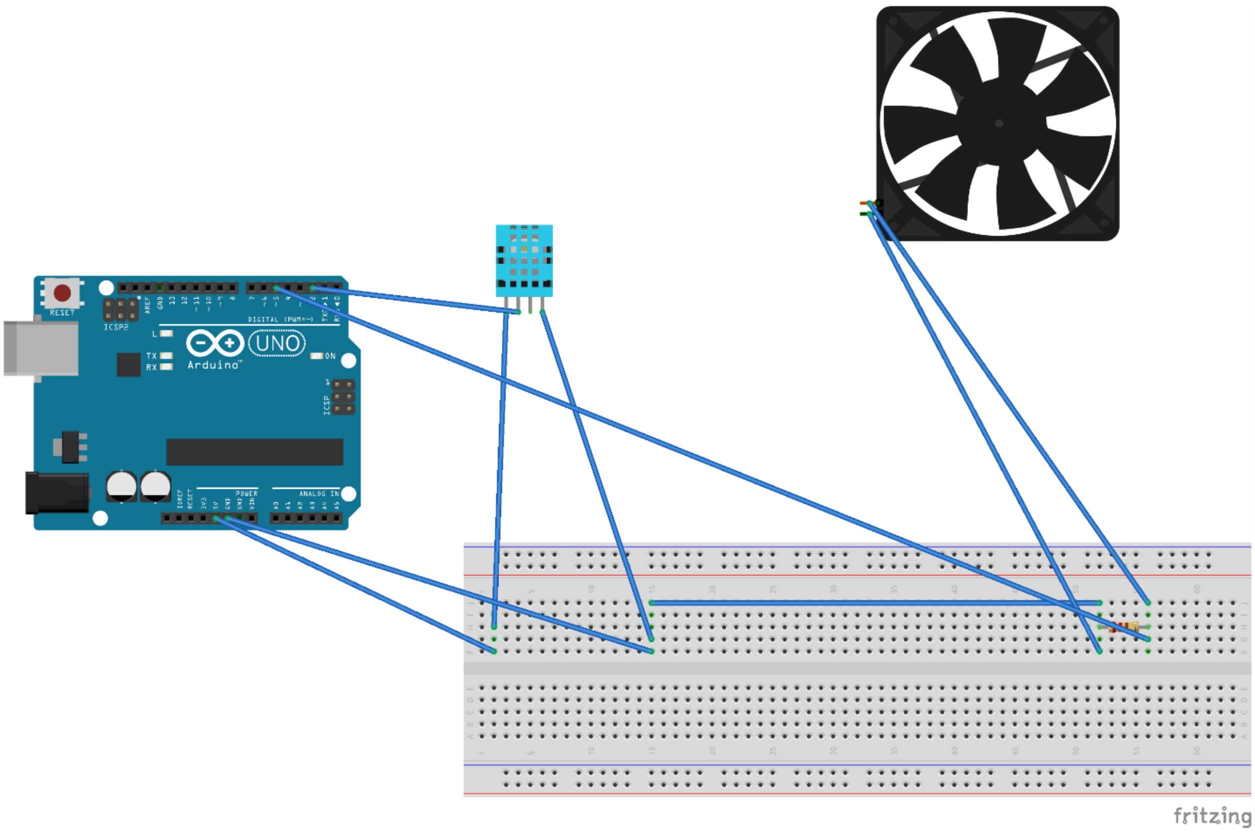
List of Components:

- Arduino Uno R3
- DHT11 Sensor
- Connecting Wire
- Breadboard
- 5V DC Fan
- Resistor

The connection between the components is described below

From the Arduino board a 5V connection and Ground connection is taken and connected on the breadboard. The VCC pin of the DHT11 is connected to the 5V on the breadboard and the GND ping of the DHT11 is connected to the Ground connection of the breadboard. The DATA pin of the DHT11 is connected to the Arduino Digital Pin 2 to get the data. The fan's live wire and ground pin is connected to the two ends of the resistor. The ground pin is connected to the ground presented on the breadboard and the live pin is connected to the Arduino Digital pin 5 which will be later programmed to be used as an Output.

The following figure illustrates the above explanation:



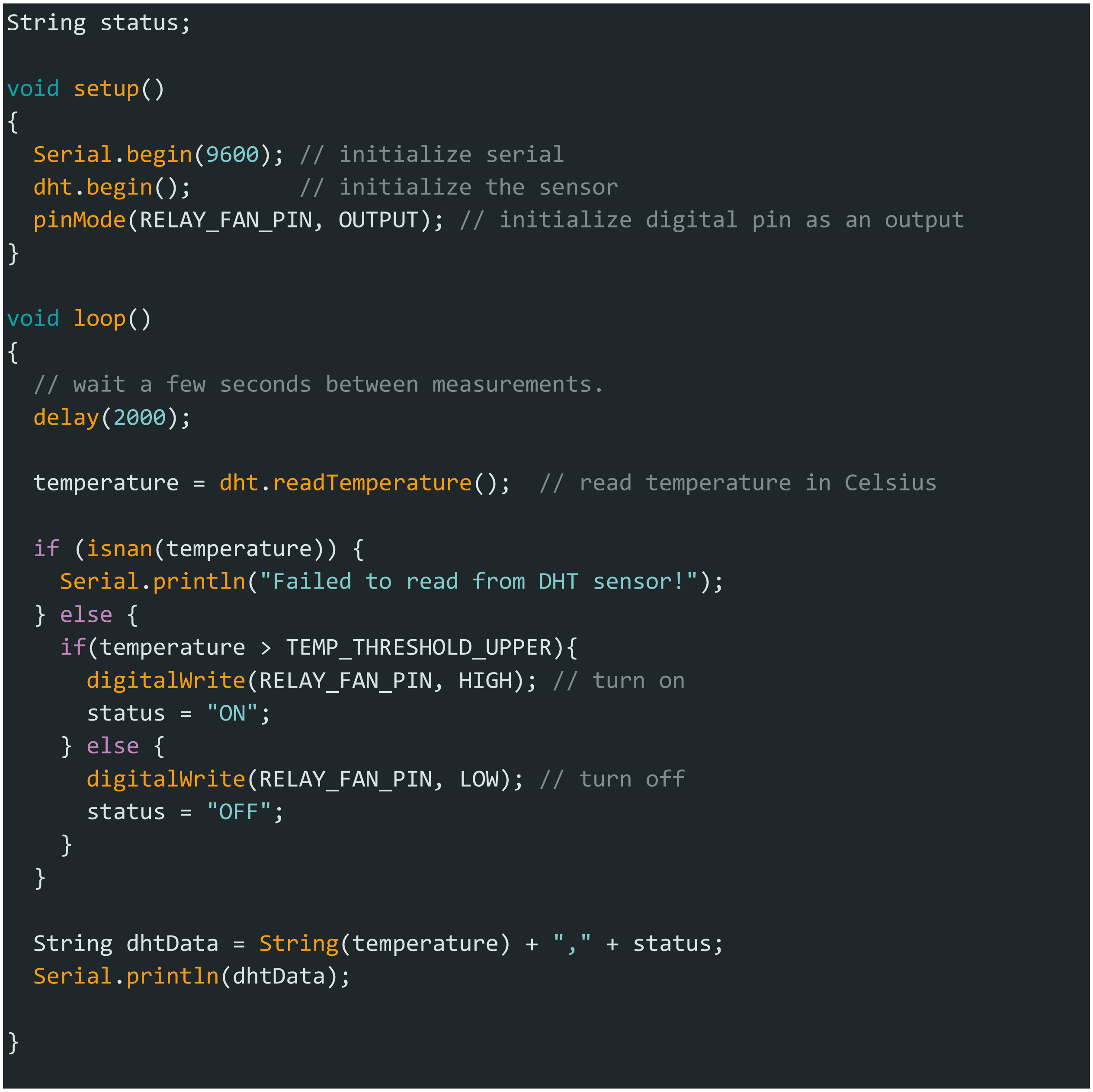
2.2. Software

The following software were used to create this project:

- Arduino IDE
- Node-RED

The Arduino IDE is used to write the code for the above mentioned functionality and upload the code to control the microcontroller. The code is attached below:

```
#include "DHT.h"

#define RELAY_FAN_PIN 5 // Arduino pin connected to relay which connected to fan
#define DHTPIN 2          // Arduino pin connected to relay which connected to DHT sensor
#define DHTTYPE DHT11

const int TEMP_THRESHOLD_UPPER = 33; // upper threshold of temperature, change to your desire value

DHT dht(DHTPIN, DHTTYPE);

float temperature;    // temperature in Celsius
```



```
String status;

void setup()
{
  Serial.begin(9600); // initialize serial
  dht.begin();        // initialize the sensor
  pinMode(RELAY_FAN_PIN, OUTPUT); // initialize digital pin as an output
}

void loop()
{
  // wait a few seconds between measurements.
  delay(2000);

  temperature = dht.readTemperature();  // read temperature in Celsius

  if (isnan(temperature)) {
    Serial.println("Failed to read from DHT sensor!");
  } else {
    if(temperature > TEMP_THRESHOLD_UPPER){
      digitalWrite(RELAY_FAN_PIN, HIGH); // turn on
      status = "ON";
    } else {
      digitalWrite(RELAY_FAN_PIN, LOW); // turn off
      status = "OFF";
    }
  }

  String dhtData = String(temperature) + "," + status;
  Serial.println(dhtData);

}
```

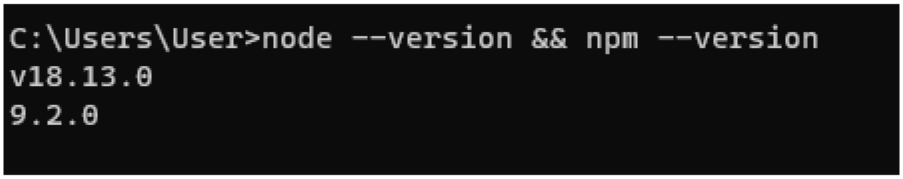
Node-RED: Node-RED is a flow-based development tool for visual programming developed originally by IBM for wiring together hardware devices, APIs and online services as part of the Internet of Things. Node-RED provides a web browser-based flow editor, which can be used to create JavaScript functions. The website is hosted locally using node red and the communication between the server and the Arduino is done through serial communication.

To install Node-RED follow the following steps:

- First install NodeJS from this website https://nodejs.org/en and download the latest version.

