

Universidad Nacional de La Matanza

Departamento de Ingeniería e Investigaciones Tecnológicas

**Internet of Things**

Sistemas embebidos y Android

“Vaso Inteligente”

**Integrantes:**

|  |  |
| --- | --- |
| **Apellido y nombre** | **DNI** |
| Aranda, Luciana Florencia |  |
| Carnovale, Pablo |  |
| Pérez, Luis |  |
| Pichetti, Gonzalo | 38.347.235 |

Introducción

El presente informe tiene como objetivo detallar el funcionamiento del proyecto Vaso Inteligente desarrollado para la catedra de la materia Sistemas Operativos Avanzados de la Universidad Nacional de La Matanza.

Objetivo

El proyecto tiene como objetivo predecir la cantidad de miligramos de alcohol por litro en sangre que una persona sumara a su cuerpo al ingerir una determinada bebida alcohólica contenida en un vaso.

Para ello se utilizan un sensor de alcohol que mide el gas etanol emitido por la bebida y un sensor de distancia para calcular el volumen lleno del vaso. Además, mediante un sensor de temperatura permite obtener la temperatura del líquido.

Descripción detallada del funcionamiento

**Arduino**

Inicialmente una persona ingresa un volumen de líquido con contenido alcohólico dentro del vaso inteligente y coloca la tapa. Esta tapa contiene 2 sensores, un sensor de alcohol/etanol (MQ3) y un sensor ultrasónico (HC-SR04).

Un transmisor/receptor bluetooth (HC-06) recibe el peso y el género de la persona.

Automáticamente el dispositivo a través del sensor del alcohol mide el porcentaje de etanol contenido en el aire emitido por la bebida y a través del sensor ultrasónico mide la distancia que hay desde el líquido a la tapa.

Obtenidos estos valores el sistema embebido calcula a partir de las fórmulas explicadas a continuación la cantidad de miligramos de alcohol en sangre que la persona sumara a su cuerpo:

**Cálculo de volumen del líquido:**

Se explica en un apartado posterior todo el procedimiento para llegar a esta ecuación.

***Volumen del líquido =******Altura del líquido*** *\** ***Área del vaso***

**Cálculo de cantidad de miligramos por litro de sangre:**

Inicialmente se obtuvo la siguiente ecuación lineal para calcular el volumen de alcohol contenido en el líquido dentro del vaso. Esta ecuación se explica en un apartado posterior.

***Valor del sensor:*** Obtenido del sensor directamente (Promedio de 10 mediciones)

***Volumen de Alcohol:***(Valor del sensor - 80) \* (0.27) (%)

***Densidad de alcohol:*** 0.79 gr/ml

***Gramos de alcohol en un volumen:*** (Volumen del líquido \* Volumen de alcohol \* 0.79)/100 (gr)

**Alcoholemia en Mujer: (*Gramos de alcohol en un volumen* / Peso de la Persona \* 0.7) (mg/L)**

**Alcoholemia en Hombre: (*Gramos de alcohol en un volumen* / Peso de la Persona \* 0.6) (mg/L)**

**Cálculo de la temperatura del líquido:**

La temperatura del líquido es obtenida directamente del controlador del sensor sin generar cálculos intermedios.

**Resultado**

Una vez obtenidos estos valores, el sistema muestra en un display LCD el grado de alcoholemia calculado dependiendo del género elegido y el peso indicado.

**Nivel de PWM**

El ancho de pulso del PWM está relacionado con la temperatura de manera lineal. Este nivel se utilizará para encender un led con una intensidad acorde a la temperatura del líquido.

Además, se muestran a continuación en el display LCD la temperatura del líquido calculado.

Luego se envía mediante el transmisor/receptor Bluetooth (HC-06) el resultado al dispositivo que envió el peso y el género.

**Descripción del código sistema embebido**

En este apartado, se explicará brevemente las funciones utilizadas en el código fuente del sistema embebido.

* **getStringDelimitar**: Esta función se encarga de dividir (parsear) los datos enviados desde el dispositivo Android.
* **volumenLiquido**: Esta función realiza el cálculo del volumen del líquido, en base a los datos obtenidos por el sensor ultrasónico.
* **volumenAlcohol**: Esta función realiza el cálculo del volumen de alcohol que posee el líquido contenido en el vaso, en base a un promedio de 10 mediciones obtenidas del sensor de gases y alcohol.
* **temperaturaLiquido**: Esta función devuelve el valor brindado por el sensor de temperatura.
* **setup**: Dentro de esta función se realiza la inicialización del módulo Bluetooth, display, sensor de temperatura y sensor de distancia. También se encuentra la configuración de los pines de entrada y salida.
* **loop**: Se encuentra la lógica principal del funcionamiento, donde utiliza todas las funciones antes mencionadas.

