



Universidad Nacional de La Matanza
Departamento de Ingeniería e Investigaciones Tecnológicas

Sistemas Operativos Avanzados
Arduino + IOT
Sleep apnea

Profesores:

- Graciela de Luca
- Esteban Carnuccio
- Gerardo García
- Waldo Valiente
- Mariano Volker

Integrantes:

Apellido y Nombre	DNI
Almaraz Bárbara Ailen	38.256.069
Culen, Fernando	35.229.859
Juffar, Sebastian	34.497.148
Rodriguez, Gabriel	36.822.462
Zimarino, Guillermo Tomás	37.541.442

Contenido

Introducción:.....	3
¿Qué es la apnea del sueño?	3
Objetivo:	3
Hardware utilizado:	4
Características de los componentes:	4
Tareas realizadas:	10
ARDUINO.....	10
ANDROID.....	11
Sensores del teléfono utilizados	12
Problemas encontrados:	12

Introducción:

¿Qué es la apnea del sueño?

La apnea del sueño es un trastorno común en el que la persona que lo sufre hace una o más pausas en la respiración o tiene respiraciones superficiales durante el sueño.

Las pausas pueden durar entre unos pocos segundos y varios minutos. Por lo general, la respiración vuelve a la normalidad, a veces con un ronquido fuerte o con un sonido parecido al que una persona hace cuando se atraganta.

Por esta razón, el sueño es de mala calidad y se siente cansancio durante el día. La apnea del sueño es una de las principales razones por las cuales una persona puede sentir mucho sueño durante el día.

A menudo la apnea del sueño se queda sin diagnosticar. Por lo general, los médicos no pueden detectarla durante las consultas corrientes. Además, no hay pruebas de sangre para este problema. La mayoría de las personas que sufren apnea del sueño no saben que la tienen porque solo ocurre durante el sueño. El primero en notar los signos de la apnea del sueño puede ser un familiar o quien duerme en la misma cama con la persona afectada.

La apnea del sueño que no recibe tratamiento puede:

- Aumentar el riesgo de sufrir presión arterial alta, ataque cardíaco, accidente cerebrovascular (derrame cerebral), obesidad y diabetes
- Aumentar el riesgo de que ocurra insuficiencia cardíaca o de que ésta empeore
- Aumentar las probabilidades de que haya arritmias o latidos irregulares del corazón
- Aumentar las probabilidades de sufrir accidentes automovilísticos o de trabajo

La apnea del sueño es un problema crónico de salud que requiere un tratamiento a largo plazo. Los cambios en el estilo de vida, los dispositivos bucales, la cirugía o los dispositivos respiratorios pueden tener éxito en el tratamiento de la apnea del sueño en muchas personas.

Objetivo:

Realizar un sistema de control para personas que sufren esta enfermedad, analizando diversos parámetros mientras el usuario duerme. Este sistema tendrá un sistema embebido compuesto por un módulo bluetooth para enviar la información entre la aplicación y la placa arduino. Este proyecto busca controlar la estabilidad del sueño, basándose en mediciones sobre las pulsaciones, nivel de respiración y temperatura corporal.

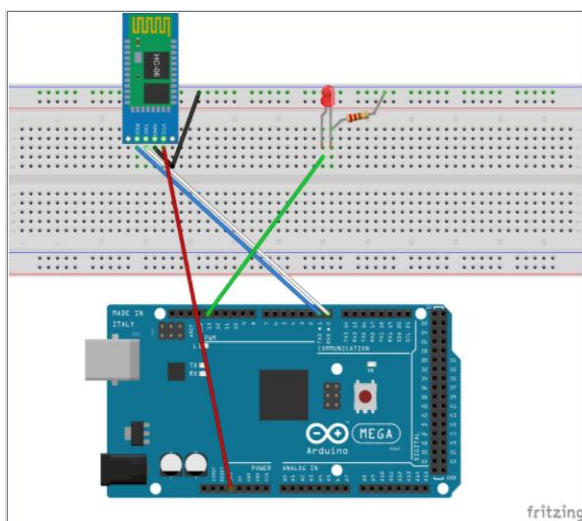
En base a los parámetros recibidos la placa tomará decisiones como: Si el nivel de respiración baja, prender un ventilador (simulado a escala con un cooler) o si las pulsaciones no se estabilizan encender una alarma (buzzer) para despertar al usuario. Además el usuario podrá enviar un mail a su médico con el informe obtenido para agilizar los análisis sobre el mismo.

Hardware utilizado:

- 1 Placa Arduino UNO
- 1 Modulo bluetooth HC-05(Maestro-Esclavo)
- 1 LED's
- 1 Protoboard 400 puntos
- 1 Sensor de temperatura DS18B20 (sensor arduino)
- 1 Sensor de pulso cardiaco (sensor arduino)
- Cables macho – macho/ Macho-Hembra
- 1 Sensor de potenciómetro (sensor Arduino)
- 1 Resistencias 4k7
- 1 Buzzer (actuador Arduino)
- 1 Cooler (actuador Arduino)

Características de los componentes:

Conexión Modulo Bluetooth HC-05: (como esclavo)



El módulo de bluetooth HC-05 dispone de 6 pines y puede actuar como maestro-esclavo. En la imagen anterior se muestra la conexión básica para el funcionamiento del módulo.

