

Escenario IoT utilizando sensor DHT11 para medir temperatura y humedad ambiente con Raspberry Pi

Valencia Mosquera, Juan Felipe, juanvalencia@unicauca.edu.co,
Universidad del Cauca

Resumen— En la actualidad los dispositivos tecnológicos cada vez son más económico y de mejor calidad, al punto de que toda persona puede llegar a acceder fácilmente a artefactos con alta capacidad tecnológica. Lo anterior ha popularizado el uso del internet de las cosas (IoT – Internet of Things) y los objetos inteligentes (IO – Intelligent Object). Por lo tanto, cada vez es más necesario generar escenarios inteligentes (compuestos por sensores, actuadores y servicios) de la IoT. Actualmente las investigaciones al respecto, han encontrado diferentes soluciones tecnológicas que incluyen el uso de objetos inteligentes como solución a un problema cotidiano de la humanidad, potenciando el uso de los mismos, la cual permite combinar elementos virtuales con el mundo real y cotidiano de los seres humanos. La mayoría de proyectos tienen como aliados diferentes tecnologías asociados a los objetos inteligentes de la IoT, entre los cuales se destacan tecnologías como la AR (Augmented Reality) para representar información de los IO a los usuarios, su dificultad es que son desarrollos a la medida, sin posibilidad de ser reutilizados.

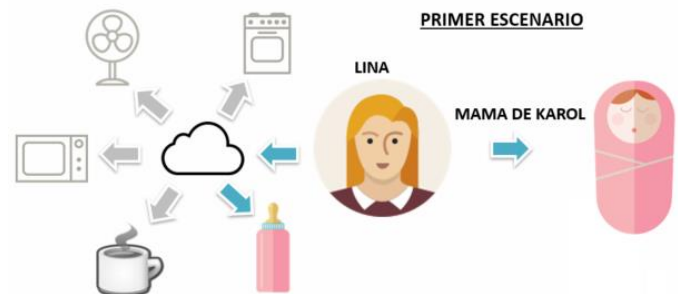
Este artículo se enfoca en presentar la construcción de un escenario IoT que permita medir temperatura y humedad ambiente y dado el caso aplicarlo a cualquier escenario de la cotidianidad humana.

Palabras claves — Internet de las cosas, Sensor de temperatura DHT11, Sensor de humedad DHT11, Raspberry Pi.

I. INTRODUCCIÓN

Los smartphones han dado pie a que la alta tecnología, sea algo que cada individuo en la actualidad “cargue en la palma de su mano”. Por primera vez se llegó a un estado de interconexión con todo el mundo donde las personas están interconectadas unas con las otras. El crecimiento exponencial de las tecnologías [1], han cerrado brechas entre los usuarios y el internet de las cosas [2], provocando que la IoT [3] siga en aumento y cada vez sea más necesario tener escenarios reales de IoT que permitan automatizar diferentes procesos de los seres humanos, como lo es la toma de temperatura y humedad ambiente y realizar según los datos una acción llevada a cabo por un actuador como encender un ventilador o apagarlo según sea necesario.

Existen diferentes escenarios sobre los cuales se piensa el uso de la IoT y sus objetos inteligentes, por ejemplo, el uso cotidiano de una madre que quiere dar de beber tetero a su hijo, sin embargo, no sabe a que temperatura exacta se encuentra el líquido contenido en el tetero, por tal motivo con el uso de los objetos inteligentes puede fácilmente observar a que temperatura se encuentra el líquido y tomar decisiones según sea necesario:



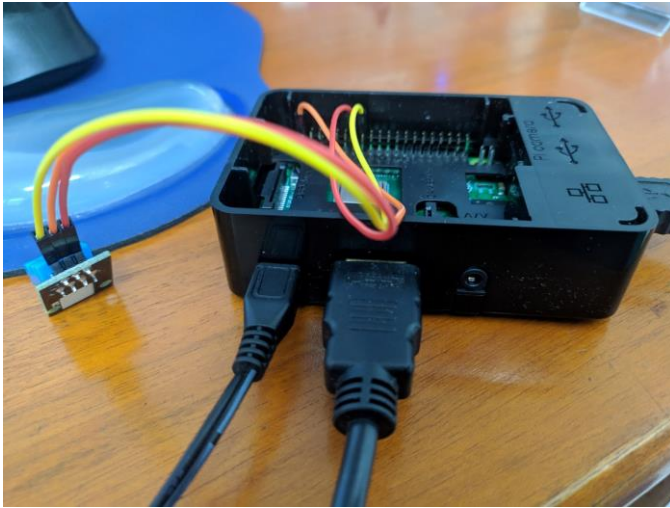
Lina puede mediante el uso de la IoT determinar a que temperatura se encuentra el líquido y tomar decisiones; otro escenario que se puede lograr es el de pedro:



Escenario sobre el cual pedro en su oficina desea tomar una tasa de café, sin embargo, no sabe a que temperatura se encuentra el líquido; bajo el uso de IoT puede ver a que temperatura se encuentra el líquido y tomar decisiones.

II. DESARROLLO DEL ESCENARIO

Una vez teniendo en cuenta el sin numero de escenarios que se pueden crear con el uso de la IoT, se presenta en este artículo la guía de pasos con el cual se llevó a cabo la creación de un escenario que me permite saber la temperatura y humedad ambiente detallando la instalación y creación del escenario con un sensor DHT11 y una raspberry pi.

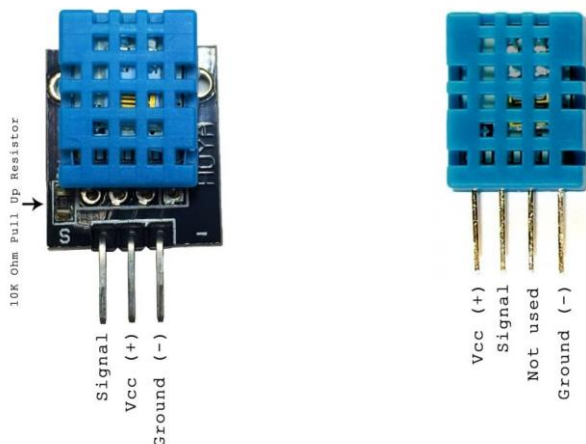


III. MATERIALES

- Raspberry pi
- Sensor de temperatura DHT11
- Jumpers de conexión (Cableado)

IV. CONSIDERACIONES PREVIAS RESPECTO AL SENSOR

Actualmente existen dos modelos de sensor DHT11. Podríamos decir que uno es más moderno que otro. En función del que tengamos las conexiones en los pines son diferentes. En la siguiente imagen vemos la comparativa de ambos sensores.



El sensor de la izquierda es el nuevo sensor DHT11 en el cual lleva integrada en su placa una resistencia de 10K necesaria para su funcionamiento. Mientras que el sensor del lado derecho es un poco más antiguo necesita que se añada una resistencia externa en la protoboard con el objetivo de cuidar el sensor y no sea susceptible a daños por variaciones en la energía. Para nuestro escenario contamos con el sensor de la izquierda el cual nos evita el uso de protoboard.

V. CONFIGURACION DE LA RASPBERRY PI

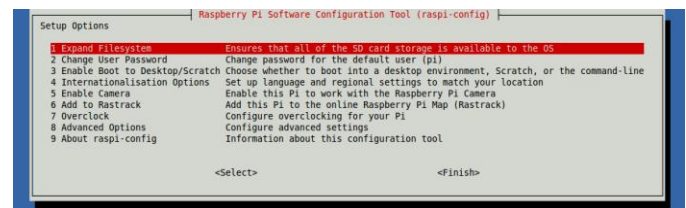
Inicialmente se debe instalar el sistema operativo de la raspberry utilizando una microSD. Para ello se descarga desde la página <https://www.raspberrypi.org/downloads/>

Una vez descargado el sistema operativo se añade el archivo descargado como imagen ISO a la tarjeta SD, posterior a ello se coloca la SD a la raspberry para poder que cuando la raspberry se encienda se ejecute el sistema operativo.