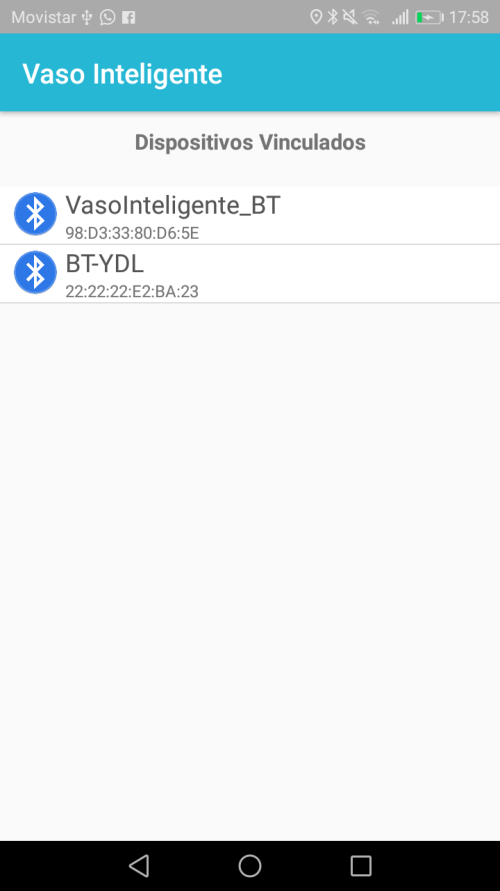
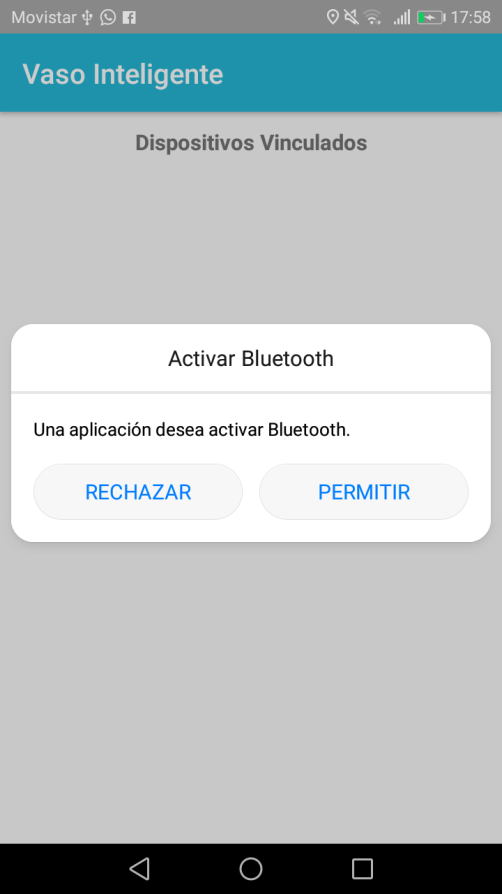
**Android**

**SplashActivity:** Esta activity es la bienvenida al sistema.

**MainActivity:** Esta activity realiza la conexión con el dispositivo bluetooth. Si el dispositivo esta deshabilitado solicita habilitación mediante un cuadro de dialogo. Una vez seleccionado el dispositivo, ésta envía a la siguiente activity la dirección MAC del dispositivo.

**ParamActivity:** Esta activity solicita al usuario el peso, estatura y género. A su vez abre un socket para enviarle al dispositivo bluetooth del sistema embebido los datos cargados por el usuario y mediante un manejador espera la respuesta del mismo. Una vez que el dispositivo responde, dispara una nueva activity que muestra los resultados.

**ResultActivity:** Esta activity muestra los resultados generados por el sistema embebido. Al hacer click sobre el botón “Salir” vuelve a comenzar el proceso desde **SplashActivity**.



Main Activity

Main Activity

Splash Activity

**Sensores utilizados Android**

ResultActivity

ctivity

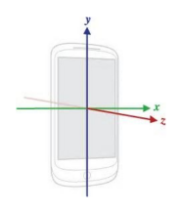
ParamActivity

Para trabajar con los sensores, utilizamos el Sensor Framework. Éste nos brinca las siguientes clases:

* SensorManager
* Sensor
* SensorEvent
* SensorEventListener

**Sensor Acelerómetro**

La pantalla principal (paramActivity) se encuentra sujeta a los cambios que se detecten en el sensor, permitiendo que, al detectar cierto movimiento muestre en el display de Arduino un mensaje con la leyenda “Shake”.



**Sensor Proximidad**

La pantalla principal (paramActivity) se encuentra sujeta a los cambios que se detecten en el sensor, permitiendo que, al detectar cierta proximidad sobre el dispositivo móvil muestre en el display de Arduino un mensaje con la leyenda “Proximidad”.



