Universidad Nacional de La Matanza

Sistemas Operativos Avanzados

Trabajo Práctico
Sistemas embebidos – IoT
Informe Final

Integrantes:

Paris	Gustavo	33778800
Gerez	Brian	37481313
Vitelli	Donatella	38258961
Currao	Martin	38029678

Informe final

Contents

Sinopsis

Nuestro proyecto consiste en la medición de la temperatura y humedad del medio ambiente. Ésta se puede obtener por medio de un sensor o de internet. Dicha medición se ve reflejada en un display (LCD 16x2) y en dos leds uno rojo y otro azul. Cuando el sensor rojo tiene mayor intensidad que el azul es porque hace más calor y viceversa. Entonces si los dos leds tienen intensidades parecidas es porque el ambiente tiene una temperatura intermedia. Por otra parte, con un sensor de sonido detectamos aplausos para prender y apagar la luz del display. En la parte IoT contamos con un módulo Wifi y un servidor (Gateway) para hacer la conexión con la aplicación de Android. Desde la aplicación podemos obtener la temperatura y la humedad del medio ambiente por medio de dos opciones diferentes, el sensor o internet (cloud). Utilizamos además dos sensores de android: acelerómetro, para cambiar de fuente de medición sensor/internet y un sensor de proximidad para prender y apagar los leds.

Material utilizado

- Arduino UNO
- Modulo Wifi ESP8266

Sensores:

- Sensor de temperatura y humedad DHT11 analógico. Se mide en grados y porcentaje respectivamente.
- Sensor de sonido analógico. Se mide en dB (decibeles).

Actuadores:

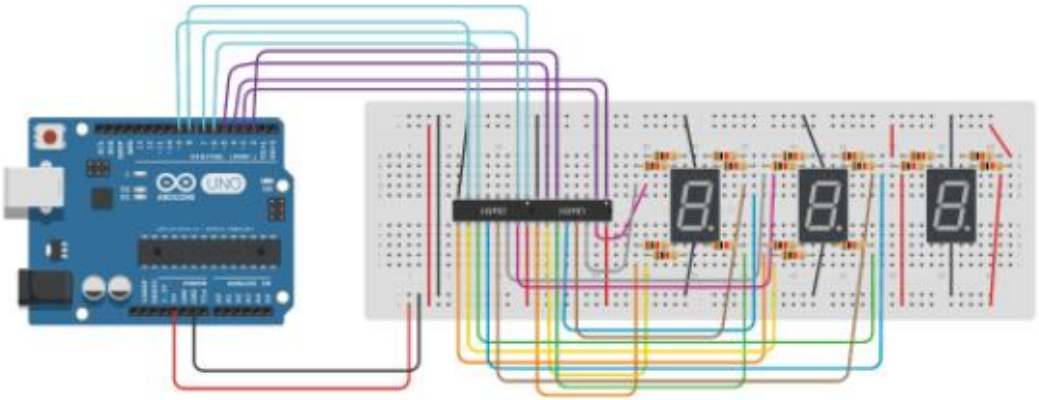
- Led azul (Analógico)
- Led rojo (Analógico)
- Display (Digital)

Entorno de desarrollo:

- Arduino IDE
- Visual Studio (Android)
- IntelliJ Idea (Servidor)

