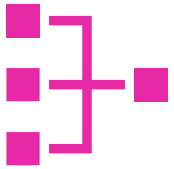
Weather Station

Luis Garcia Gamez

Adrian Dominguez Molina

Eros Ortega Cobos

INDEX



Demostration of the solution

Technical presentation

Technology choice

Behaviour of a thermistor

Argumented choice between technologies

Operation method explained

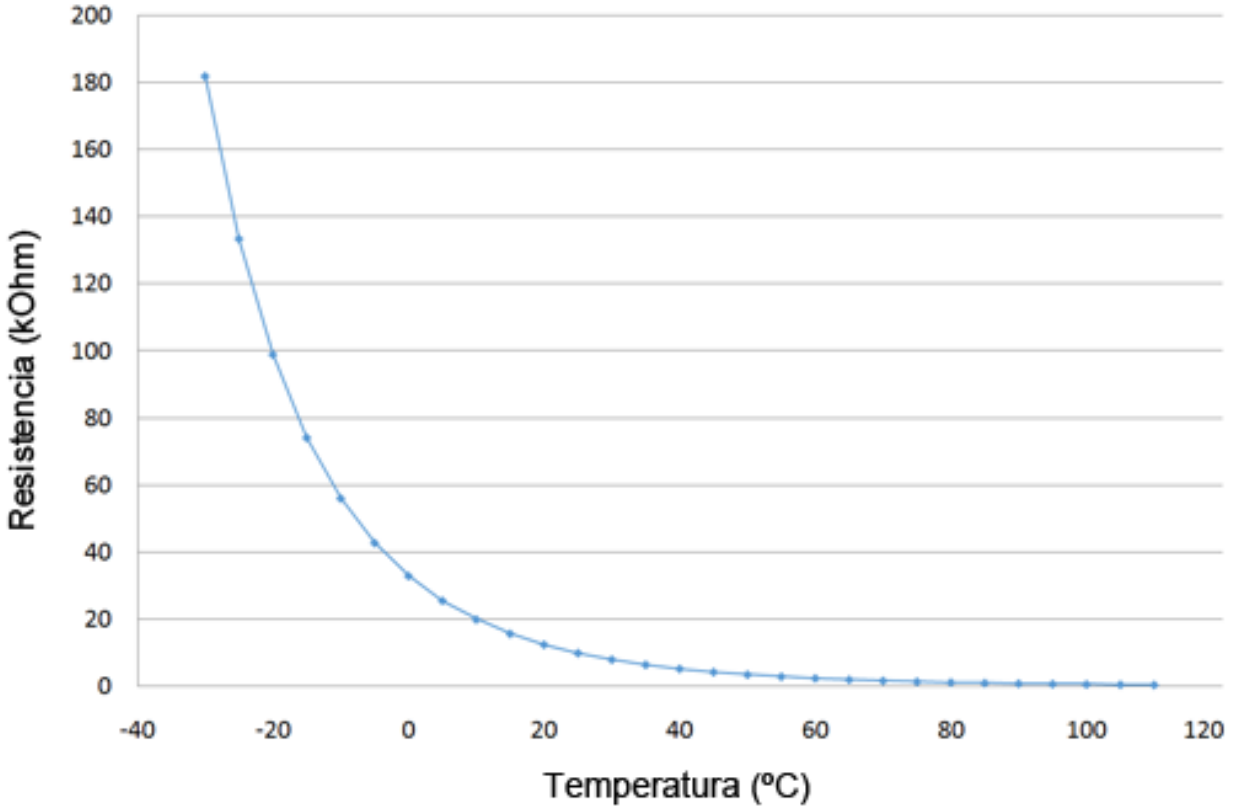
Capital gain ideas

Technical Presentation: Technology choice

Communication / Transport layer

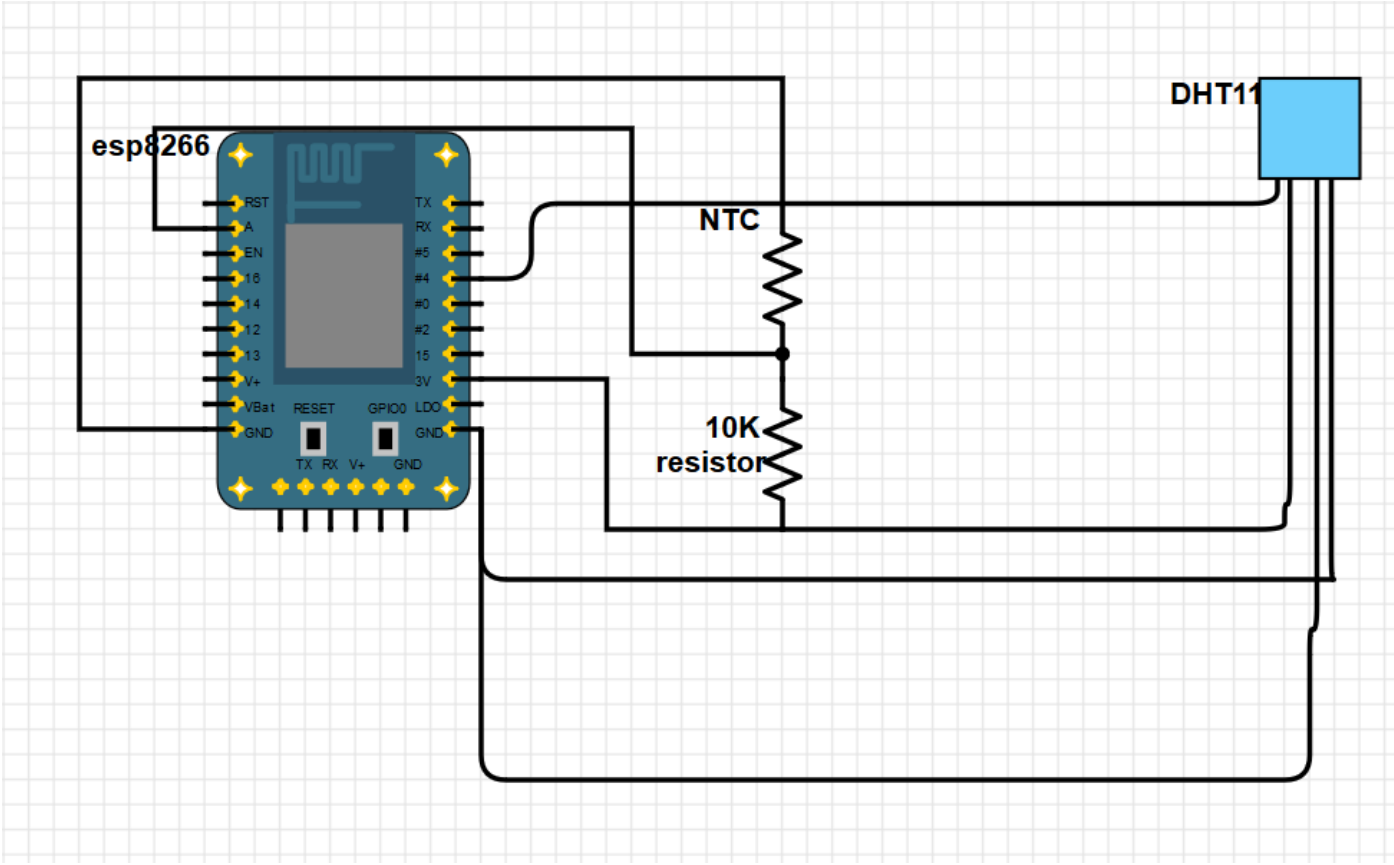
Technology	Frequency	Data Rate	Range	Power Usage	Cost
2G/3G	Cellular Bands	10 Mbps	Several Miles	High	High
Bluetooth/BLE	2.4Ghz	1, 2, 3 Mbps	~300 feet	Low	Low
802.15.4	subGhz, 2.4GHz	40, 250 kbps	> 100 square miles	Low	Low
LoRa	subGhz	< 50 kbps	1-3 miles	Low	Medium
LTE Cat 0/1	Cellular Bands	1-10 Mbps	Several Miles	Medium	High
NB-IoT	Cellular Bands	0.1-1 Mbps	Several Miles	Medium	High
SigFox	subGhz	< 1 kbps	Several Miles	Low	Medium
Weightless	subGhz	0.1-24 Mbps	Several Miles	Low	Low
Wi-Fi	subGhz, 2.4Ghz, 5Ghz	0.1-54 Mbps	< 300 feet	Medium	Low
WirelessHART	2.4Ghz	250 kbps	~300 feet	Medium	Medium
ZigBee	2.4Ghz	250 kbps	~300 feet	Low	Medium
Z-Wave	subGhz	40 kbps	~100 feet	Low	Medium