**Sensor Luz**

La pantalla principal (paramActivity) se encuentra sujeta a los cambios que se detecten en el sensor, permitiendo que, al detectar cierta disminución de luz muestre en el display de Arduino un mensaje con la leyenda “Luminosidad”.

Diagrama de Conexiones

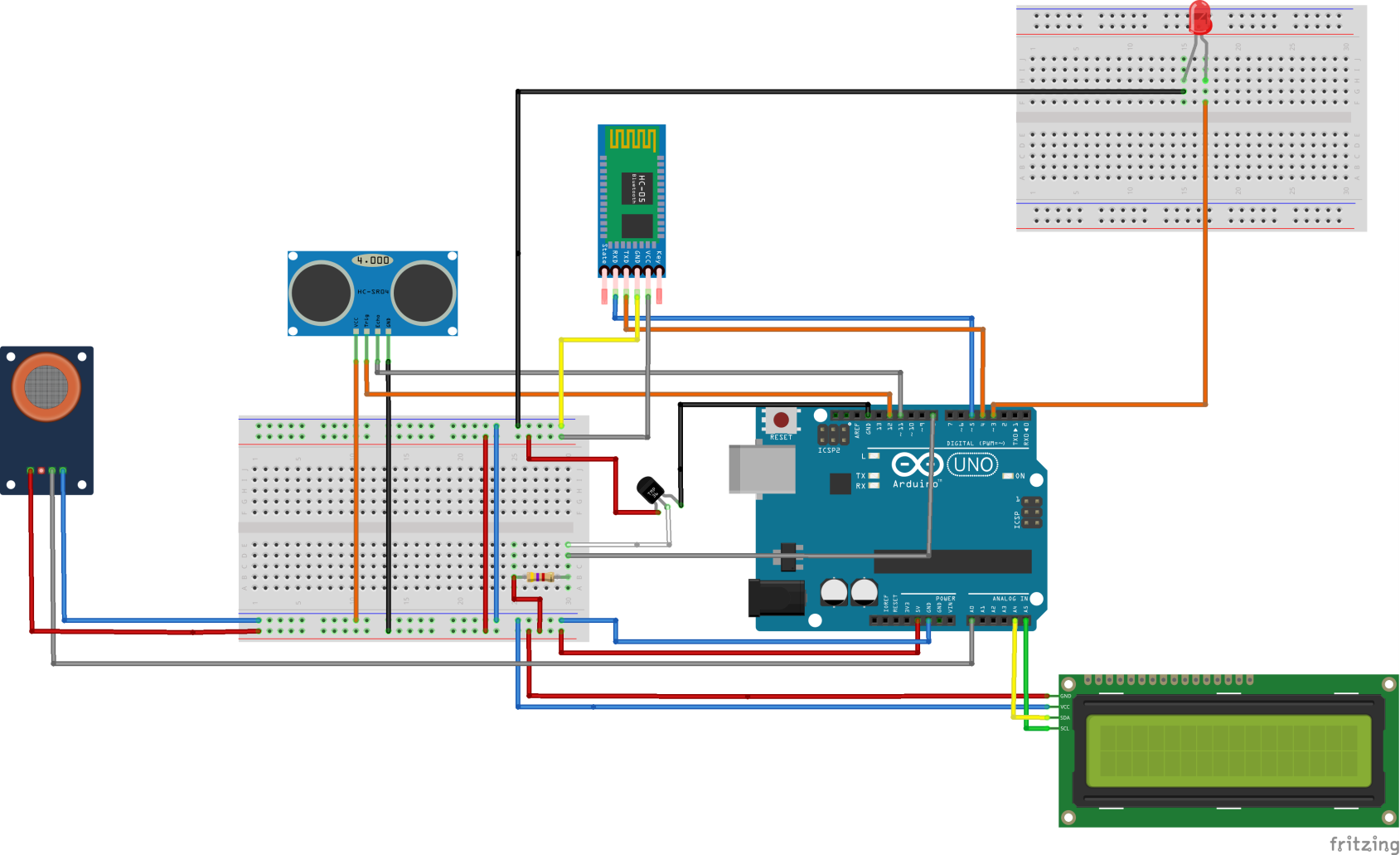


Diagrama de Componentes

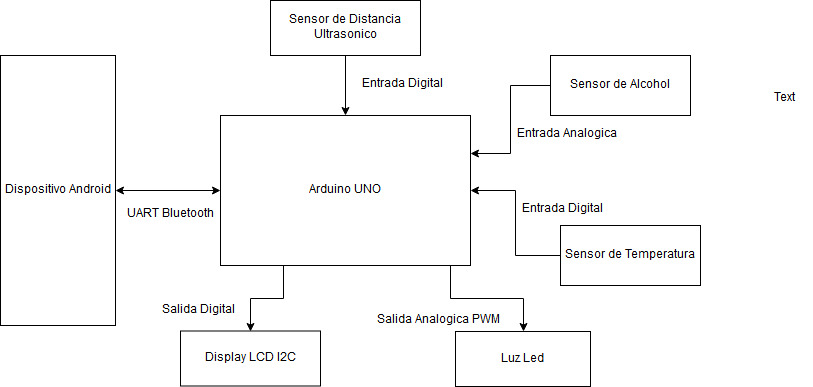


Diagrama en bloques Funcional

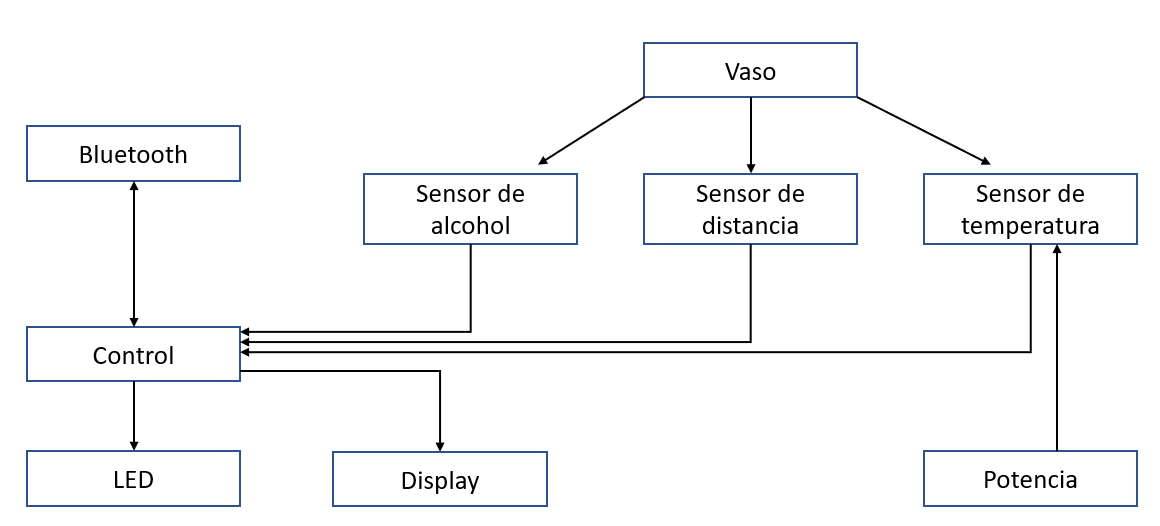


Diagrama en bloques Físico

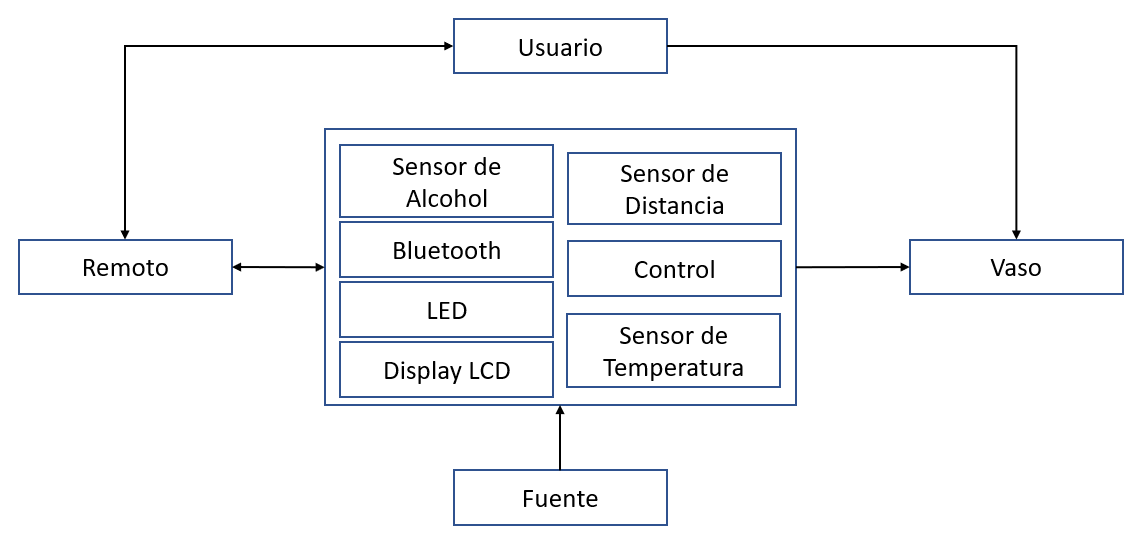
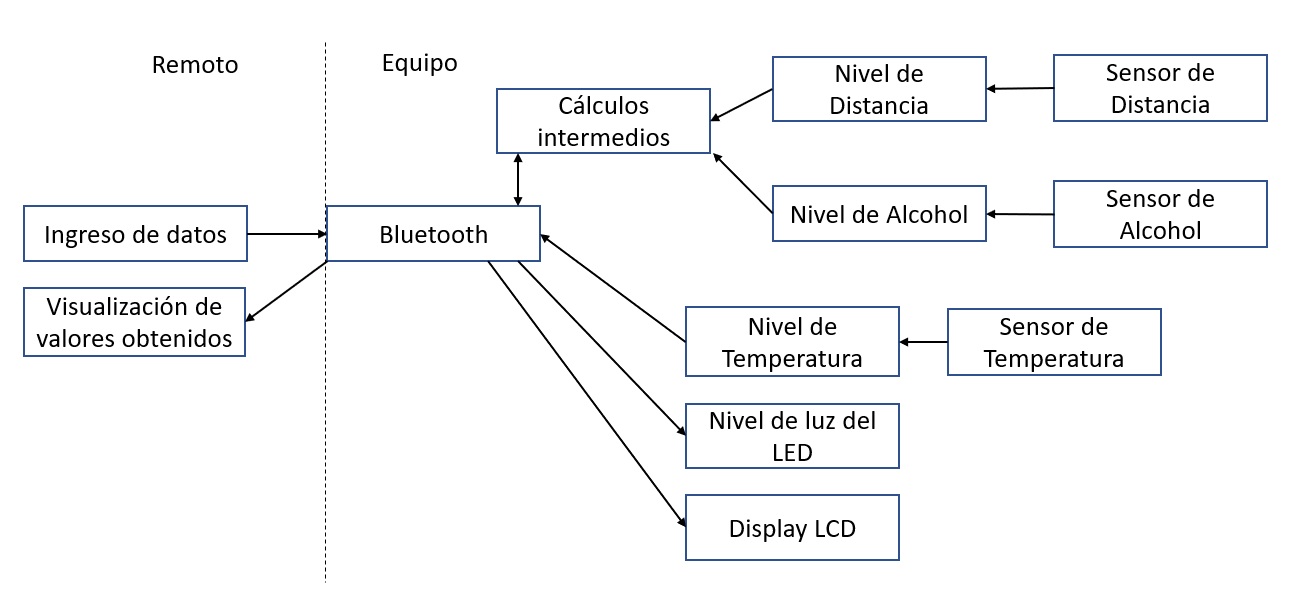