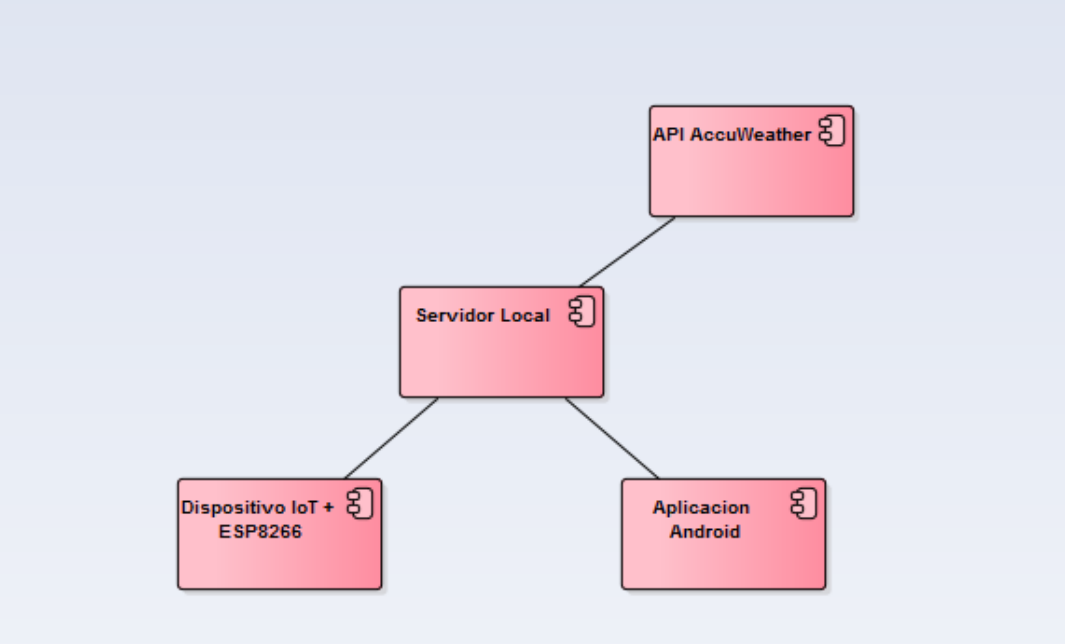
Decisiones tomadas y criterio aplicado

En un principio íbamos a utilizar dos siete segmentos como actuadores digitales para mostrar la temperatura, pero dado que cada siete segmentos necesitan siete entradas para mostrar los números del 0 al 9, determinamos usar dos buffers BCD-7 Segmentos, los cuales requieren solo 4 entradas cada uno y poseen las salidas suficientes para manejar un siete segmentos. El proyecto entonces se veía de la siguiente manera:



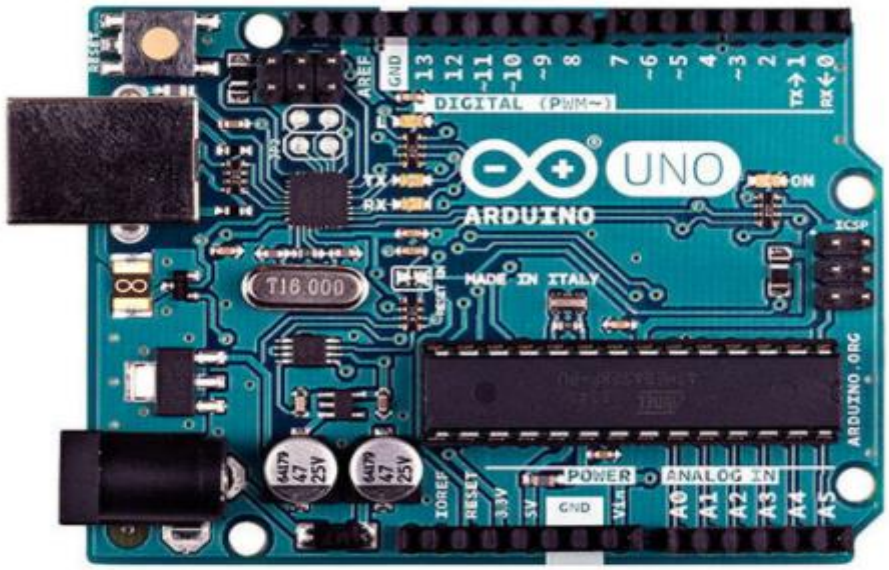
Aun así nos quedamos con pocas entradas/salidas digitales en el arduino dado que ocho estaban siendo usadas por estos dos buffers. Por eso decidimos usar un sensor de temperatura analógico para no utilizar estas salidas. El problema fue que el módulo wifi también requería de dos entradas digitales y los leds requerían las salidas PWM, de esta forma nos quedamos sin entradas para un posible sensor digital. Por lo tanto, cambiamos los siete segmentos por un display LCD 16x2. Éste nos permite mostrar dos líneas con 16 caracteres cada una utilizando tan solo 6 salidas digitales del arduino. Como el sensor de temperatura dejo de funcionar por motivos desconocidos decidimos invertir en un sensor digital que midiera temperatura y humedad, y como sensor analógico agregamos el de sonido.