Technical Presentation: Behaviour of a thermistor



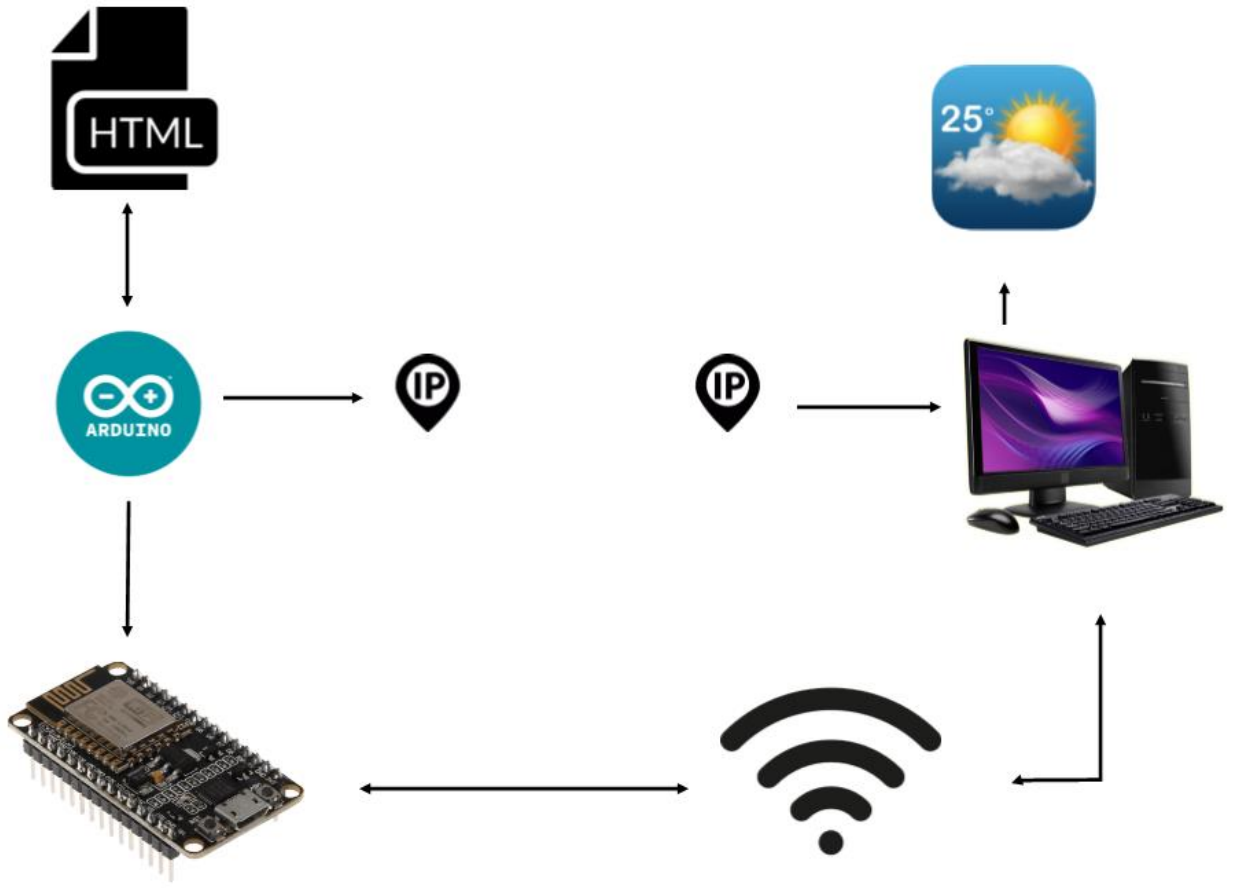
Technical presentation: scheme

- Wi-Fi technology with http protocol was chosen
- An NTC was used due a higher sensibility than the DHT11
- We read its value using a voltage divider