Diagrama en bloques Lógico



Detalle de los componentes utilizados

**Sensor de Alcohol MQ-3:**

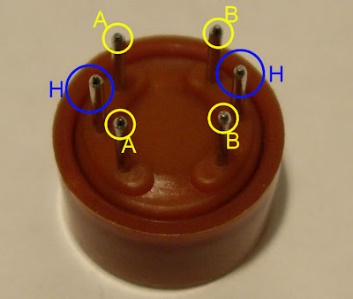
Este sensor detecta la concentración de alcohol en aire. Simplemente se conecta a una entrada analógica de un microcontrolador como Arduino y podremos medir la concentración de alcohol.

**Características**

* Alimentación: 5 Vdc
* Integrado amplificador [LM393](http://www.ti.com/lit/ds/symlink/lm393-n.pdf) con ganancia variable mediante potenciómetro.
* 2 pines de salida (salida analógica y salida de nivel TTL).
* Salida de nivel TTL válida de bajo nivel, se puede conectar directamente al microcontrolador.
* Salida analógica de 0 ~ 5 V , el voltaje más alto equivale a una concentración más alta.
* Condiciones de trabajo: Temperatura ambiente: -10°C a 65°C Humedad: ≤ 95% RH
* Condiciones estándar de trabajo: Temperaturas 20°C ± 2°C Humedad 65%±5% RH
* Detecta concentraciones que van de los 0.05ml/L a los 10mg/L de alcohol.
* Para que funcione correctamente con el alcohol se recomienda una RL o resistencia de carga de 200 K Ohms (En el caso de no contar con el LM393)

**Funcionamiento**

Dentro del sensor, se encuentra un pequeño tubo. Este microtubo cerámico está compuesto por oxido de aluminio y una capa sensible de dióxido de estaño. Interiormente contiene una bobina que calienta este microtubo para su correcto funcionamiento ocasionando mayor sensibilidad para la detección de gases.

Básicamente, tiene 6 pines, la tapa y el cuerpo. Aunque a pesar de que tiene 6 pines, se suelen utilizar sólo 4 de ellos. Dos de ellos son para el sistema de calefacción, lo que marco como H y los otros 2 son para la conexión de alimentación y de tierra, que están marcados como A y B.

Al entrar en contacto con el Etanol (aunque también detecta otros gases como el Benceno) el sensor varía la resistencia del circuito aumentando o disminuyendo y esto permite detectar la presencia del gas.