Para resolver la parte de IoT consideramos mucho más prolijo el uso de un servidor (Gateway) que conecte el módulo wifi con la aplicación de android e internet a través de una conexión http, como se muestra en la siguiente figura:



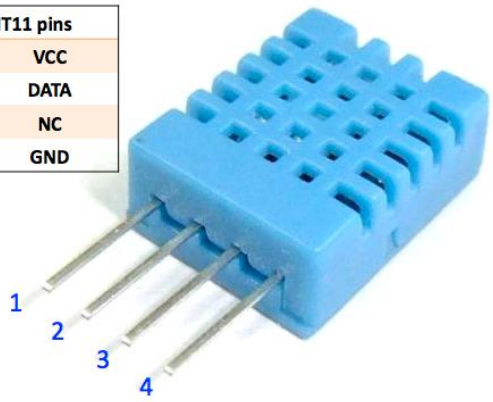
Descripción detallada del funcionamiento

El Arduino UNO posee una conexión serial para conectarse a la computadora e interactuar con el IDE, un botón de reset en la esquina superior izquierda para reiniciarlo, posee también pines analógicos y digitales, los que tienen una A delante son analógicos y los demás digitales. De estos últimos, los que tienen un símbolo delante son salidas PWM para simular salidas analógicas. Los pines 3.3V y 5V son las fuentes de energía o positivos y los pines GND son los de tierra o negativos. A estos pines es donde se conectan todos los sensores y actuadores que se verán más adelante.



El sensor de temperatura y humedad (DHT11) posee cuatro entradas, el primer pin se conecta a positivo y el cuarto a negativo, el segundo a uno de los pines analógicos del arduino (en este caso el A1), el tercero no se usa.

DHT11 pins	
1	VCC
2	DATA
3	NC
4	GND



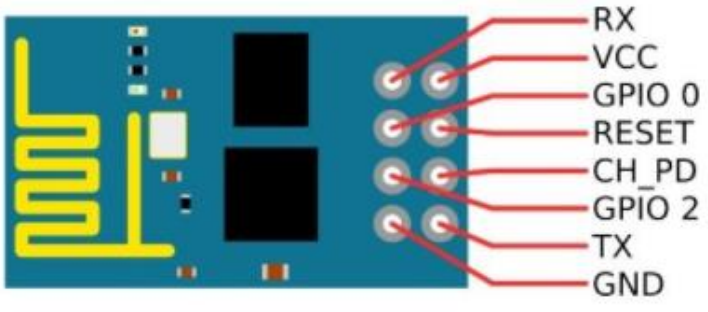
Para poder hacerlo funcionar tuvimos que agregar la librería "dht.h" y para tomar datos de él tuvimos que leer desde el pin en el cual está conectado al arduino.

El sensor de sonido posee cuatro entradas, la primera desde la izquierda es la analógica que va conectada al pin A0 del arduino en nuestro caso, la segunda va a negativo y la tercera a positivo. No usamos la cuarta que es para entrada digital. Para que el sensor funcione en este caso no fue necesario agregar una librería.
Este sensor posee un tornillo pequeño para calibrar la sensibilidad, junto con esa calibración más un umbral 822 seteado desde el arduino logramos que cuando el sensor capte un aplauso el display apague sus luces o encienda.