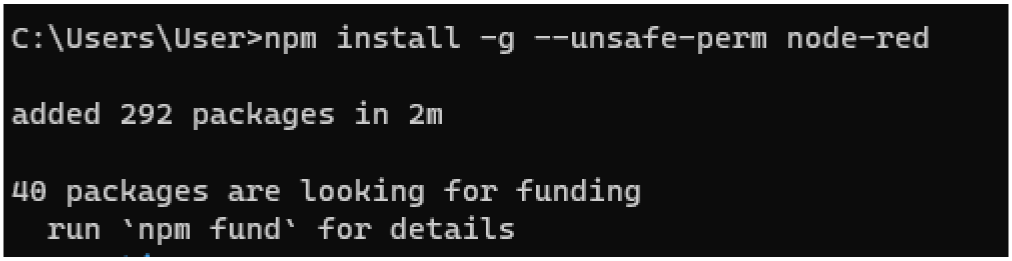
- After the installation is complete, Open Command Prompt and run this command

node --version && npm --version to make sure the installation is done properly. The output of it is given below:



```
C:\Users\User>node --version && npm --version
v18.13.0
9.2.0
```

- To install Node-RED on Windows the manual can be found in this website https://nodered.org/docs/getting-started/windows or follow these steps
- Open Command Prompt
- Run the command **npm install -g --unsafe-perm node-red** to install Node_RED. The output is given below:



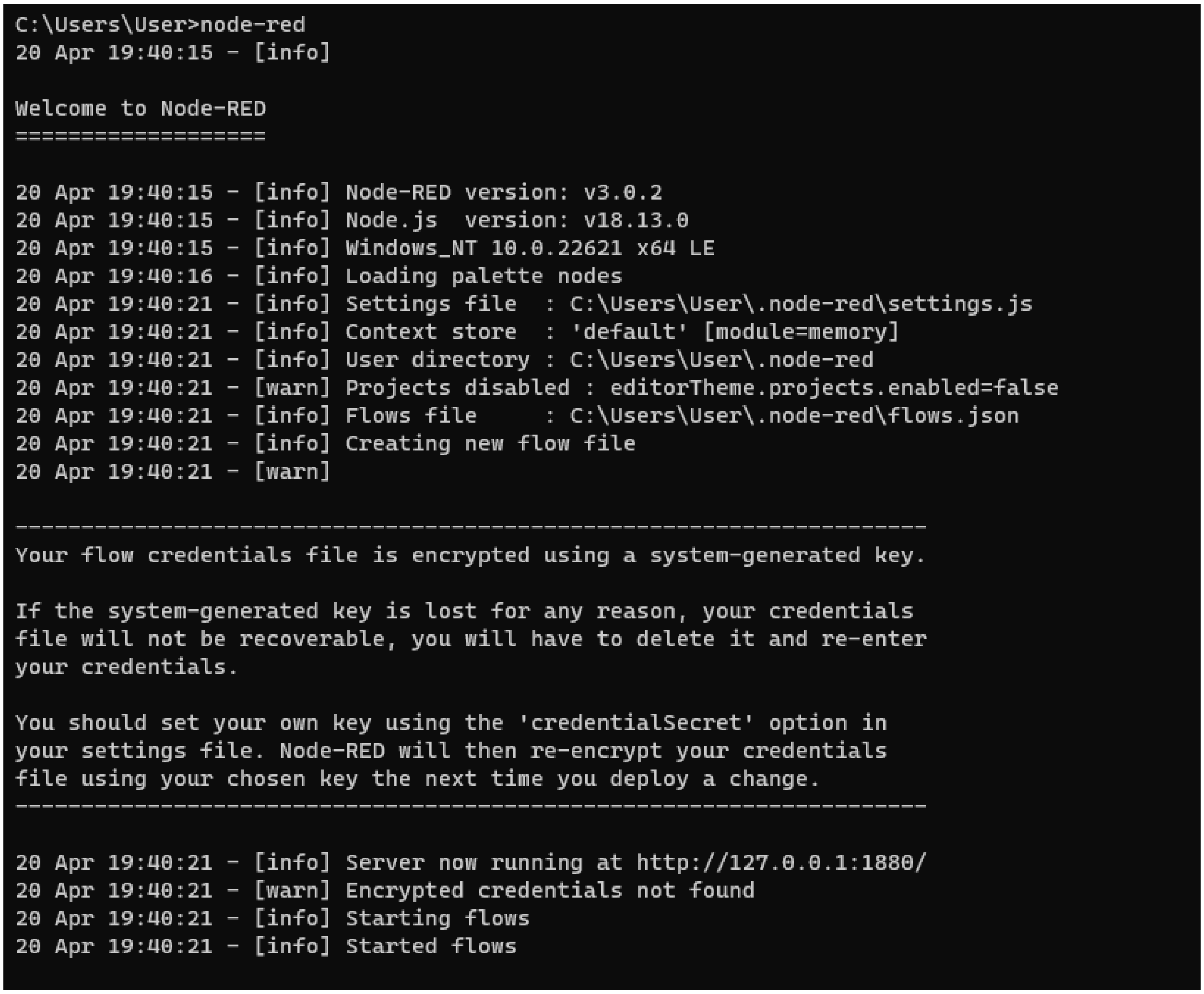
```
C:\Users\User>npm install -g --unsafe-perm node-red

added 292 packages in 2m

40 packages are looking for funding
  run `npm fund` for details
```

The installation is Node-RED is now complete.

Node-RED can be used in multiple ways such as deploying in Cloud-Server such as AWS and Microsoft Azure or can also be used locally. The instruction manual for all of them is given on this website: https://nodered.org/docs/getting-started/.

