Sistema para el control de sensores, actuadores y periféricos en Java

Tecnología de Software para Electrónica

Trabajo de Investigación

Autor:
Jordy Bayas Escobar (alumno)
Joshua Reyes Jurado (alumno)
Roger Espinoza Naranjo (alumno)
Tutor/es:
Darwin Alulema Flores (tutor1)

10 de Mayo de 2019

Sistema para el control de sensores, actuadores y periféricos en Java

Subtítulo del proyecto

Autor/es
Jordy Bayas Escobar (alumno)
Joshua Reyes Jurado (alumno)
Roger Espinoza Naranjo (alumno)

Tutor/es
Darwin Alulema Flores (tutor1)
Departamento de Eléctrica y Electrónica

Tecnología de Software para Electrónica

Quito, 10 de Mayo de 2019

Índice general

Índice de figuras

Índice de Códigos

1 Planteamiento del Problema

Desarrollar un sistema permite el control de distintos sensores, actuadores y periféricos en Java

Con el avance constante de la tecnología, es posible que una o mas computadores estén conectados de manera serial a través del RS232, esto a su vez puede realizar interaciones de las distintos sensores y actuadores que se pueda dar, entre maquina-usuario, con ello realizan distintos tipos de funciones con la intención de mostrar datos precisos y veraces dependiendo si el usuario los solicita. En la actualidad multitud de plataformas se han venido desarrollando tanto de manera de hardware y software con un formato libre, además de un coste reducido, gran documentación y una comunidad bastante activa, la plataforma Arduino es una de ellas esta a su vez dispone de suficientes librerías y una comunidad activa que facilitan la interconexión del Arduino con diferentes módulos de sensado, facilitando de esta forma la realización de proyectos deseados, entre ellas las conexiones con otros tipos de lenguaje como es el caso de Java.

2 Objetivos

2.1 Objetivo General

Desarrollar un sistema permite el control de distintos sensores, actuadores y periféricos en Java

2.1.1 Objetivos específicos

- Especificar las variables para el desarrollo del sistema para los distintos sensores y actuadores.
- Identificar los distintos sensores de CO2, Humedad, Temperatura y de proximidad a controlar mediante Java.
- Deducir las distintas aplicaciones mediante la ulizacación del cable serial, además de las interfaces GUI de Java.
- Diseñar un prototipo de software para que cada uno de los sensores está asociado a un relé para controlar un ventilador, una niquelina, un foco y un LCD para el despliegue de información del sistema.

3 Estado del Arte

Al revisar distintos ambitos en donde se esta llegando a aplicar los distintos módulos, se puede encontrar grandes temas, por el cual se llegó al artículo realizado por J.Chandramohan, R.Nagarajan, K. Satheeshkumar, N.Ajithkumar, A.Gopinath, S.Ranjithkumar los cuales en su paper " Intelligent Smart Home Automation and Security System Using Arduino and Wi-fi" donde se resume una aplicación de un sistema de control y monitoreo del hogar flexible y de bajo costo con la ayuda de un equipo integrado.[**?**]
Servidor de micro-web con conectividad de protocolo de Internet (IP) para acceder y controlar equipos y dispositivos de forma remota mediante aplicación para smartphone basada en Android, y utilizando Arduino, como evidencia se puede aseverar que en la actualidad hay diversas apelaciones que se vienen realizando estos procedimientos, además de incluir una base de datos donde se guarda todo los cambios que has realizado en los equipos, pero además hay otro artículos donde se realizan mediante conexiones LAN como es el caso de "Home automation using arduino WiFi module ESP8266" de Kotiyal Bandanawaz,Baig Iliyas,Muzamil Muzamil Chiktay, Dalv Salahuddin donde ellos proponen un enfoque diferente donde se presenta un diseño y una implementación prototipo de un nuevo sistema de automatización del hogar que utiliza la tecnología Wi-Fi como una infraestructura de red que conecta sus partes. Donde consta de un servidor (servidor web), que presenta el núcleo del sistema que administra, controla y supervisa la casa de los usuarios el cual se puede administrar nivel local (LAN) o de forma remota (internet). La segunda parte es el módulo de interfaz de hardware, que proporciona una interfaz adecuada para los sensores y el actuador del sistema domótico. Esto llega a ser interesante, porque se puede intuir 2 cosas hoy en día ya no se es limitado por una red de área local, es decir ahora se puede controlar ingresando directamente vía internet a un servidor web especifico que controle distintas funciones, en varios ámbitos.[1]
Otro artículo "IoT based smart water tank with Android application" hecho por Priyen P. Shah, Anjali A. Patil,Subodh S. Ingleshwar, el cual menciona otro uso mediante aplicaciones esta vez tratándose monitoreos contantes del nivel del agua, pero usando sensores, la señal de estos sensores es captada por el arduino y transmitida por el módulo wi-fi para poder receptar mediante la aplicación basada en android. Este paper abarca una propuesta diferente utilizando el internet de las cosas,que permite la interacción con aplicaciones externas, como la interfaz ofrecida por los componentes individuales de la infraestructura, que permite la interacción y la coordinación entre los componentes. Básicamente su base viene dada para contribuir a una proyección

estándar de las mediciones y control del agua dentro de tanques además eliminar cualquier inconveniente y proporcionar una solución eficiente y económica.[2]
Al observar el artículo "Intelligent multi-sensor control system based on innovative technology integration via ZigBee and Wi-Fi networks" de los autores Kuang-Yow Lian, Sung-Jung Hsiao y Wen-Tsai Sung, donde se descrbe que una red de transmisión de datos es uno de los problemas más difíciles de resolver en los sistemas de redes de sensores inalámbricos . ZigBee y Wi-Fi pertenecen a diferentes protocolos de red . Si un sistema de red debe usar ZigBee y Wi-Fi al mismo tiempo para transmitir datos, se presenta un desafío considerable. Este artículo presenta un nuevo método de hardware que integra ZigBee y Wi-Fi. El método propuesto se basa en el módulo portátil de Arduino ZigBee y el concepto de Ethernet. Este estudio construye un sistema inteligente de control de electrodomésticos utilizando la red ZigBee. Este sistema de control inteligente utiliza una arquitectura de red integrada ZigBee y Wi-Fien la casa. Nuestro estudio envía los mensajes del sensor ZigBee a una base de datos en la nube a través de la red de protocolo TCP / IP que contiene la red física y las líneas de dispositivos de red inalámbrica . El control deacceso a la gestiónse logra mediante teléfonos inteligentes. El método propuesto es muy simple y fácil de implementar utilizando circuitos Arduino. La efectividad del método propuesto se verifica mediante la simulación ylos resultados experimentales. Loscomponentes de hardwareincluyen el controlador Arduino, elmódulo de comunicacióninalámbrica XBee Serie 2ylossensores deldispositivo final. Loslenguajes de programación deAndroid yJava. Se utilizan para escribir el teléfono inteligente y los programas de reconocimiento del servidor.[3]