Características del módulo Bluetooth HC05:

- Fácil de usar y completamente encapsulado
- Chipset: CSR
- Bluetooth V2.0
- Alimentación: 3.6V a 6V
- Niveles lógicos: 3.3V
- Velocidad de transmisión: 1200, 2400, 4800, 9600, 19200, 38400, 57600 y 115200 . Seteable por comandos AT
- Configuración predeterminada del puerto serie: 38400, N, 8, 1

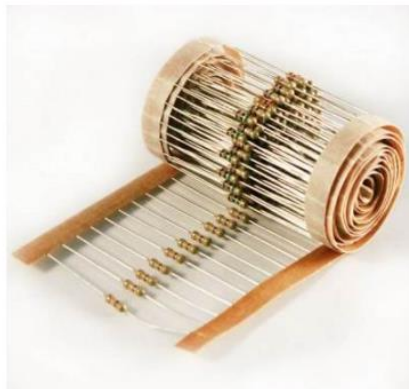
Hacer la comunicación entre la PC y el módulo de forma indirecta a través de un Arduino:



Resistencias 4k7:

0	0	x 1	
1	1	x 10	± 1%
2	2	x 100	± 2%
3	3	x 1000	± 5%
4	4	x 10.000	± 10%
5	5	x 500.000	
6	6	x 1.000.000	
7	7	x 10.000.000	
8	8	10%	
9	9	100%	

TOLERANCIA



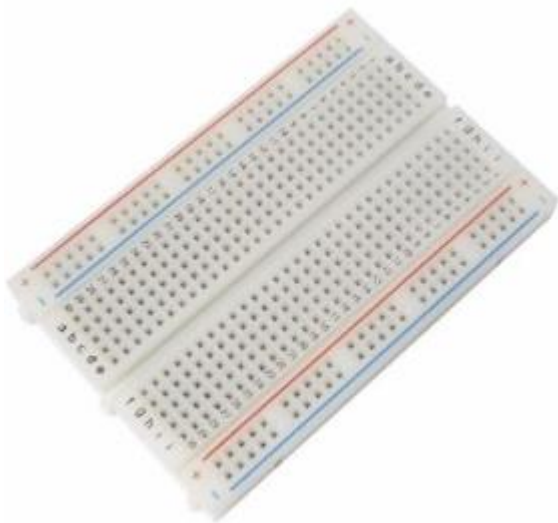
Resistencias para controlar la cantidad de corriente. Se oponen al paso de la corriente, en nuestro caso utilizamos las de 4k7

Cooler:



Es un ventilador utilizado en los gabinetes de computadoras y otros dispositivos electrónicos para refrigerarlos. Este dispositivo normalmente saca el aire caliente desde el interior. Simularemos un ventilador a escala con este Cooler.

Protoboard 400 puntos:



Características:

- ✓ Consta de pistas para colocar conexiones de componentes electrónicos como integrados, resistencias.
- ✓ Al protoboard se conoce en español como "placas de prototipos"
- ✓ La tableta sirve para experimentar con circuitos electrónicos.

Leds alta luminosidad:

Voltaje: 3.0 ~ 3.2V

Corriente: 20mA

Longitud de onda: 11000-13000

Sensor DS18B20



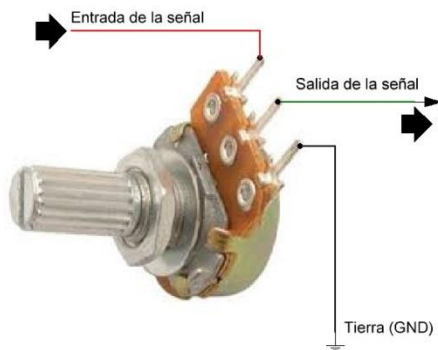
El sensor DS18B20 permite medir temperaturas de hasta 125°C de forma fácil y además está sellado en un envoltorio estanco que permite sumergirlo en un líquido o protegerlo de la intemperie. Dado que es un sensor digital, la señal leída no se degrada debido a la distancia del cableado. Con este sensor mediremos la temperatura corporal del usuario.

Sensor de pulsaciones



El sensor funciona con un sensor de ritmo cardiaco óptico, una etapa de amplificación y un filtro para el ruido, lo cual hace que su señal de salida sea confiable y estable. El consumo de corriente es bajo siendo de 4mA con una alimentación de 5V. Esta basado en un LED emisor y un sensor receptor de intensidad, la cantidad de luz reflejada por el dedo cuando hay paso de corriente sanguínea define la salida del sensor.