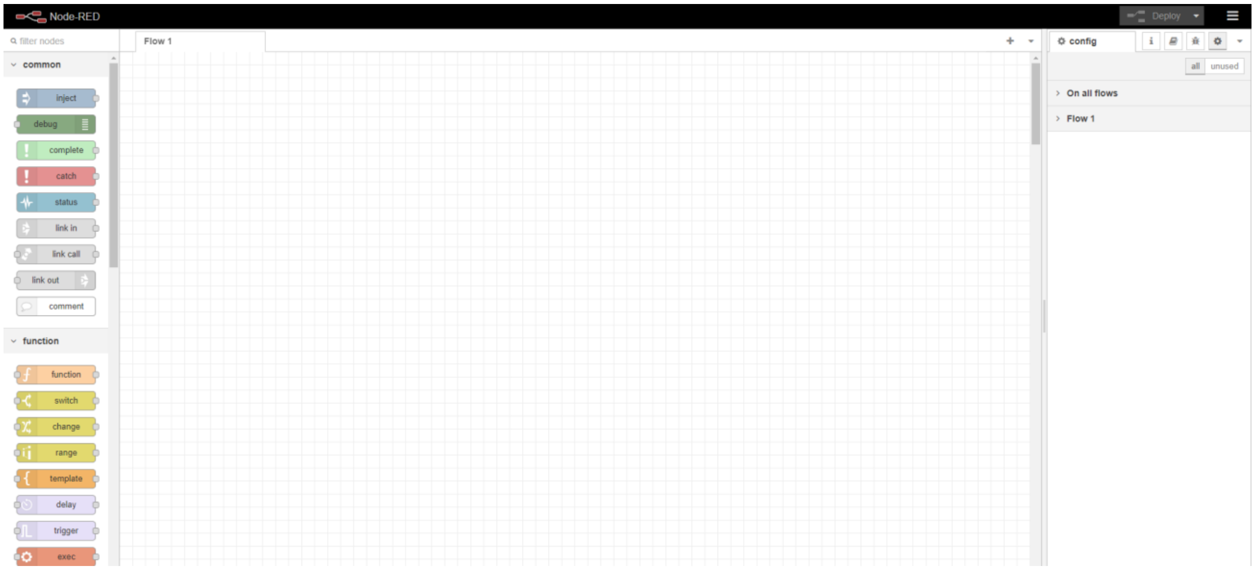
This report will explain how to run Node-RED locally on a device. The tutorial can be found in https://nodered.org/docs/getting-started/local or follow the steps below:

- First Open Command Prompt and run the command **node-red** this will initiate the server locally. The output is given below:



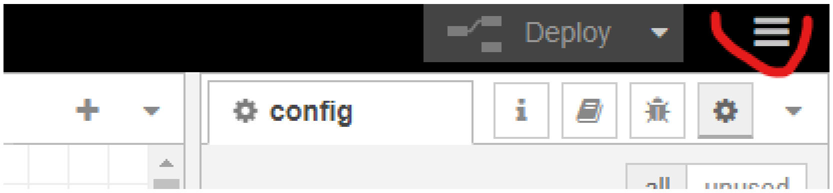
```
C:\Users\User>node-red
20 Apr 19:40:15 - [info]

Welcome to Node-RED
===================

20 Apr 19:40:15 - [info] Node-RED version: v3.0.2
20 Apr 19:40:15 - [info] Node.js  version: v18.13.0
20 Apr 19:40:15 - [info] Windows_NT 10.0.22621 x64 LE
20 Apr 19:40:16 - [info] Loading palette nodes
20 Apr 19:40:21 - [info] Settings file  : C:\Users\User\.node-red\settings.js
20 Apr 19:40:21 - [info] Context store  : 'default' [module=memory]
20 Apr 19:40:21 - [info] User directory : C:\Users\User\.node-red
20 Apr 19:40:21 - [warn] Projects disabled : editorTheme.projects.enabled=false
20 Apr 19:40:21 - [info] Flows file     : C:\Users\User\.node-red\flows.json
20 Apr 19:40:21 - [info] Creating new flow file
20 Apr 19:40:21 - [warn]

---------------------------------------------------------------------
Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
---------------------------------------------------------------------

20 Apr 19:40:21 - [info] Server now running at http://127.0.0.1:1880/
20 Apr 19:40:21 - [warn] Encrypted credentials not found
20 Apr 19:40:21 - [info] Starting flows
20 Apr 19:40:21 - [info] Started flows
```

- To access the server, open a browser and in the address bar input this link: http://localhost:1880/. This will open the Node-RED flow to program the devices.