Viendo otros usos que se pueden dar las utilizaciones del módulo se puede encontrar el articulo "Taking Arduino to the Internet of Things: The ASIP programming model" realizado por Gianluca Barbona, Michael Margolis, Filippo Palumbo, Franco Raimondi, Nick Weldin nos meciona que los microcontroladores como Arduino son ampliamente utilizados por todo tipo de fabricantes en todo el mundo. La popularidad se debe a la simplicidad de uso de Arduino y la gran cantidad de sensores y bibliotecas disponibles para ampliar las capacidades básicas de estos controladores. La última década ha sido testigo de una oleada de soluciones de ingeniería de software para "Internet of Things", pero en varios casos estas soluciones requierenrecursos computacionalesque son más avanzados que los microcontroladores simples y con recursos limitados.En el presenten documento presentan el modelo de Programación de Interfaz de Servicio de Arduino (ASIP), un nuevo modelo que aborda los problemas anteriores proporcionando una abstracción de "Servicio" para agregar fácilmente nuevas capacidades a los microcontroladores, y brindando soporte para tableros en red que utilizan una variedad de estrategias, que incluyen conexiones de socket , dispositivos de puente, publicación basada en MQTT, mensajería de suscripción, servicios de descubrimiento, etc. Ofrecemos una implementación de código abierto del código que se ejecuta en las placas Arduino y las bibliotecas de clientes en Java, Python, Racket y Erlang. Mostramos cómo ASIP permite el rápido desarrollo de aplicaciones no triviales (coordinación de entrada / sa-

lida en tableros distribuidos e implementación de un algoritmo de seguimiento de línea para un robot remoto) y evaluamos el rendimiento de ASIP de varias maneras, tanto cuantitativas como cualitativas. [4]

Como se puede observar el tema de la investigación tiene diversas investigaciones, artículos que poseen cierta relación con lo que se esta investigando, pero este trabajo se diferencia de los demás en que se quiere integrar Arduino de manera serial mediante la conexión entre 2 PC. Se profundizará en varios temas que no se han investigado juntos y se orientara más a las distintas ventajas que ofrece la conexión serial, partiendo de que en el arduino se puede realizar tanto conexiones en paralelo como en serie y convertir a su vez la salida como tener actuadores, es decir, los distintos actuadores, como es el ventilador, la niquelina, un foco, y la apreciación de los datos mediante la pantalla LCD, la misma que s epuede encontrar comunmente las libreías en Arduino para ocuparlas. Por lo que nuestro trabajo unirá los beneficios de plataformas gratuitas que ofrezcan servicios especiales con las características del Arduino, la interfaz de Java y además de una aplicación para demostrar el potencial que puede tener el uso de estos.

4 Marco Teórico

4.1 Modos Comunicacion

Se pueden establecer canales para la comunicación de acuerdo a tres técnicas:

- Simplex
- Semi duplex
- Totalmente duplex

4.1.1 Simplex

En ella la comunicación serie usa una dirección y una línea de comunicación. Siempre existirá un transmisor y un receptor, no ambos.
La ventaja de este sistema consiste en que es necesario sólo un enlace a dos hilos.
La desventaja radica en que el extremo receptor no tiene ninguna forma de avisar al extremo transmisor sobre su estado y sobre la calidad de la información que se recibe. Esta es la razón por la cual, generalmente, no se utiliza.

4.1.2 Semi duplex

La comunicación serie se establece a través de una sola línea, pero en ambos sentidos. En un momento el transmisor enviará información y en otro recibirá, por lo que no se puede transferir información en ambos sentidos de forma simultánea .

Este modo permite la transmisión desde el extremo receptor de la información, sobre el estado de dicho receptor y sobre la calidad de la información recibida por lo que permite así la realización de procedimientos de detección y corrección de errores.

4.1.3 Full duplex

Se utilizan dos líneas (una transmisora y otra receptora) y se transfiere información en ambos sentidos. La ventaja es que se puede transmitir y recibir información de manera simultánea.

La mayoría de los dispositivos especializados para la comunicación pueden transferir información tanto en full duplex como en half duplex (el modo simplex es un caso especial dentro de half duplex).

4.2 Comunicación Serie

Es el proceso de envío de datos de un bit a la vez, de forma secuencial, sobre un canal de comunicación o un bus. En cambio, en la "comunicación en paralelo" todos los bits de cada símbolo se envían al mismo tiempo, y por ello debe haber al menos tantas líneas de comunicación como bits tenga la información a transmitida.
La ventaja de la comunicación serie es que necesita un número más pequeño de líneas de transmisión que una comunicación paralela que transmita la misma información. Esta última necesita tantas líneas de transmisión como la cantidad de bits que componen la información, mientras que la primera se puede llevar a cabo con una sola línea de transmisión. Por otra parte, surgen una serie de problemas en la transmisión de un gran número de bits en paralelo, como los problemas de interferencia o de-sincronización. A la misma frecuencia de transmisión, la comunicación paralela tiene un mayor rendimiento. La comunicación serie tiene que compensar esta debilidad con una frecuencia más alta.