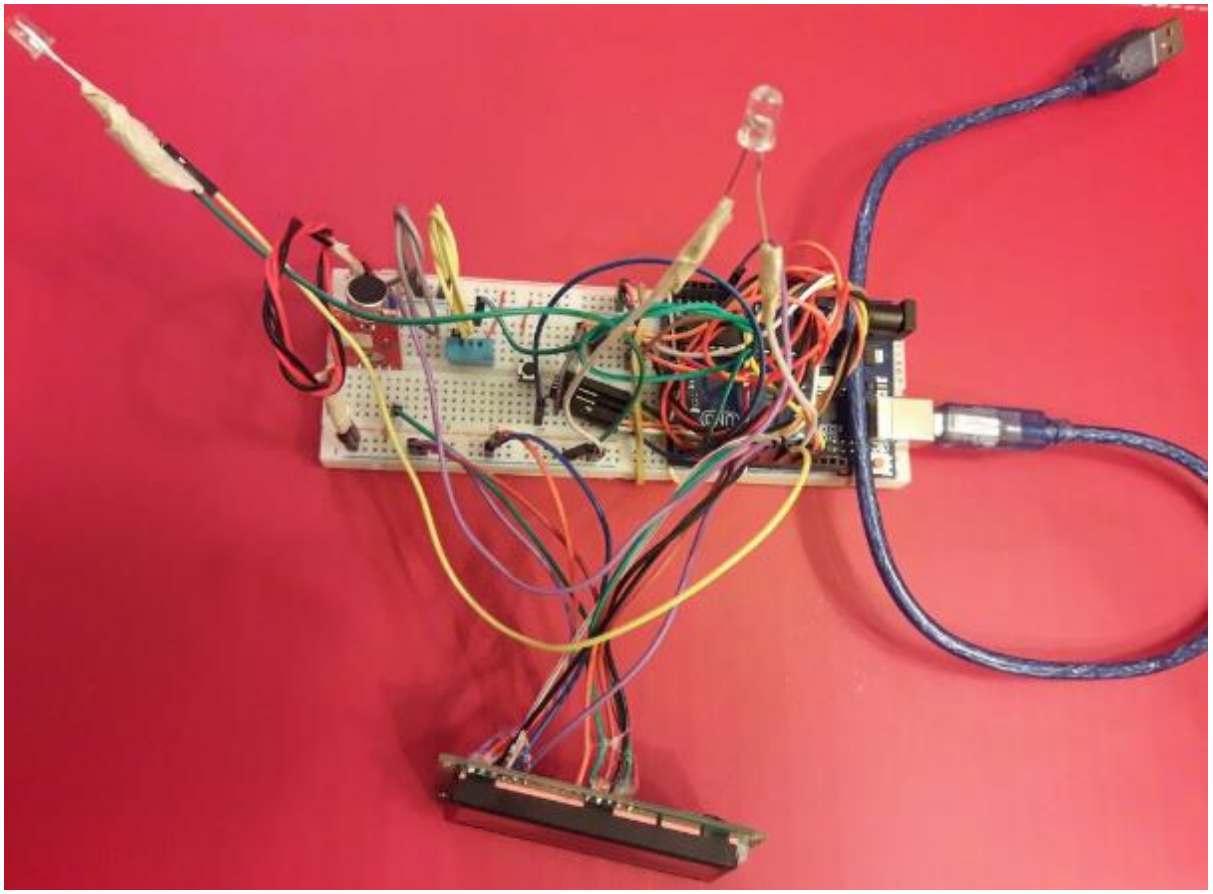
El display que utilizamos para mostrar la humedad y temperatura es un LCD de 16 x 2 (16 caracteres en dos líneas). Para hacerlo funcionar utilizamos la librería "liquidCrystal.h". De las 16 entradas usamos solamente 12. La entrada 1 va a ground o negativo, la 2 va a positivo, el 3 va a pwm o sea pin 6 (porque con eso manejamos el contraste del display por medio de la función analogWrite(número de pin, valor de contraste)), el 4 va al pin 8, el 5 al ground, el 6 al pin 7, el 11 al pin 5, el 12 al pin 4, el 13 al pin 3, el 14 al pin 2, el 15 a positivo con resistencia de 220 ohm, al pin 9 y el 16 a ground.