Technical presentation: Brief operation method explanation

- The Arduino code communicates with html file to send and receive data
- We upload it to our ESP8266 board
- With a pc connected to the same network as the board we can go to an IP using our favorite browser
- The Arduino code prints the IP where we must go



Technical presentation: Most important commands in the code

- One of the key commands is server.on
- "server.on" reclaims the data from the sensors and sends it to the server to be later displayed
- We later pick the data and represent it in charts and in a real time display

```
// Route for root / web page
server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){
  request->send(SPIFFS, "/index.html");
});
server.on("/temperature", HTTP_GET, [](AsyncWebServerRequest *request){
  request->send_P(200, "text/plain", readSensorTemperature().c_str());
});
server.on("/humidity", HTTP_GET, [](AsyncWebServerRequest *request){
  request->send_P(200, "text/plain", String(h).c_str());
});


// Start server
server.begin();
```

Technical presentation: HTML file

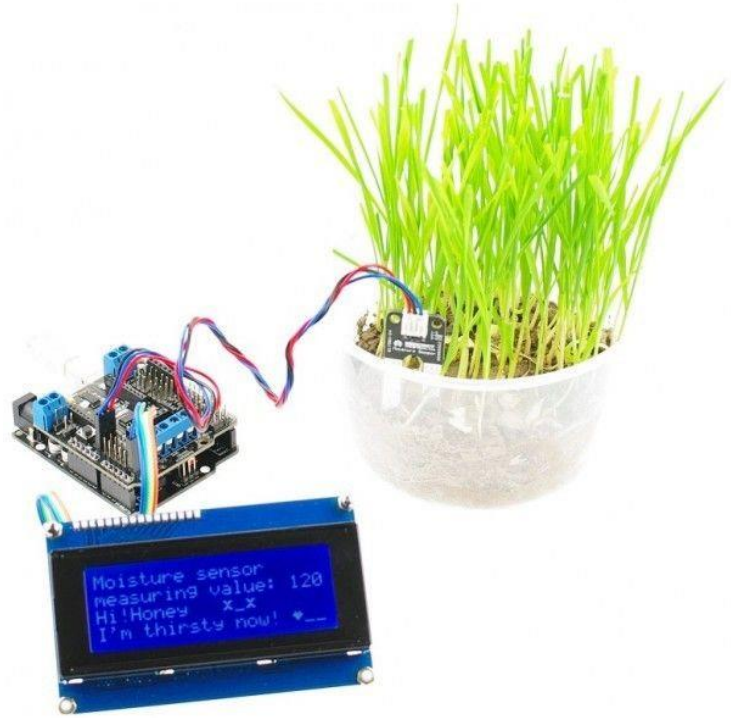
```
<!DOCTYPE HTML><html>
<head>
 <meta name="viewport" content="width=device-
  <link rel="stylesheet" href="https://use.for
  <style>
    html {
     font-family: Arial;
     display: inline-block;
     margin: 0px auto;
     text-align: center;
    }
    h2 { font-size: 3.0rem; }
    p { font-size: 3.0rem; }
    .units { font-size: 1.2rem; }
    .dht-labels{
      font-size: 1.5rem;
      vertical-align:middle;
      padding-bottom: 15px;
    }
  </style>
</head>
```

```
setInterval(function ( ) {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200)
      document.getElementById("humidity").innerHTML
    }
  };
  xhttp.open("GET", "/humidity", true);
  xhttp.send();
}, 10000 ) ;

var chartT = new Highcharts.Chart({
  chart:{ renderTo : 'chart-temperature' },
  title: { text: 'Temperature graphic' },
  series: [{
    showInLegend: false,
    data: []
  }],
  plotOptions: {
    line: { animation: false,
      dataLabels: { enabled: true }
    },
    series: { color: '#059e8a' }
  },
```

We have designed a well-looking webpage thanks of:

- Html structure
- CSS styles
- JavaScript for variables and graphs



Capital gain ideas

- Temperature and humidity control of rooms
- Ambience control for plants