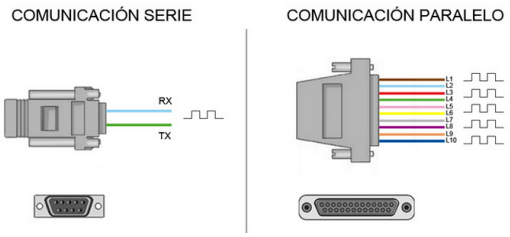


Figura 4.1: Series vs Paralelo

4.3 RS-232

RS-232 (Estándar ANSI/EIA-232) es el conector serial que se encuentra en las PCs compatibles con IBM.
Se lo utiliza con diversos propósitos, como el conectar periféricos, impresoras, o módems, así como para instrumentación industrial.
El RS-232 está limitado a conexiones punto a punto entre puertos seriales y dispositivos PC.
La comunicación del RS-232 es asíncrona, esto quiere decir que los datos no son enviados de acuerdo a un reloj, sino que, cada byte de dato, es enviado utilizando un bit de

inicio.

4.4 Arduino

Arduino es una plataforma de creación de electrónica de código abierto, la cual está basada en hardware y software libre, flexible y fácil de utilizar para los creadores y desarrolladores. Esta plataforma permite crear diferentes tipos de microordenadores de una sola placa a los que la comunidad de creadores puede darles diferentes tipos de uso.

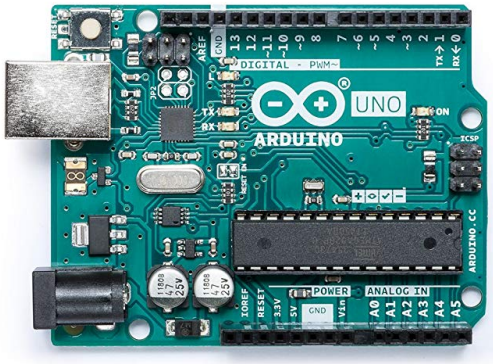


Figura 4.2: Arduino Uno

4.4.1 Hardware y Software Libre

El hardware libre son los dispositivos cuyas especificaciones y diagramas son de acceso público, de manera que cualquiera puede replicarlos. Esto quiere decir que Arduino ofrece las bases para que cualquier otra persona o empresa pueda crear sus propias placas, pudiendo ser diferentes entre ellas, pero igualmente funcionales al partir de la misma base. El software libre son los programas informáticos cuyo código es accesible por cualquiera para que quien quiera pueda utilizarlo y modificarlo. Arduino ofrece la plataforma Arduino IDE (Entorno de Desarrollo Integrado), que es un entorno de programación con el que cualquiera puede crear aplicaciones para las placas Arduino, de manera que se les puede dar todo tipo de utilidades.

4.4.2 Funcionamiento y estructura

El Arduino es una placa basada en un microcontrolador ATMEL. Los microcontroladores son circuitos integrados en los que se pueden grabar instrucciones, las cuales las escribes con el lenguaje de programación que puedes utilizar en el entorno Arduino

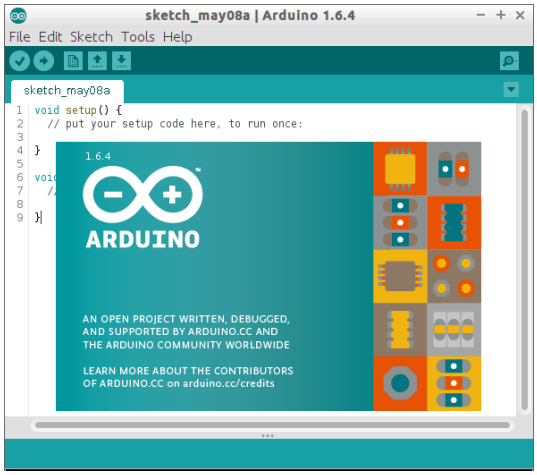


Figura 4.3: Arduino Uno

IDE. Estas instrucciones permiten crear programas que interactúan con los circuitos de la placa.

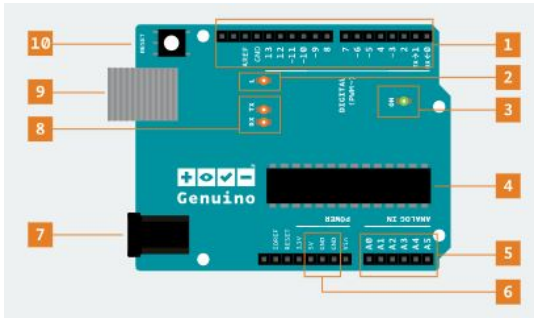


Figura 4.4: Arduino Uno

1. Pines digitales Utilice estos pines con digitalRead (), digitalWrite () y analogWrite (). analogWrite () funciona solo en los pines con el símbolo PWM.

2. Pin 13 LED El único actuador incorporado a su tablero. Además de ser un objetivo práctico para su primer boceto de parpadeo, este LED es muy útil para la depuración.

3. LED de alimentación Indica que su Genuino está recibiendo alimentación. Útil para la depuración.

4. ATmega microcontrolador El corazón de tu tablero.

5. Analógico en Utilice estos pines con analogRead ().

6. Pines GND y 5V Use estos pines para proporcionar alimentación de + 5V y tierra a sus circuitos.

7. Conector de alimentación Así es como alimenta su Genuino cuando no está enchufado a un puerto USB para la alimentación. Puede aceptar voltajes entre 7-12V.

8. LED de TX y RX Estos LED indican la comunicación entre su Genuino y su computadora. Espere que parpadeen rápidamente durante la carga del boceto, así como durante la comunicación en serie. Útil para la depuración.

9. Puerto USB Se utiliza para alimentar su Genuino Uno, cargar sus bocetos en su Genuino y para comunicarse con su bosquejo de Genuino (a través de Serial. Println () etc.).

10. Botón de reinicio Reinicia el microcontrolador ATmega .

4.5 MQ7

Es un sensor fácil de usar para detección de Monóxido de Carbono (CO), ideal para detectar concentraciones de CO en el aire. El MQ-7 puede detectar concentraciones en el rango de 20 a 2000ppm.
El módulo posee una salida analógica que proviene del divisor de voltaje que forma el sensor y una resistencia de carga. También tiene una salida digital que se calibra con un potenciómetro, esta salida tiene un led indicador.