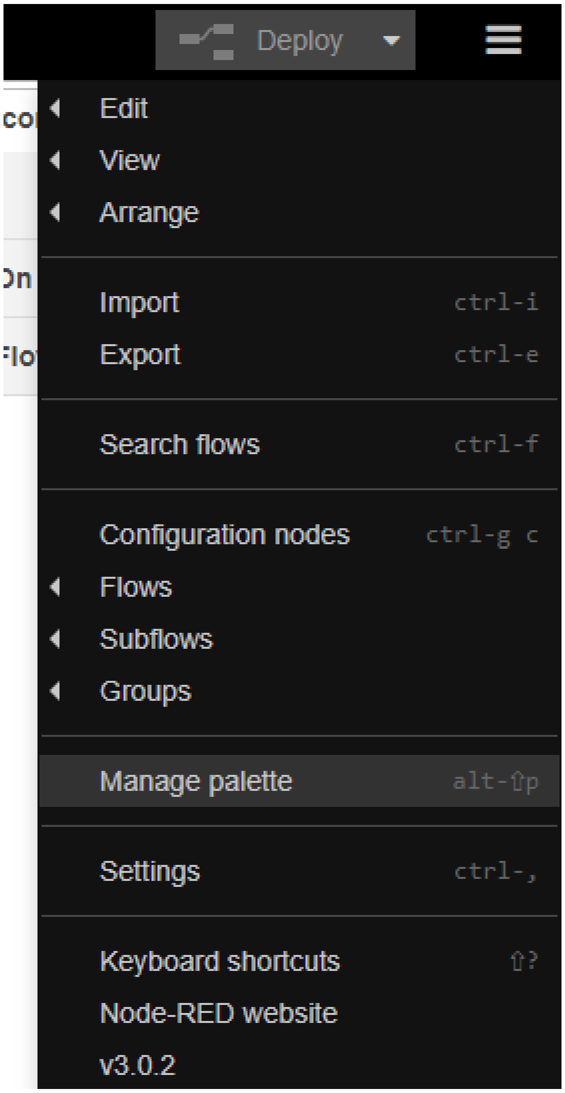


As mentioned above the connection between the Arduino and the Server is done serially. To do this some dependencies need to be installed.

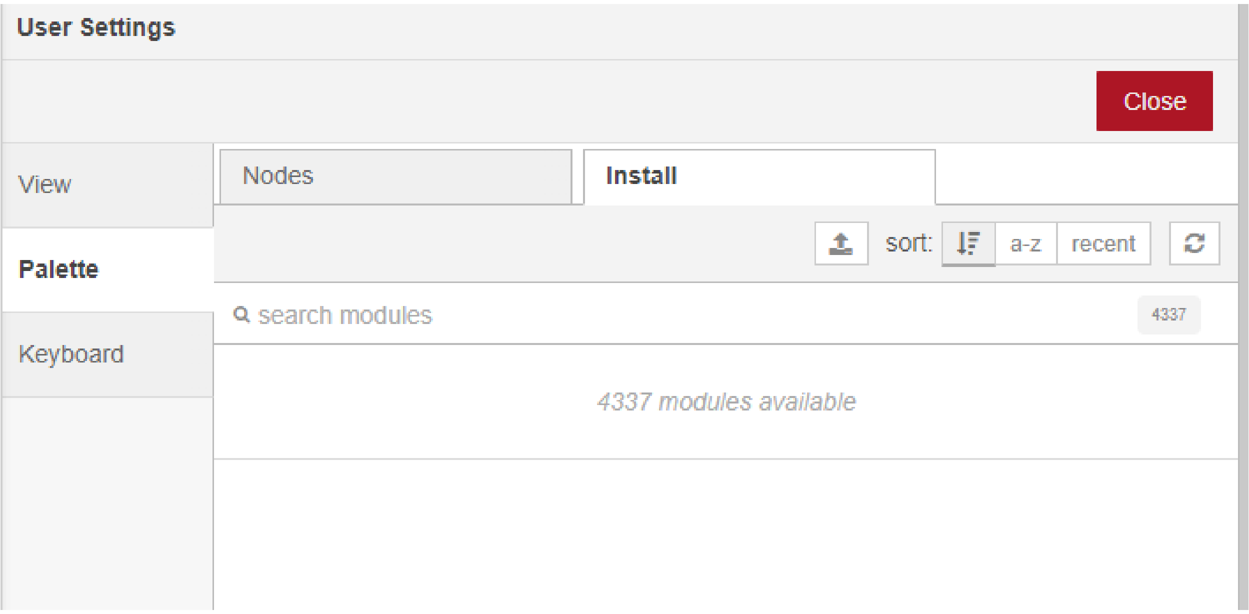
- On the Node-RED flow opened previously and click on the Top Right drop down menu