**Sensor de Distancia HC-SR04:**

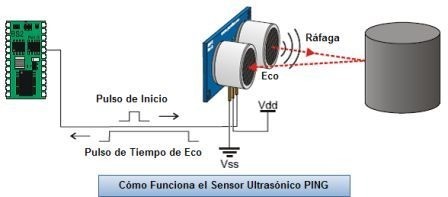
Este sensor posee un emisor y un receptor de ultrasonidos que trabajan a una frecuencia de 40KHz (una frecuencia inaudible para las personas).

**Características:**

**Forma de conexión**

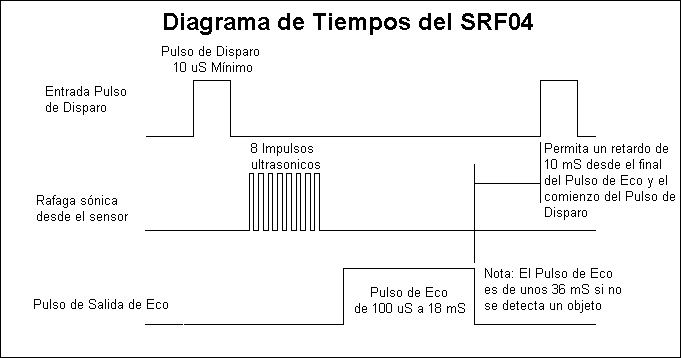
Este sensor consta de 4 pines, alimentación (Vcc), un disparador (Trig), el receptor (Echo) y masa (GND). Para el correcto funcionamiento del sensor es necesario el conectar estos 4 pines.

**Funcionamiento:**



El sensor SRF04 funciona emitiendo impulsos de ultrasonidos inaudibles para el oído humano. Los impulsos emitidos viajan a la velocidad del sonido hasta alcanzar un objeto, entonces el sonido es reflejado y captado de nuevo por el receptor de ultrasonidos.

Luego de emitir un pulso de inicio de 10us, lo que hace el controlador incorporado es emitir una ráfaga de 8 impulsos y a continuación comienza a contar el tiempo que tarda en llegar el eco. Este tiempo se traduce en un pulso de eco de anchura proporcional a la distancia a la que se encuentra el objeto.

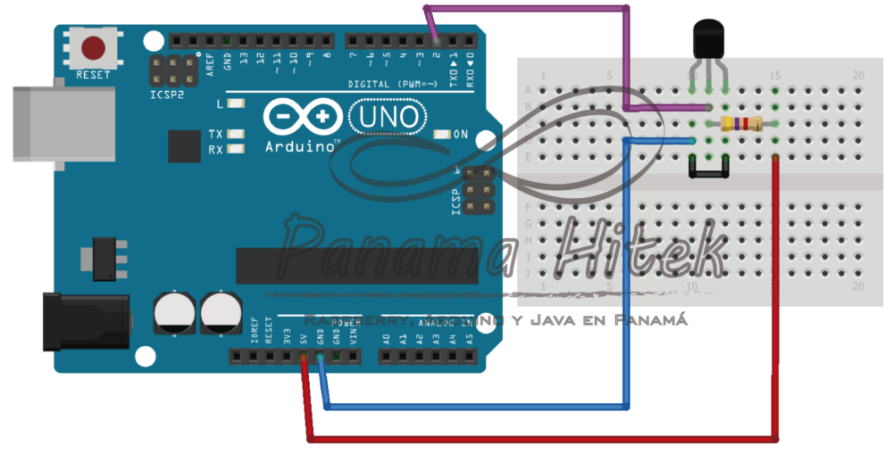


Luego mediante la ecuación D = V \* T sabiendo que el pulso viaja a la velocidad del sonido (340m/s) puede obtenerse la distancia del objeto al sensor.

**Sensor de Temperatura DS18B20:**

El sensor de temperatura DS18B20 es un dispositivo que se comunica de forma digital. Cuenta con tres terminales, los dos de alimentación y el pin “data”.

Utiliza la comunicación OneWire, que se trata de un protocolo especial que permite enviar y recibir datos utilizando un solo cable, a diferencia de la mayoría de los protocolos que requiere dos vías.

**Forma de conexión:**

**Características:**

* Es un termómetro digital de alta precisión, entre 9 y 12 bits de temperatura en grados Celsius (el usuario puede escoger la precisión deseada).
* Su temperatura operativa se encuentra entre -50 y 125 grados Celsius. La precisión, en el rango comprendido entre -10 y 85 grados es de ±0.5 grados.
* Para temperaturas entre -10ºC y 85ºC podemos tener ±0,5ºC de error. Para el resto de las temperaturas entre -55ºC y 125ºC el error es de ± 2ºC.
* Su precio es económico, su interfaz de funcionamiento es sencilla y su uso es muy provechoso para proyectos que requieran mediciones precisas y confiables.
* Para más información, consultar la hoja de datos (Datasheet) del dispositivo (está en inglés): <http://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>
* Se puede escoger entre el modelo sumergible y los modelos para uso en placas de circuitos.