4.5.1 Especificaciones Técnicas

- Voltaje de Operación: 5V DC
- Voltaje de Calentamiento: 5V (alto) y 1.4V (bajo)
- Resistencia de carga: regulable
- Resistencia de calentamiento: 33 Ohm
- Tiempo de Calentamiento: 60s (alto) 90s (bajo)
- Consumo de Resistencia: aprox. 350mW
- Concentración de Oxigeno: 21

Figura 4.5: Ejemplo Serial.begin()

4.6 Sensor Ultrasónico HC-SR04

El HC-SR04 es un sensor de distancias por ultrasonidos capaz de detectar objetos y calcular la distancia a la que se encuentra en un rango de 2 a 450 cm. El sensor funciona por ultrasonidos y contiene toda la electrónica encargada de hacer la medición. Su uso es tan sencillo como enviar el pulso de arranque y medir la anchura del pulso de retorno. De muy pequeño tamaño, el HC-SR04 se destaca por su bajo consumo, gran precisión y bajo precio por lo que esta reemplazando a los sensores polaroid en los robots mas recientes.

4.6.1 Características

- Dimensiones del circuito: 43 x 20 x 17 mm
- Tensión de alimentación: 5 Vcc
- Frecuencia de trabajo: 40 KHz
- Rango máximo: 4.5 m
- Rango mínimo: 1.7 cm
- Duración mínima del pulso de disparo (nivel TTL): 10 S.
- Duración del pulso eco de salida (nivel TTL): 100-25000 S.
- Tiempo mínimo de espera entre una medida y el inicio de otra 20 mS.

4.6.2 Pines de COnexión

- Trig (Disparo del ultrasonido)
- VCC
- Trig (Disparo del ultrasonido)
- GND

Figura 4.6: Ejemplo Serial.begin()

4.7 Comandos para la Comunicación Serial

4.7.1 Serial.begin(rate)

Abre el puerto serie y fija la velocidad en baudios para la transmisión de datos en serie. El valor típico de velocidad para comunicarse con el ordenador es 9600, aunque otras velocidades pueden ser soportadas.[**?**] Nota: Cuando se utiliza la comunicación

```
void setup()

{
Serial.begin(9600);    // abre el Puerto serie
}    // configurando la velocidad en 9600 bps
```

Figura 4.7: Ejemplo Serial.begin()

serie los pines digitales 0 (RX) y 1 (TX) no pueden utilizarse para otros propósitos.

4.7.2 Serial.println(data)

Imprime los datos en el puerto serie, seguido por un retorno de carro y salto de línea. Este comando toma la misma forma que Serial.print (), pero es más fácil para la lectura de los datos en el Monitor Serie del software.[**?**]

```
Serial.println(analogValue);    // envía el valor 'analogValue' al puerto
```

Figura 4.8: Ejemplo Serial.println()

4.7.3 Serial.println(data, data type)

Vuelca o envía un número o una cadena de carateres al puerto serie, seguido de un caracter de retorno de carro ”CR” (ASCII 13, or ’\r’)y un caracter de salto de línea ”LF”(ASCII 10, or ’\n’). Toma la misma forma que el comando Serial.print()

Serial.println(b) vuelca o envía el valor de b como un número decimal en caracteres ASCII seguido de ”CR” y ”LF”.
Serial.println(b, DEC) vuelca o envía el valor de b como un número decimal en caracteres ASCII seguido de ”CR” y ”LF”.
Serial.println(b, HEX) vuelca o envía el valor de b como un número hexdecimal en caracteres ASCII seguido de ”CR” y ”LF”.
Serial.println(b, OCT) vuelca o envía el valor de b como un número octal en caracteres ASCII seguido de ”CR” y ”LF”.
Serial.println(b, BIN) vuelca o envía el valor de b como un número binario en caracteres ASCII seguido de ”CR” y ”LF”.
Serial.print(b, BYTE) vuelca o envía el valor de b como un byteseguido de ”CR” y ”LF”.
Serial.println(str) vuelca o envía la cadena de caracteres como una cadena ASCII seguido de ”CR” y ”LF”.
Serial.println() sólo vuelca o envía ”CR” y ”LF”.

4.7.4 Serial.avaible()

Devuelve un entero con el número de bytes (carácteres) disponibles para leer desde el buffer serie, ó 0 si no hay ninguno. Si hay algún dato disponible, SerialAvailable() será mayor que 0. El buffer serie puede almacenar como máximo 128 bytes.[**?**]

```
int incomingByte = 0; // almacena el dato serie
void setup() {
Serial.begin(9600); // abre el puerto serie, y le asigna la velocidad de 9600 bps
}
void loop() {
// envía datos sólo si los recibe:
if (Serial.available() > 0) {
// lee el byte de entrada:
incomingByte = Serial.read();
//lo vuelca a pantalla
Serial.print("He recibido: "); Serial.println(incomingByte, DEC);
}
}
```

Figura 4.9: Ejemplo Serial.avaible()

4.7.5 Serial.read()

Lee o captura un byte (carácter) desde el puerto serie. Devuelve :El siguiente byte (carácter) desde el puerto serie, ó -1 si no hay ninguno.[**?**]

```
int incomingByte = 0; // almacenar el dato serie
void setup() {
Serial.begin(9600); // abre el puerto serie,y le asigna la velocidad de 9600 bps
}
void loop() {
// envía datos sólo si los recibe:
if (Serial.available() > 0) {
// lee el byte de entrada:
incomingByte = Serial.read();
//lo vuelca a pantalla
Serial.print("He recibido: "); Serial.println(incomingByte, DEC);
}
}
```

Figura 4.10: Ejemplo Serial.read()

5 Diagramas

Figura 5.1: Conexión Serial Entre dos computadoras, y control de sensores y actuadores