- From the menu select **Manage Pallete**

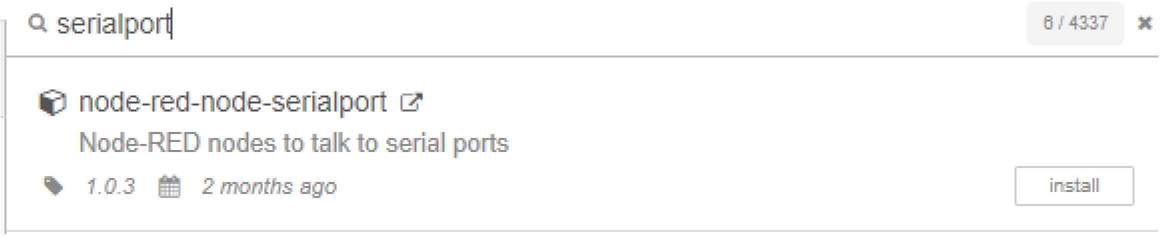


- It will open a user setting and from there select **Install**



4337 modules available

- On the search bar type **serialport** and install the first module



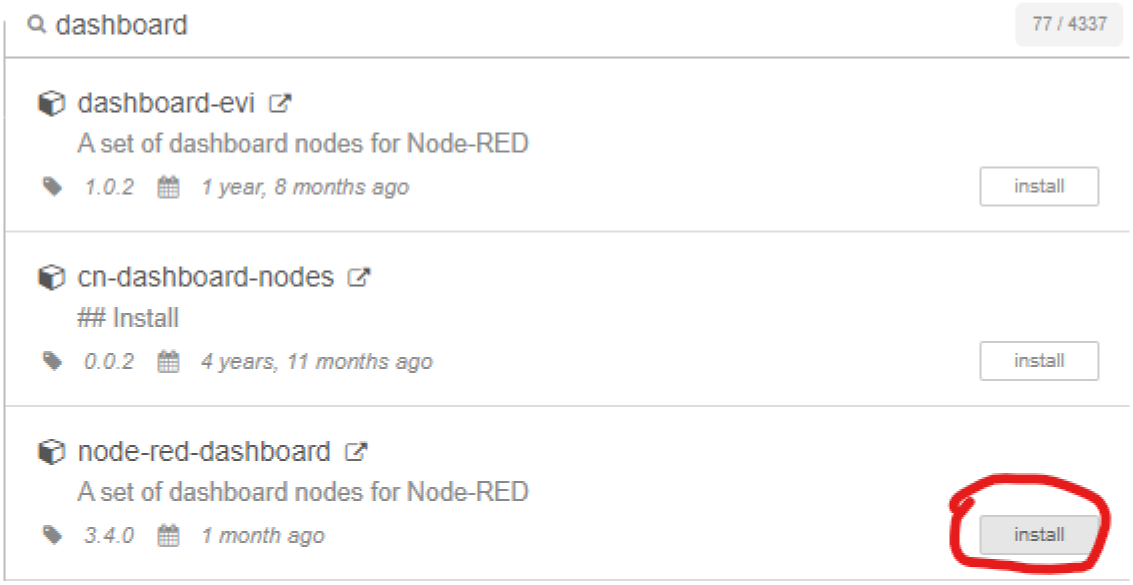
serialport

node-red-node-serialport

Node-RED nodes to talk to serial ports

1.0.3 2 months ago

- Now search **dashboard** and install the first module as well to get a live preview of the data in the website.



dashboard

dashboard-evi

A set of dashboard nodes for Node-RED

1.0.2 1 year, 8 months ago

cn-dashboard-nodes

Install

0.0.2 4 years, 11 months ago

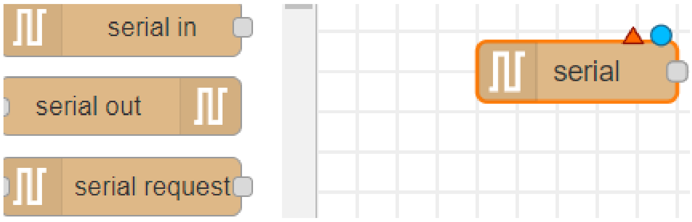
node-red-dashboard

A set of dashboard nodes for Node-RED

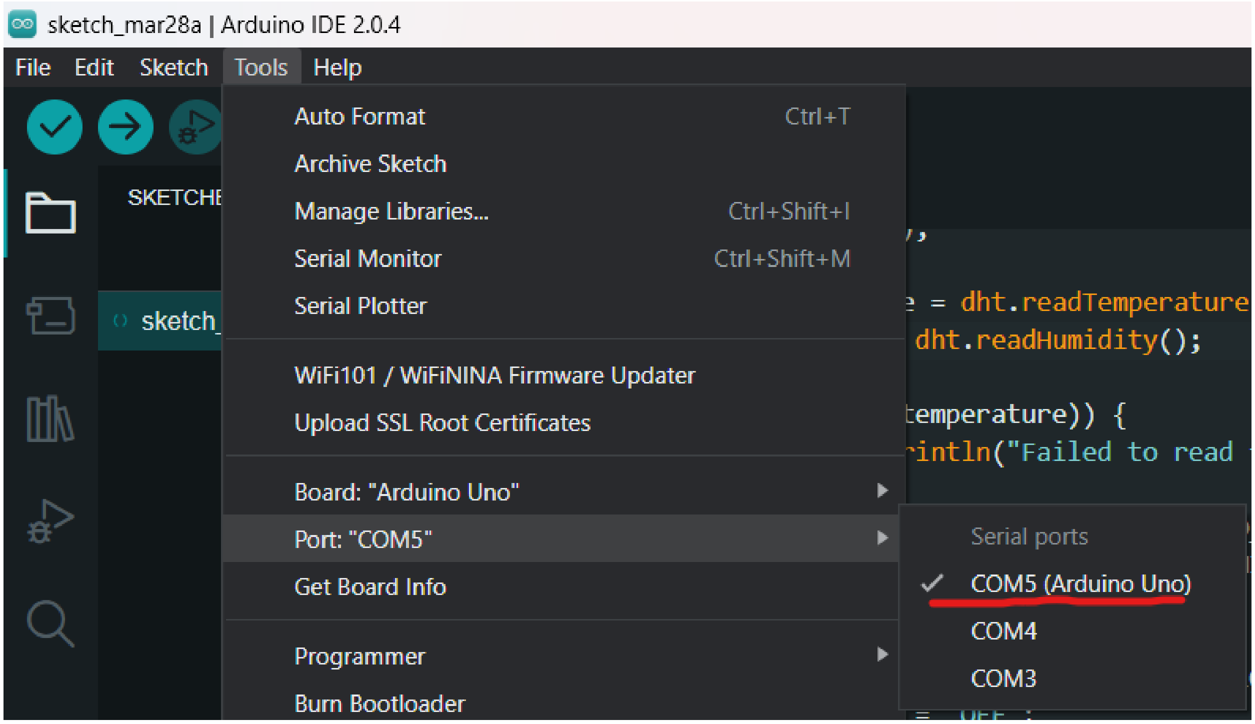
3.4.0 1 month ago

After the necessary dependencies are installed. The flow can be used to get the data from Arduino.