VI. CONFIGURACION DE SSH PARA LA RASPBERRY

Para poder realizar conexiones remotas con la raspberry Pi desde nuestro PC. Escribiremos en un terminal el comando

\$ sudo raspi-config



Posterior a ejecutar el comando desde la terminal se muestra la imagen anterior otorgando las opciones, para encender el SSH se realizan los siguientes pasos:

1. Seleccionamos la opción **8 Advanced Options**
2. Elegimos la opción de menú **SSH**
3. Seleccionamos **Yes**
4. Presionamos **Ok**
5. Elegimos la opción **Finish**

Una vez realizados estos pasos, la raspberry tiene habilitado la opción de generar conexiones de escritorio remotas (SSH).

VII. PREPARACION DEL SOFTWARE DHT11

Para iniciar la configuración del escenario principalmente actualizaremos la lista de paquetes en nuestra Raspberry Pi con el siguiente comando:

\$ sudo apt-get update && sudo apt-get upgrade

Posterior a esto, cabe resaltar que la tecnología que utilizaremos será NodeJS para la conexión del escenario. Por lo tanto, descargamos la última versión de nodejs directamente desde la web oficial bajo el siguiente comando desde el terminal:

```
$ curl -sL https://deb.nodesource.com/setup_6.x |
sudo -E bash -
```

Una vez descargado el paquete realizamos la instalación en la Raspberry Pi con el comando:

```
$ sudo apt-get install nodejs
```

Adicional a lo anterior, se debe instalar ciertas librerías necesarias para el sensor DHT11, de igual manera necesitamos instalar el gestor de paquetes npm bajo el comando

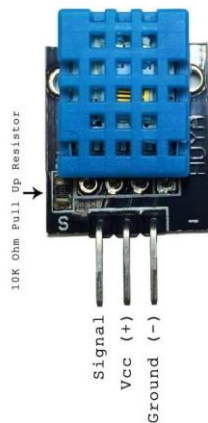
```
$ sudo npm install
```

VIII. ESQUEMA DE CONEXION DEL SENSOR A LA RASPBERRY PI

Para la conexión física del sensor DHT11 usaremos los pines del GPIO de la raspberry 1 (3v3),14 (Ground),16 (GPIO23) situados en la posición que se indica en la siguiente figura:



Mientras tanto el sensor DHT11 tiene 3 pines que son signal, Vcc (+) y Ground (-):



El esquema de conexión entre el sensor DHT11 y la raspberry PI quedaría entonces de la siguiente manera [4]:

- Pin 1 del DHT11 (Signal) conectado al Pin 16 (GPIO23) del GPIO de la Raspberry Pi
- Pin 2 del DHT11 (Vcc (+)) conectado al Pin 1 (3V3/Power) del GPIO de la Raspberry Pi
- Pin 3 del DHT11 (Ground (-)) conectado al Pin 14 (Ground) del GPIO de la Raspberry Pi

IX. INSTALACIÓN DE LOS COMPONENTES SOFTWARE PARA EL SENSOR DHT11

El sensor DHT11 necesita dos componentes software principales para funcionar.