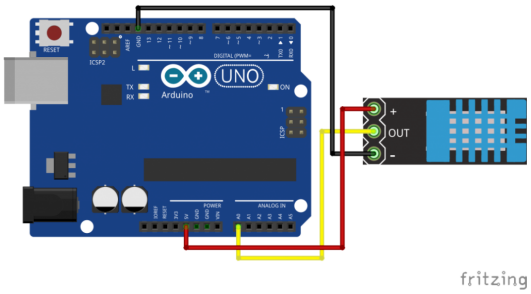


Figura 5.2: Conexión Sensor de Temperatura

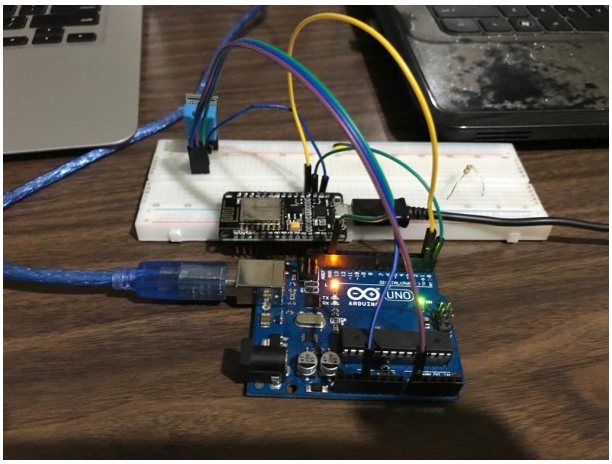


Figura 5.3: Circuito Implementado

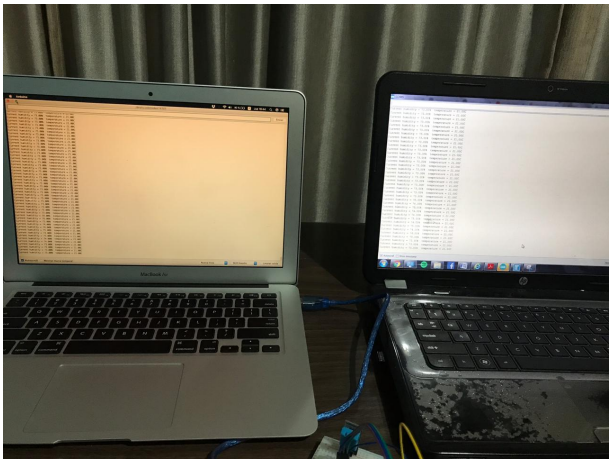


Figura 5.4: Monitores Serie

6 Lista de Componentes

6.1 Arduino Uno

Arduino UNO es una placa de microcontrolador open source basado en el microcontrolador ATmega328P de Microchip y desarrollado por Arduino.cc. el tablero está equipado con conjuntos de analógicos y digitales entrada/salida (E/S) los pernos que pueden conectarse a diversas juntas de expansión (escudos) y otros circuitos.

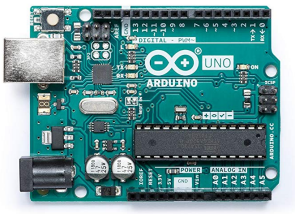


Figura 6.1: Arduino Uno

6.2 MQ7

Es un sensor fácil de usar para detección de Monóxido de Carbono (CO), ideal para detectar concentraciones de CO en el aire. El MQ-7 puede detectar concentraciones en el rango de 20 a 2000ppm.El módulo posee una salida analógica que proviene del divisor de voltaje que forma el sensor y una resistencia de carga. También tiene una salida digital que se calibra con un potenciómetro, esta salida tiene un led indicador.

Figura 6.2: MQ7

6.3 Sensor Ultrasónico HC-SR04

El HC-SR04 es un sensor de distancias por ultrasonidos capaz de detectar objetos y calcular la distancia a la que se encuentra en un rango de 2 a 450 cm.

Figura 6.3: HC-sr04

6.4 Arduino IDE

Es una plataforma electrónica de código abierto basada en hardware y software fáciles de usar. Las placas Arduino pueden leer entradas (luz en un sensor, un dedo en un botón o un mensaje de Twitter) y convertirla en una salida: activar un motor, encender un LED y publicar algo en línea. Puede decirle a su tarjeta qué debe hacer enviando un conjunto de instrucciones al microcontrolador de la tarjeta. Para hacerlo, utiliza el lenguaje de programación Arduino (basado en Wiring) y el software Arduino (IDE) , basado en el procesamiento .

6.5 Sensor DHT11

El sensor DHT11 que nos permite medir la temperatura y humedad con Arduino. Una de las ventajas que nos ofrece el DHT11, además de medir la temperatura y la humedad, es que es digital. A diferencia de sensores como el LM35, este sensor utiliza un pin digital para enviarnos la información y por lo tanto, estaremos más protegidos frente al ruido.

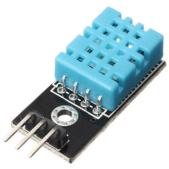


Figura 6.4: Sensor DHT11

6.6 Relé

Es un dispositivo electromagnético. Funciona como un interruptor controlado por un circuito eléctrico en el que, por medio de una bobina y un electroimán, se acciona un juego de uno o varios contactos que permiten abrir o cerrar otros circuitos eléctricos independientes.

Figura 6.5: Relé ELectrónico

6.7 Niquelina

Es un mineral compuesto de arseniuro de níquel,4 NiAs, que contiene 43,9 porciento de níquel y el 56,1 porciento de arsénico.

Suelen contener pequeñas cantidades de azufre, hierro y cobalto, y a veces el arsénico es reemplazado en gran medida por el antimonio. Forma una serie isomorfa con la breithauptita (antimoniuro de níquel).

Figura 6.6: Niquelina

6.8 Ventilador

Es una máquina de fluido, más exactamente una turbo máquina que transmite energía para generar la presión necesaria con la que se mantiene un flujo continuo de aire. Se utiliza para usos muy diversos como: ventilación de ambientes, refrescamiento de máquinas u objetos o para mover gases, principalmente el aire, por una red de conductos.

Figura 6.7: Ventilador

6.9 LCD

Es una pantalla delgada y plana formada por un número de píxeles en color o monocromos colocados delante de una fuente de luz o reflectora. A menudo se utiliza en dispositivos electrónicos de pilas, ya que utiliza cantidades muy pequeñas de energía eléctrica.

Figura 6.8: LCD

7 Mapa de Variables

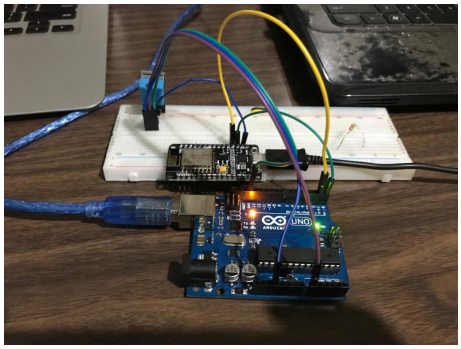


Figura 7.1: Arduino Uno y ESP8266 Interconectados.