Potenciómetro



Un **potenciómetro** es uno de los dos usos que posee la resistencia o resistor variable mecánica (con cursor y de al menos tres terminales). ... Según la potencia que disipe en su funcionamiento, como regulador de tensión, así debe ser la potencia de la resistencia variable mecánica a utilizar.

Buzzer



A través del Arduino u otros controladores, este módulo será capaz de controlar el sonido de la alarma o la música MID fácilmente. Se extiende con los sensores de placa Arduino utilizados en combinación para lograr el control de un sonido interactivo y obras de luz.

Configuración de pines del módulo:

- 1 Salida,
- 2 Fuente de alimentación
- 3 GND.

Módulo de conexión de interfaz para PH2.0.

Características:

- Tipo: Digital.
- Fuente de alimentación: 5VDC.

Tareas realizadas:

ARDUINO

1. Realizamos la conexión al módulo de bluetooth para realizar el intercambio de mensajes entre la aplicación y la placa Arduino. La distancia máxima de medición de este módulo es de 10m.
2. Definición de pines

```
#Define potenciómetro A0
#define buzzer 10 //era 8
#define ventilador 7
#define led 11 // era 9
#define termómetro 2
#define sensorPulsaciones A1
#define LED13 13
```

3. Se importan las bibliotecas en Arduino OneWire.h y DallasTemperature.h
4. Para el BT se incluye <softwareSerial.h> y <Altsoftserial.h>
5. Definición de tiempos

```
long tiempoAnterior;
long tiempoAnteriorLectura;
long tiempoAnteriorAlarma;
long tiempoInicioAlarma = 0;
long tiempoPulsaciones;
long tiempoEsperaBuzzer;
long tiempoCambioIntensidad;
long tiempoCalibracion;
long marcaTiempoRespira;
long muestrasTiempoRespiracion[cantidadMuestrasRespiracion];
```

- ```

long mediaTiempoRespiracion;
long desvioTiempoRespiracion;
long intervaloInferiorMediaRespiracion;
long intervaloSuperiorMediaRespiracion;

```
6. Info útil : Si se confirmó la conexión y todavía no estaba durmiendo, empieza la fase de sueño e inicializa la calibración. Si no, ignora. Si se confirmó la conexión y estaba durmiendo, termina la fase de sueño. Si no, ignora. Si estuvo 17 segundos en modo alarma, entra en modo emergencia y envía el mensaje por bt
  7. CalibraSensorRespiracion() -- calibra el sensor de respiración con la media de tiempo entre de inhalar y exhalar, se toman x muestras en y minutos.
  8. Cuando se lee del pote se llama a CalculaDatosEstadisticos, que basándose en la media, en el desvío y en un intervalo de confianza.
  9. Verifica la respiración comparando con los parámetros obtenidos en la calibración
  10. Se verifica que el tiempo entre Inhalar y Exhalar este dentro del intervalo de confianza calculado para la media muestral.
  11. Luego de un X tiempo que la persona no respira, se prende la alarma (variando la intensidad) y el cooler (simula un ventilador).

## ANDROID

1. En el [AndroidManifest.xml](#) : se agregaron diversos permisos, para el bluetooth, para la conexión con internet, para la red, para el storage, entre otros.
2. Clases .JAVA

|                        |
|------------------------|
| ConexionBluetooth.java |
| Control.java           |
| DatosSensores.java     |
| DeviceList.java        |
| ECG.java               |
| FoundDeviceList.java   |
| Grafico.java           |
| Inicio.java            |
| Mail.java              |
| Reporte.java           |

3. Se utiliza un intent para buscar dispositivos Bluetooth.
4. Tenemos un thread que nos permite leer la data del bluetooth

5. Utilizamos otro thread cuando inicia el sensor de pulsaciones
6. Utilizamos intents para el manejo entre activities.
7. Las clases que necesitamos para enviar un correo con adjuntos con JavaMail son :  
Session y Transport y para la conexión con gmail y envío del mensaje:  
MimeMessage, MimeMultiPart y MimeBodyPart Para construir el mensaje.

## Sensores del teléfono utilizados

1. GPS para enviar vía mail la ubicación del usuario
2. Acelerómetro
3. Luminosidad

## Problemas encontrados:

- 1- Para graficar el ECG en tiempo real
- 2- Para enviar mail con adjunto
- 3- Acomodar los componentes en las activities
- 4- Enviar información mediante Bluetooth.
- 5- Calibración del potenciómetro. No pudimos implementarlo en una faja ya que las medidas que se obtenían al calibrarlo eran muy difíciles de procesar.
- 6- Inconveniente en la soldadura del sensor de pulsaciones, debimos soldarlo nosotros.
- 7- Obtener temperatura; teníamos un termómetro analógico del cual no podíamos implementar la formula para obtener los datos, por lo tanto, utilizamos uno digital prestado por lo profesores. Como tiene mucha inercia, no podíamos utilizarlo como proponíamos en un principio (detectar con el aliento), y terminamos utilizando el digital midiendo la temperatura en una zona corporal (axilas).
- 8- Al utilizar el GPS no logramos enviar la localización, no registra cambios.