1. Librería bcm2835
2. Paquete node-dht-sensor

Para la instalación del primero ejecutamos los siguientes comandos:

```
$ wget
http://www.airspayce.com/mikem/bcm2835/bcm2835
-1.46.tar.gz
$ tar zxvf bcm2835-1.46.tar.gz
$ cd bcm2835-1.46
$ ./configure
$ make
$ sudo make check
$ sudo make install
$ cd ..
```

Una vez instalado la librería bcm2835 se instala el paquete npm:

```
$ sudo npm install -g --unsafe-perm node-dht-sensor
```

Ahora se procede a conectar físicamente el sensor DHT11 a los pines del GPIO de la raspberry pi, posterior a esto creamos un archivo "sensor-text" con extensión .js:

```
//Funcion que permite la toma de datos del sensor DHT11
setInterval(function() {
//Requerimos las librerias para funcionamiento del sensor
var sensor = require('/usr/lib/node_modules/node-dht-
sensor');
//Utilizamos la funcion read de la librería del sensor
pasandole como parametro 11 por el tipo de sensor
"DHT11", y 23
//que es el valor del GPIO023 el pin 16 de la raspberry
sensor.read(11, 23, function(err, temperature, humidity) {
if (!err) {
//Si no hay errores en lectura de libreria y ejecución de la
función del sensor entonces imprime la temperatura
"temp"
//ambiente y el valor de la humedad "humidity"
console.log('temp: ', temperature.toFixed(1) + '°C,' +
'humidity: ', humidity.toFixed(1) + '%');
}
});
}, 2000);
// Finalmente ejecutamos la función automáticamente
tomando los valores del sensor cada 2 segundos.
```

```

1 //Funcion que permite la toma de datos del sensor DHT11
2 setInterval(function() {
3   //Requerimos las librerias para funcionamiento del sensor
4   var sensor = require('usr/lib/node_modules/node-dht-sensor');
5   //Utilizamos la funcion read de la libreria del sensor pasandole como parametro 11 por el tipo de sensor "DHT11", y 23
6   //que es el valor del GPIO el pin 18 de la raspberry
7   sensor.read(11, 23, function(err, temperature, humidity) {
8     if (!err) {
9       //Si no hay errores en lectura de libreria y ejecución de la función del sensor entonces imprime la temperatura "temp"
10      //ambiente y el valor de la humedad "humidity"
11      console.log('temp: ', temperature.toFixed(1) + '°C,' + 'humidity: ', humidity.toFixed(1) + '%');
12    }
13  });
14 }, 2000);
15 // Finalmente ejecutamos la función automáticamente tomando los valores del sensor cada 2 segundos
16

```

Lo importante del código es la función Read y los parámetros que se le pasan, como se observa se le ha pasado 11 por el tipo de sensor que estamos utilizando y 23 que es el valor del GPIO con pin 16 en la raspberry.

Finalmente, desde la terminal ejecutamos el comando:

\$ sudo node ./sensor-test.js

```

pi@raspberrypi:~ $ sudo node ./sensor-test.js
temp: 19.0°C, humidity: 57.0%
temp: 19.0°C, humidity: 57.0%
temp: 19.0°C, humidity: 58.0%
temp: 19.0°C, humidity: 58.0%
temp: 19.0°C, humidity: 58.0%
temp: 19.0°C, humidity: 58.0%
^C
pi@raspberrypi:~ $

```

X. REFERENCIAS

- [1] S. Electric, "IoT 2020 Business Report. The future of the Internet of Things: From sensor to business sense," 2016.
- [2] N. Bari, G. Mani, and S. Berkovich, "Internet of Things as a Methodological Concept," in *Computing for Geospatial Research and Application (COM.Geo)*, 2013 *Fourth International Conference on*, 2013, pp. 48-55.
- [3] Z. Yu and W. Tie-Ning, "Research on the Visualization of Equipment Support Based on the Technology of Internet of Things," in *Instrumentation, Measurement, Computer, Communication and Control (IMCCC)*, 2012 *Second International Conference on*, 2012, pp. 1352-1357.
- [4] Zorrilla Jose, <https://www.internetdelascosas.cl/2017/05/19/raspberry-pi-conectando-un-sensor-de-temperatura-y-humedad-dht11/>, Santiago de Chile, 2017