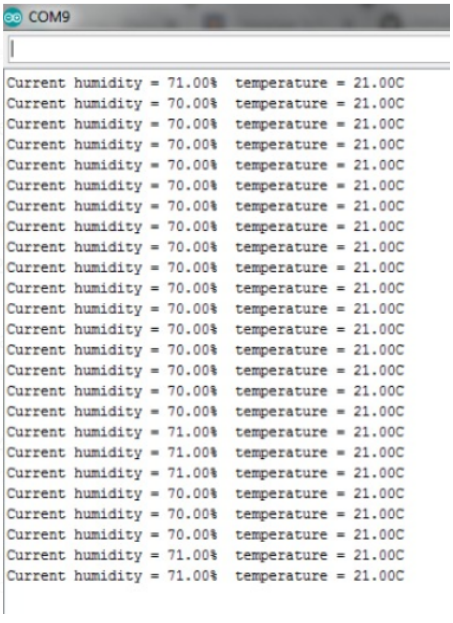


Figura 7.2: Datos en el monitor Serie.

8 Explicación del Código Fuente

8.1 Código Arduino UNO

El código completo cargado en el Arduino UNO se vería de las siguiente manera:

Código 8.1: Arduino UNO

```
#include <dht.h>

#define RLOAD 22.0
#define RZERO 879.13
#include "MQ135.h"
dht DHT;
#define dht_temp A3
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27,20,4);

long tiempo;
int disparo = 2;
int echo = 3;
float distancia;
MQ135 gasSensor = MQ135(A0);
int val;
int sensorPin = A0;
int sensorValue = 0;


void setup() {
        lcd.init();
    lcd.backlight();
   lcd.setCursor(10, 0);
  Serial.begin(9600);
    pinMode(disparo, OUTPUT);
  pinMode(echo, INPUT);
  digitalWrite(disparo, LOW);
  pinMode(sensorPin, INPUT);
}

void loop() {
  val = analogRead(A0);
  float ppm = gasSensor.getPPM();
  Serial.print ("ppm: ");
  Serial.println (ppm);
  delay(1000);
  digitalWrite(disparo, HIGH); //Se envia el pulso de activacion del sensor
  delayMicroseconds(10);
  digitalWrite(disparo, LOW);
  tiempo = pulseIn(echo, HIGH); // Obtengo el tiempo del sensor
  distancia = 0.0343*tiempo/2; //Calculo la distancia
  Serial.print(distancia);
  Serial.println("cm");//Muestro la distancia en el monitor serie
  delay(1000);
```

```
47
48  //TEMPERATURA//
49   DHT.read11(dht_temp);
50  Serial.print("Temperatura = ");
51  Serial.print(DHT.temperature);
52  Serial.println(" C");
53  //HUMEDAD RELATIVA//
54  Serial.print("Humedad = ");
55  Serial.print(DHT.humidity);
56  Serial.println(" %");
57  delay(1000);
58 sensorValue = map(analogRead(A1), 0, 1023, 0, 100);
59  Serial.print("Umbral de Ruido :");
60  Serial.print(sensorValue);
61   Serial.println(" %");
62     delay(1000);
63  lcd.setCursor(0,0);
64   lcd.print("d:");
65       lcd.print(distancia);
66         delay(100);
67         lcd.setCursor(6,0);
68
69   lcd.print("G:");
70       lcd.print(ppm);
71  delay(100);
72       lcd.setCursor(12,0);
73
74   lcd.print("R:");
75       lcd.print(sensorValue);
76        delay(100);
77          lcd.setCursor(0,1);
78
79   lcd.print("T: ");
80       lcd.print(DHT.temperature);
81       lcd.setCursor(8,1);
82       lcd.print("H: ");
83       lcd.print(DHT.humidity);
84       lcd.print(" %");
85         delay(100);
86 }
```

En el codigo anterior se explica el funcionamiento de los sensores y de como va a enviar los datos serialmente el Arduino y que datos nomas va a enviar.
Como se puede ver vamos a tener datos de humedad, temperatura, distancia, ruido y CO2.

9 Descripción de Prerrequisitos y Configuración

Para la realizacion del sistema se necesita una lista de programas y ya previamente instalados como:

- JAVA
- Netbeans
- Arduino IDE

9.0.1 JAVA

Necesitamos tener instalado java de 32 bits ya que para poder utilizar el API de comunicaciones es necesario tener instalado esa version. Una ves obtenido el java de 32 bits necesitamos descargar aparte el API de comunicaciones de java llamado javaxcomm

9.0.2 Arduino IDE

En la actualidad el desarrollo de aplicaciones donde se pueda interactuar en tiempo real y sobre todo, saber el funcionamiento, y las distintas características que estas pueda tener, es por ello que se necesita el software libre, el Arduino IDE.

Al ser un software libre se lo puede descargar directamente de la página del mismo, de manera totalmente gratuito, aunque si se desea dejar un aporte para ayudar a las personas detrás del trabajo, se lo puede realizar.

Además de ello se debe agregar la correspondiente dirección conde se encuentra todos los archivos destinados del módulo nodeMCU.