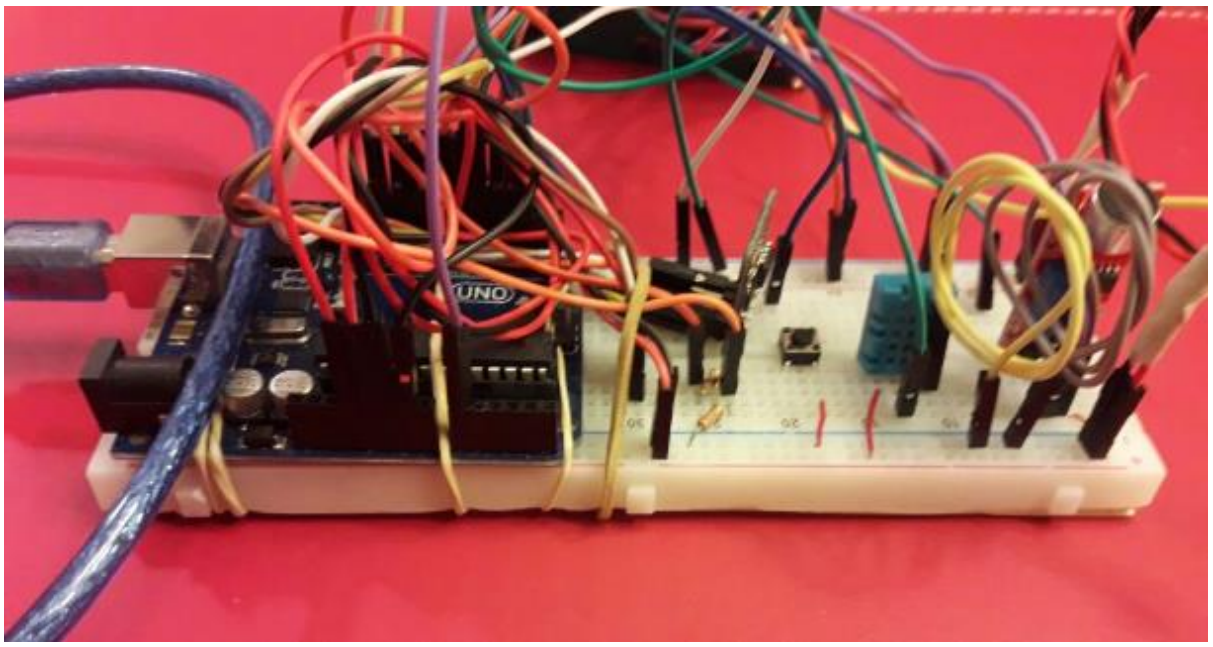
Finalmente conectamos dos leds uno azul y otro rojo a las salidas PWM del arduino para simular una salida analógica y poder ver distintas intensidades de los leds para los diferentes niveles de temperatura. El led rojo está conectado al pin 11 y el led azul al pin 10 del arduino.