**Display LCD I2C:**

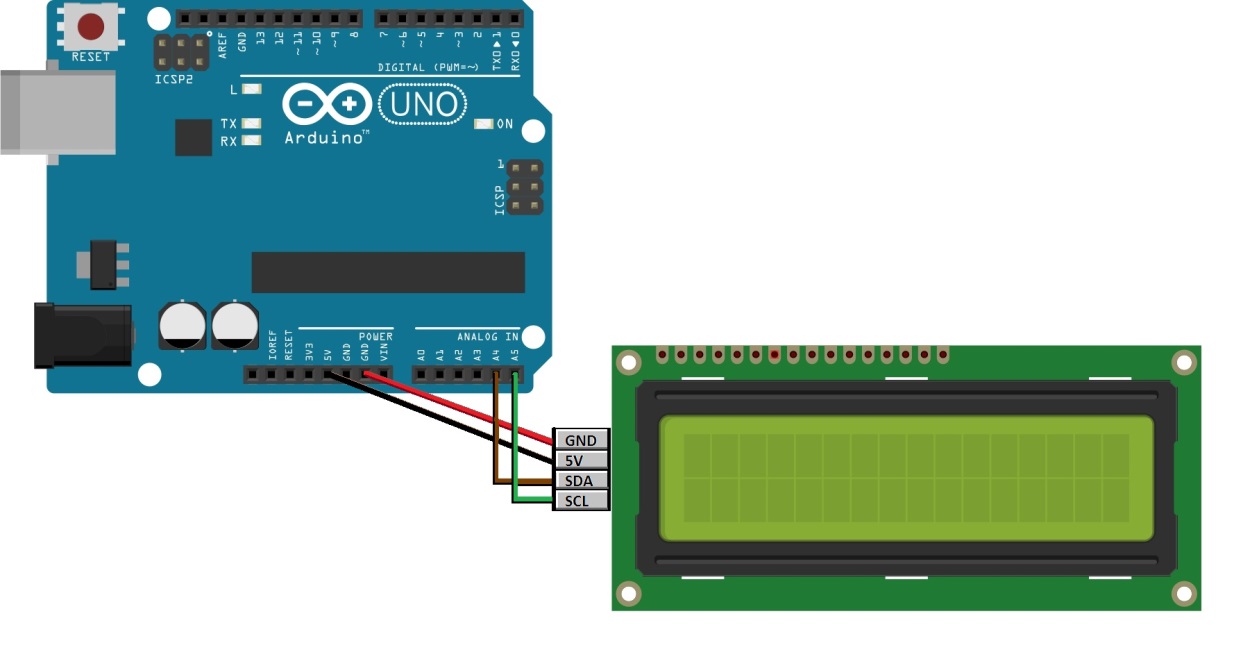
El módulo I2C LCD es un display de 2 líneas por 16 caracteres. Consta de 4 pines: alimentación (Vcc), masa (GND), transmisión de datos (SDA) y reloj asincrónico que indica cuando leer los datos (SCL).

Para el manejo del display, utiliza dos librerías: “LiquidCrystal\_I2C”  y “Wire”.

**Características:**

|  |  |
| --- | --- |
| **Especificaciones** | |
| Rango de direcciones I2C | 2 líneas de 16 caracteres |
| Tensión de funcionamiento | 5 Vdc |
| Luz de caracteres | Blanco |
| Contraste | Ajustable por potenciómetro |
| Tamaño | 80mm x 36mm x 20mm |
| Área visible | 66mm x 16mm |

**Forma de conexión:**



**Módulo Bluetooth HC-06:**

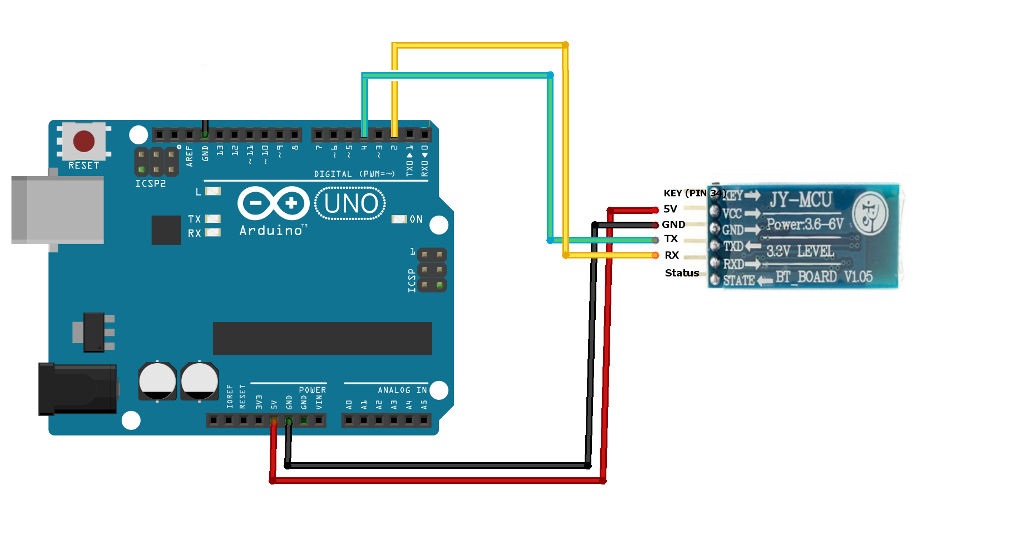
El módulo HC-06 es un dispositivo bluetooth que actúa únicamente como esclavo, y dispone de 4 pines: alimentación (VCC), masa (GND), transmisión de datos (TX) y recepción de datos (RX).