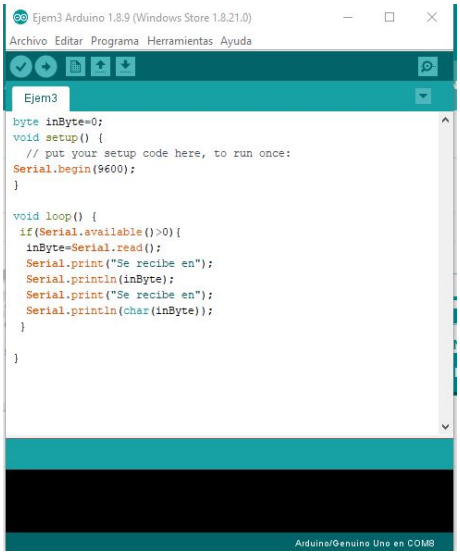


Figura 9.1: Arduino IDE instalada

Figura 9.2: Descarga de Arduino IDE

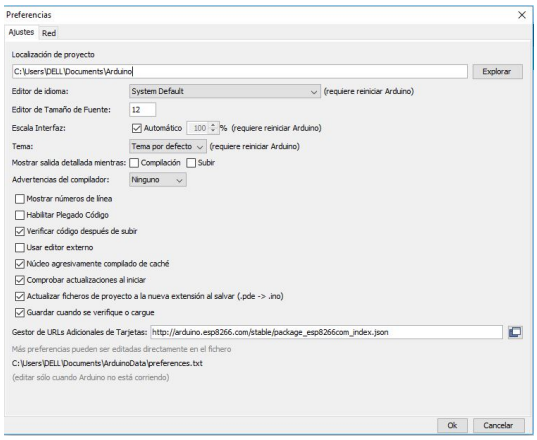


Figura 9.3: Dirección donde se encuentra la librería

Realizando este proceso correspondiente, se actualizaran las librerías instalas en su Arduino IDE, y aquellas que aun no se han instalado, por ello se debe buscar el nombre del módulo a ocupar, en este caso se utilizara el ESP8266, y por lo tanto procederá a

proporcionar las librerías correspondientes.

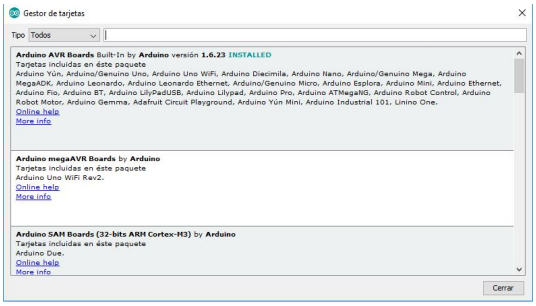


Figura 9.4: Instalación de la Librería en arduino IDE

10 Aportaciones

Utilizacion del sensor DHT11 que nos permite medir la temperatura y humedad con Arduino.
Una de las ventajas que nos ofrece el DHT11, además de medir la temperatura y la humedad, es que es digital. A diferencia de sensores como el LM35, este sensor utiliza un pin digital para enviarnos la información y por lo tanto, estaremos más protegidos frente al ruido.
Inclusión de un minitutorial de como incluir librerías que no están previamente incluidas.
Inclusion de un sensor de ruido para tambien detectar la cantidad de ruido d eun lugar.
Inclusion del sensor HCSR04.
Inclusion sensor MQ7.

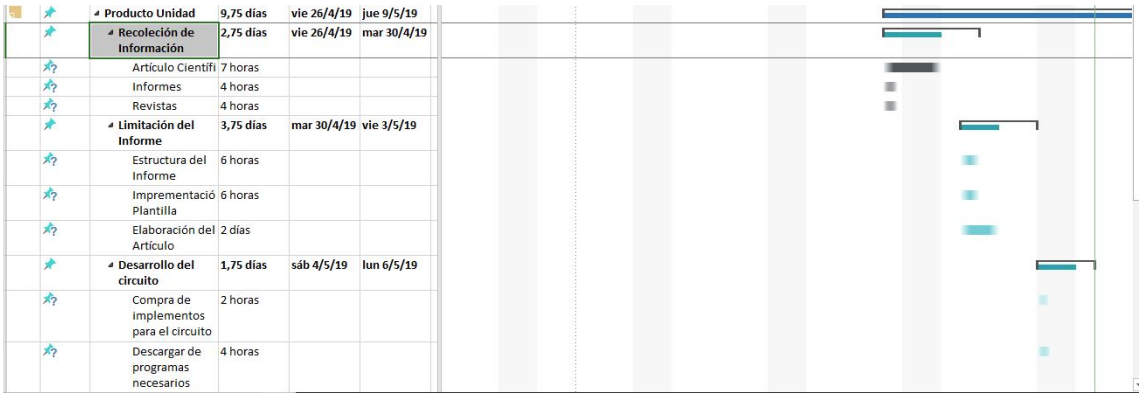
11 Conclusiones

- Arduino es muy versátil para el desarrollo de proyectos ya que es fácil de trabajar y acoplar sus módulos con lo cual nos da una amplia capacidad para realizar sistemas, aplicaciones en casi cualquier tema.
- Con el javaxcomm y sin necesitar de otra libreria es posible realizar la conexion semiduplex del sistema propuesto.
- Con la facilidad que se realiza las conexiones seriales se podría para un trabajo futuro realizar un sistema mas complejos donde se tendria mas dispositivos conectados en delta.
- Las distintas conexiones seriales de arduino, pueden conectarse indistintamente en cualquier tipo de placa basada en el mismo, como nodeMCU o un sensor de temperatura, se puede concluir que la ventaja de arduino es poder incluir este tipo de uniones, y también encontrar dichas librerías para una óptima utilización de la programación realizada.

12 Recomendaciones

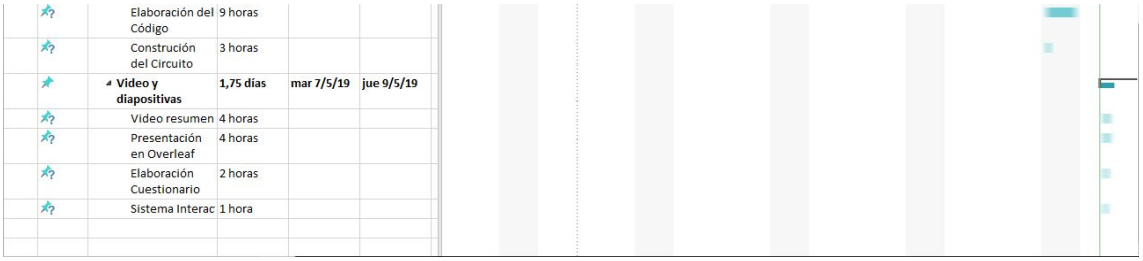
- Tener conocimientos básicos de lo que es el código arduino, java y lo que representa las conexiones seriales.
- Comprender la naturaleza de los sensores y actuadores aplicados al sistema.
- Estudiar el diagrama de conexiones del arduino, y de los sensores ocupados.
- Comprender el funcionamiento básico de las interfaces GUI de Java.
- Se recomienda verificar detalladamente la estructura del prototipo a diseñar, teniendo ideas claras, y el enfoque a donde va dirigido.