First, drag and drop the serial-in node from the side bar.



First get the USB port at which Arduino is connected from Arduino IDE


sketch_mar28a | Arduino IDE 2.0.4
File
Edit
Sketch
Tools
Help
SKETCHE
sketch_
Auto Format Ctrl+T
Archive Sketch
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter
WiFi101 / WiFiNINA Firmware Updater
Upload SSL Root Certificates
Get Board Info
Programmer
Burn Bootloader
Serial ports
COM4
COM3

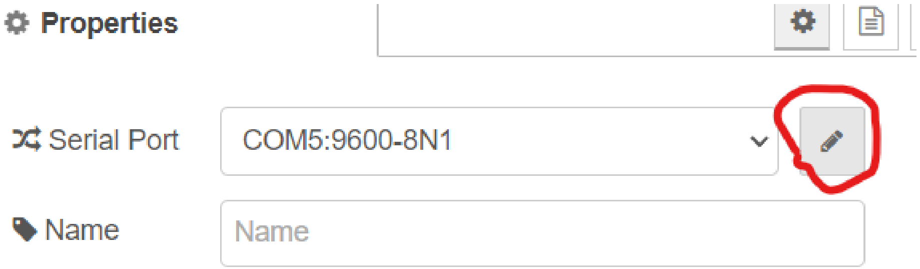
Board: "Arduino Uno"

Port: "COM5"

✓ COM5 (Arduino Uno)

```
e = dht.readTemperature
 dht.readHumidity();
temperature)) {
rintln("Failed to read
```

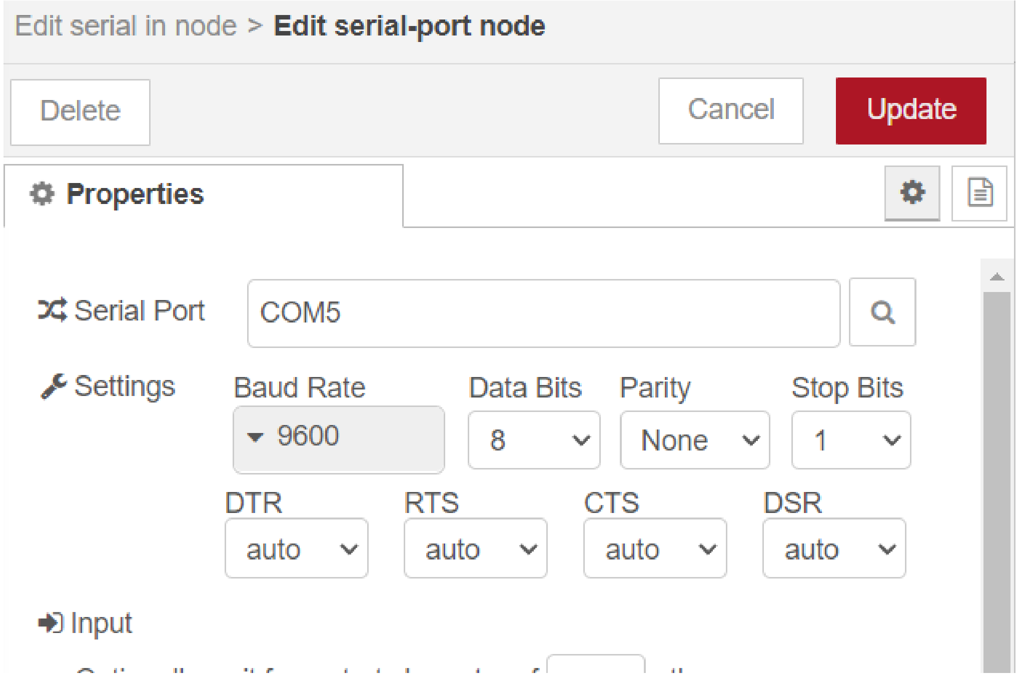
Configure the serial in by double clicking it. Click on edit on the panel.



Serial Port COM5:9600-8N1

Name Name

On the Serial Port type your serial port in this it is COM5 and set the Baud rate to 9600 as it is the rate for data transfer for Arduino UNO and set the name to Arduino for better understanding but its optional.



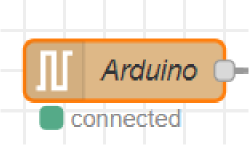
Serial Port: COM5

Settings: Baud Rate 9600, Data Bits 8, Parity None, Stop Bits 1

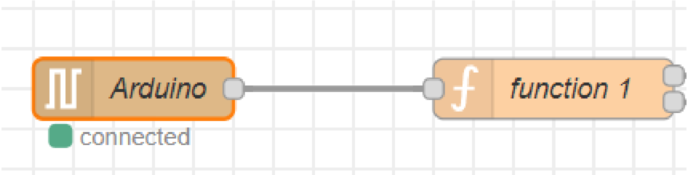
DTR auto, RTS auto, CTS auto, DSR auto

Input

If the configuration is successful it would appear connected below the serial in.



Now to process the data send by the Arduino to the Website we use a function node that takes in JavaScript code to process the data and give output to be shown in the website. In case of this project the data that are being send from the Arduino are Temperature and the Fan status. Both of the value are String.