Este módulo posee un led que indica el estado en el que se encuentra. Si este led parpadea, significa que ningún dispositivo está vinculado a él. Cuando un dispositivo se conecta al módulo, la luz del led quedará fija y es la forma de saber si hay conexión o no.



Arduino posee un soporte incorporado para comunicación serial en los pines 0 y 1. Es por esto que utilizamos la librería “SoftwareSerial” que permite la comunicación serial (transmisión y recepción) en otros pines digitales del Arduino.

**Forma de conexión:**



Sensores, ecuaciones y funciones

**Ecuación de la recta para obtener el volumen de alcohol:**

Para generar la ecuación con la cual se obtiene el volumen de alcohol contenido en el recipiente se tomó el valor mínimo y máximo que el sensor de alcohol puede medir en un ambiente ideal a 21°C.

El sensor comienza a funcionar de forma estable con un tiempo de precalentamiento de 24 horas.

Estas pruebas se realizaron con el sensor recientemente prendido, debido a que la idea es hacer un dispositivo que funcione en el momento que se enciende. Para solucionar el inconveniente de la estabilidad, se calcula un promedio de 10 mediciones para obtener la ecuación que permite obtener el volumen de alcohol.

Valor Mínimo:

Se tomaron en el frasco del vaso cerrado con aire limpio 10 mediciones dando ellas en promedio 80 como resultado.

Valor Máximo:

Para el cálculo del valor máximo se vertió alcohol en el vaso del dispositivo hasta su tope máximo, con una concentración de alcohol al 96% (considerado grado de pureza más alto) y se obtuvo un promedio de 10 mediciones un valor de 420.

A partir de estos dos puntos se generó una ecuación de la recta para obtener el volumen de alcohol:

***Volumen de Alcohol = (Valor del sensor - 80) \* (0.27)***

**Ecuación para obtener el porcentaje de temperatura**

El sensor de temperatura DS18B20 capta temperaturas en el rango de -55°C a 125°C, tomando como porcentaje 0% de temperatura a -55 y como 100% a 125 se obtiene la siguiente ecuación para calcular el porcentaje de temperatura:

***PorcentajeTemp = (Valor Sensor + 55) \* (0.5)***

**Ecuación para obtener el nivel PWM del Led:**

A partir del porcentaje de temperatura se obtiene la salida PWM del dispositivo

Valor Mínimo:

Valor mínimo de la salida PWM = 0

Valor Máximo:

Valor máximo de la salida PWM = 255

***SalidaPWM = Redondear((PorcentajeTemp) \* (2.55))***

**Ecuación para obtener el volumen en mililitros del líquido en el vaso:**

Altura Vaso: 13.5 cm (Valor constante)

Distancia del sensor al liquido:Obtenida a partir del sensor ultrasónico.

Altura del líquido: (Altura del Vaso - Distancia del sensor al liquido)

Diámetro del vaso (Cilíndrico):7.5cm (Valor constante)

Área del vaso: Pi \* (Radio del Vaso) \* radio;

***Volumen del líquido:******Altura del líquido*** *\** ***Área del vaso***

Materiales utilizados

**Hardware:**

* Placa Arduino UNO
* 2 Protoboard
* Sensor de Gases y Alcohol MQ-3
* Sensor Ultrasónico HC-SR04
* Modulo Bluetooth HC-06
* Resistencia de 4.7 K Ohms
* Sensor de Temperatura DS18B20
* Display LCD I2C (16X2)
* Led

**Software:**

* Liberia LiquidCrystal\_I2C.h (Display LCD I2C)
* Librería DallasTemperature.h (Sensor DS18B20)
* Librería SoftwareSerial.h (Modulo HC-06)
* Android Studio
* Arduino
* Fritzing

**Materiales:**

* Vaso
* Tapa a rosca
* Botella de Alcohol Etílico al 96%