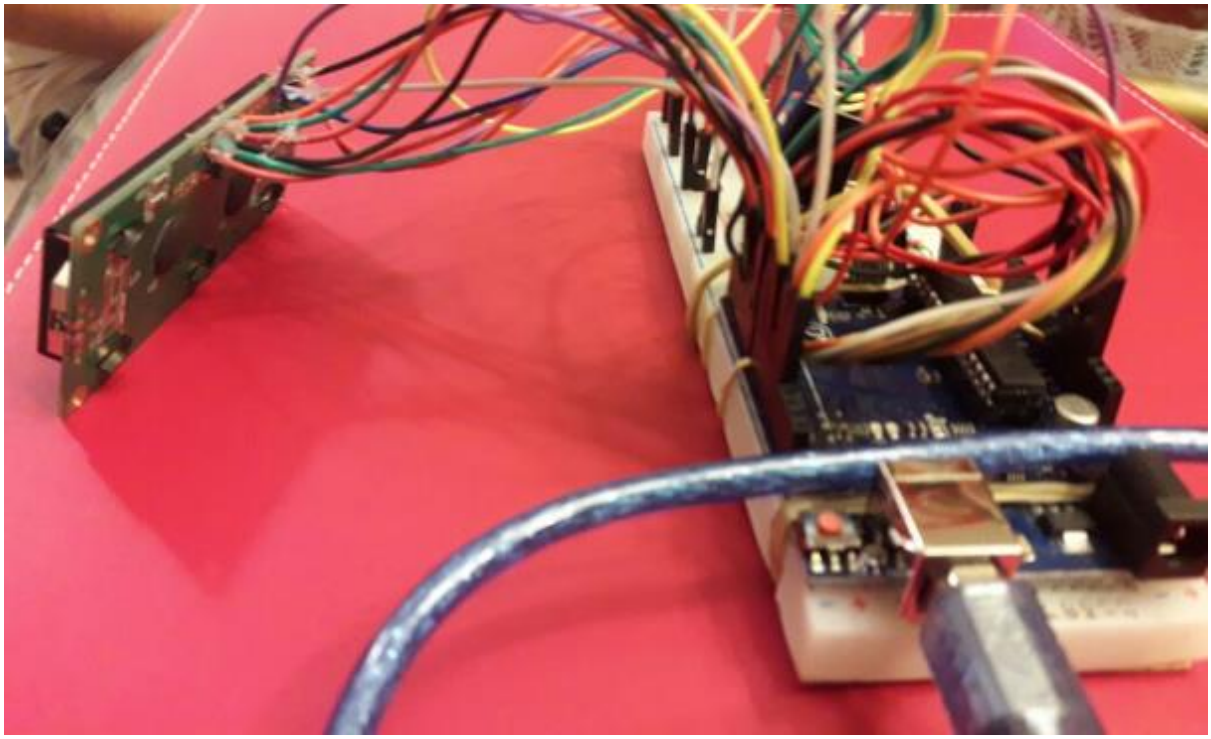
Tenemos además conectado al arduino un módulo Wifi ESP8266. Las entradas VCC y CH_PD van conectadas a 3.3v del arduino, GND va conectada a negativo, TX al pin 13 y RX al pin 12. Al RESET se le añadió un botón (pulsador) que está conectado a negativo para resetear el módulo.
La entrada GPIO-0 se utilizó para flashear el módulo ya que necesita de un flasheo para actualizar el firmware. Sin éste no respondía a los comandos AT. Para ello se buscó el firmware y el software necesario (NODEMCU). En este programa se setearon parámetros de velocidad de flash, tamaño del flash y los baudios (115200). Para iniciar el módulo en nuestro código debemos iniciarlo con 115200 baudios y luego pasarlo a 9600 debido a que el arduino trabaja a 9600 y no podremos observar las salidas de los comandos AT por el puerto serial.
Para que el arduino pueda comunicarse con el módulo wifi tuvimos que utilizar una librería que simula una conexión serial por software cuyo nombre es "SoftwareSerial.h".



Por ende la conexión de todos estos elementos juntos quedó de la siguiente manera:







Por la parte de conexión del módulo con la aplicación android usamos un servidor rest, con comunicación http. El módulo cuando inicia le pasa la IP al servidor que va a utilizar para las futuras request.
Cada un minuto el servidor se comunica con la API de accuWeather para actualizar las condiciones del clima, cada 2.5 segundos el servidor actualiza los datos del arduino que incluye enviarle la nueva posición de las luces, los nuevos datos de accuWeather y que opción mostrar según seleccionado en el android (temperatura sensor, temperatura internet).
La aplicación android le manda una request al servidor, con método GET y un endpoint para obtener la temperatura y humedad del sensor o bien de internet, y además para prender o apagar las luces de los leds del arduino.

Por la parte de android utilizamos dos sensores. El acelerómetro para cambiar de fuente de medición entre sensor e internet. Cada vez que el acelerómetro detecta un estímulo la aplicación cambia de cloud a sensor y viceversa. El otro sensor utilizado es el de proximidad para cambiar el estado de las luces, si detecta un estímulo y las luces están prendidas, estas se apagan y viceversa.

GitHub Repositorio Público

- https://github.com/gusparis/chobi_temp_soa

Bibliografía

- https://stackoverflow.com/questions/33594832/how-to-get-the-esp8266-to-work-in-update-mode
- http://www.instructables.com/id/ESP8266-WiFi-Module-for-Dummies/
- https://www.digikey.com/en/resources/conversion-calculators/conversion-calculator-resistor-color-code-4-band
- http://www.instructables.com/id/The-Simple-Guide-to-Flashing-Your-ESP8266-Firmware/
- http://fab.cba.mit.edu/classes/865.15/people/dan.chen/esp8266/
- https://room-15.github.io/blog/2015/03/26/esp8266-at-command-reference/
- http://apidev.accuweather.com/developers/samples
- https://www.arduino.cc/en/Tutorial/HelloWorld
- http://panamahitek.com/dht11-sensor-de-humedadtemperatura-para-arduino/