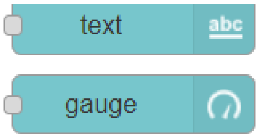


Inside the function 1 the following JS code is given and its configured to give two output as explained on the code:

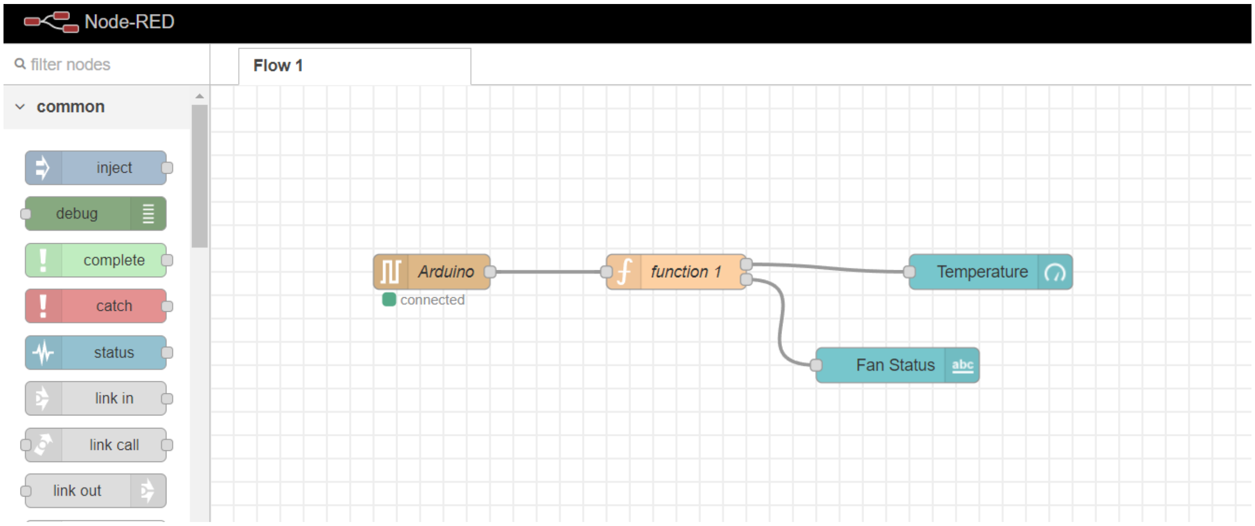
The JavaScript function used to transform the data from Arduino is given below:

```
var m = msg.payload.split(',');
var T = { payload: m[0] };
var H = { payload: m[1] };

return [T, H];
```

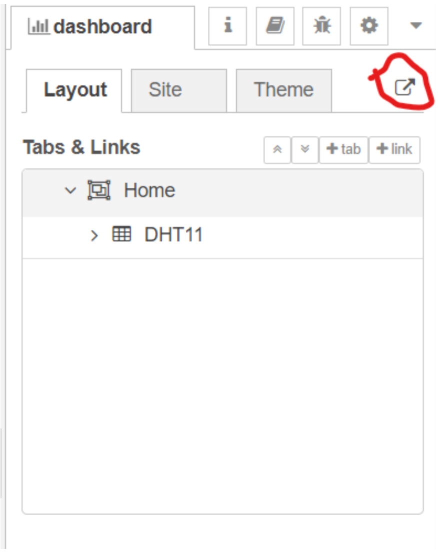
Now from the two output the text module of dashboard is used for fan status and the gauge module for the temperature data and is connected to the function.



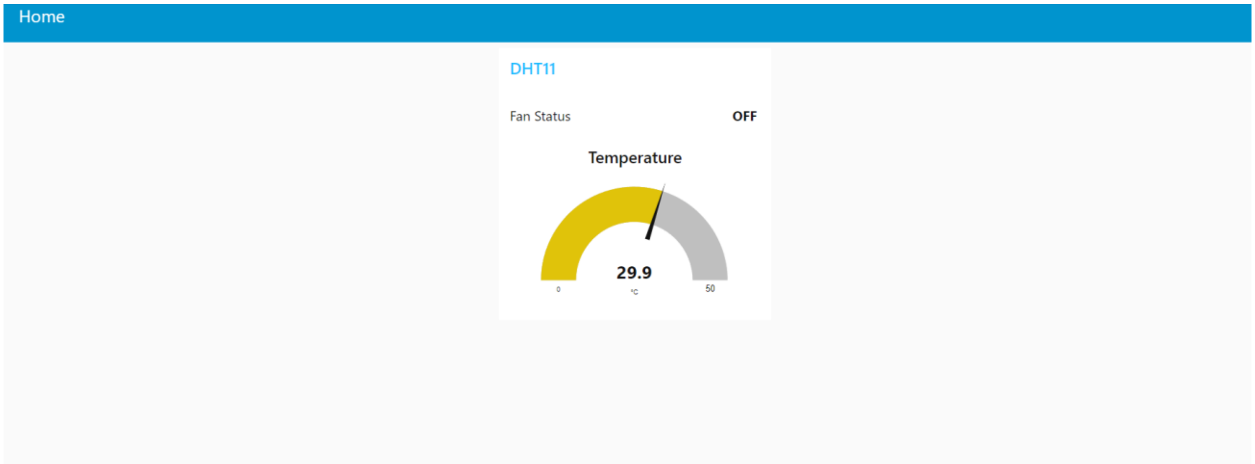
The completed flow in Node-RED is given below:



To get to the website click this button:



The finished website is given below:



Fan Status OFF