13 Cronograma



Producto Unidad	9,75 días	vie 26/4/19	jue 9/5/19	
Recolección de Información	2,75 días	vie 26/4/19	mar 30/4/19	
Artículo Científi	7 horas			
Informes	4 horas			
Revistas	4 horas			
Limitación del Informe	3,75 días	mar 30/4/19	vie 3/5/19	
Estructura del Informe	6 horas			
Impremantació Plantilla	6 horas			
Elaboración del Artículo	2 días			
Desarrollo del circuito	1,75 días	sáb 4/5/19	lun 6/5/19	
Compra de implementos para el circuito	2 horas			
Descargar de programas necesarios	4 horas			

Figura 13.1: Cronograma de actividades parte 1



Elaboración del Código	9 horas			
Construción del Circuito	3 horas			
Video y diapositivas	1,75 días	mar 7/5/19	jue 9/5/19	
Video resumen	4 horas			
Presentación en Overleaf	4 horas			
Elaboración Cuestionario	2 horas			
Sistema Interac	1 hora			

Figura 13.2: Cronograma de actividades parte 2

14 Repositorio

14.1 Git Hub

https://github.com/JOSHUARYES/producto-unidad1

15 Uso de Herramientas Cloud

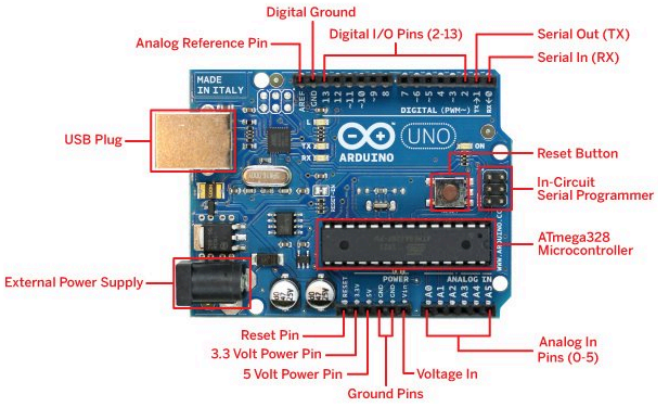
15.1 Links Overleaf

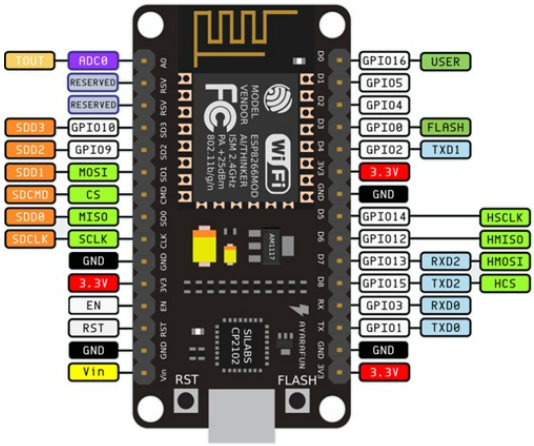
https://es.overleaf.com/6578684664rrhxyvhpfhfd
https://es.overleaf.com/5764412614jswtpgxbyhfz

16 Anexos

16.1 Manual de usuario

Para el correcto funcionamiento del proyecto se debe tomar en cuenta los pines de las tarjetas Arduino





Tambien debemos tener instalado el IDE de arduino para poder observar por el monitor serie los datos.
Y debemos tener las respectivas conexiones entre las tarjetas y las computadoras. Una ves conectados no se necesita hacer nada mas automáticamente comienza a funcionar.

Bibliografía

[1] F Baig, Muzamil dan Dalvi. Home Automation Using Arduino Wifi Module Esp8266 a Project Report Ilyas Baig Chiktay Muzamil Salahuddin Dalvi. 2016. ISSN 2015-2016. URL https://core.ac.uk/download/pdf/55305294.pdf.

[2] Subodh S. Ingleshwar Priyen P. Shah, Anjali A. Patil. IoT based smart water tank with Android application. In *IoT based smart water tank with Android application*, Palladam, India, 2017. IEEE. URL https://ieeexplore.ieee.org/abstract/document/8058250.

[3] Wen-Tsai Sung Kuang-Yow Lian, Sung-Jung Hsiao. Intelligent multi-sensor control system based on innovative technology integration via ZigBee and Wi-Fi networks. *Network and Computer Applications*, Volume 36,:756–767, 2013. URL https://www.sciencedirect.com/science/article/pii/S1084804512002597.

[4] Nick Weldin Gianluca Barbona, Michael Margolis, Filippo Palumbo, Franco Raimondi. Taking Arduino to the Internet of Things: The ASIP programming model. *Comunicaciones informaticas*, 89-90:128–140, 2016. URL https://www.sciencedirect.com/science/article/pii/S0140366416300743.

[5] J.Chandramohan A, R. Nagarajan, K. Satheeshkumar, N. Ajithkumar, P.A. Gopinath, and S. Ranjithkumar. Intelligent Smart Home Automation and Security System Using Arduino and Wi-fi. *International Journal Of Engineering And Computer Science*, (March), 2017. doi: 10.18535/ijecs/v6i3.53.

[6] Dani Sasmoko and Yanuar Arief Wicaksono. Implementasi Penerapan Internet of Things (IoT) Pada Monitoring Infus Menggunakan ESP 8266 Dan web Untuk Berbagi Data. *Jurnal Ilmiah nformatika*, 2(1):90–98, 2017.

[7] Poonam N. Railkar Mayur Sunil Jawalkar, Nayan Desale, Fanil Suratwala ,Amol Lamkhade. Intelligent Wildlife Tracking Using Ubiquitous Technological Suite. *International Journal of Synthetic Emotions (IJSE)*, 8:16, 2017. URL https://www.igi-global.com/article/intelligent-wildlife-tracking-using-ubiquitous-technological-suite/181640.