

concordance=TRUE

Sistema para visualización en la nube de la polución de la ciudad a través de Drones

Nombre del Estudiante: **Andres Julian Moreno Moreno**

1. Introducción

Los niveles de contaminación del aire en algunas ciudades son alarmantes. Hace tiempo que las autoridades buscan fórmulas para reducir las emisiones y mejorar la calidad del aire. Uno de los requisitos para que esto suceda es contar con una alta precisión en las mediciones.

Es por esta razón que se propone este proyecto, mediante el sistema planteado se pretende realizar el monitoreo de los niveles de contaminación en áreas críticas, como las zonas donde operan grandes fábricas. Este sistema tiene ciertas ventajas sobre las estaciones base que se usan habitualmente para esta tarea y también sobre los medidores portátiles. Un Drone puede recoger datos de distintas zonas, midiendo los niveles en varios puntos de forma mucho más rápida.

2. Descripción del proyecto

Para el proyecto de la materia se contemplan 2 temas de los propuestos en clase, Internet de las Cosas (IoT) y Computación en Nube con base en Wireless Sensor Networks (WSN) y el procesamiento masivo de datos procedentes de drones, el objetivo es monitorear la cantidad de smoke que hay a determinadas alturas sobre la ciudad en tiempo real, para esto se propone que un Drone integre un sensor de smoke y un sistema de comunicaciones basado en WSN que se comunicará a la estación terrena en tiempo real, estos datos de smoke una vez obtenidos por la estación terrena, estarán conectados a una plataforma en la nube, el usuario se podrá conectar a través de su navegador y podrá visualizar el monitoreo del smoke (monóxido de carbono CO) de acuerdo a la altura del Drone.

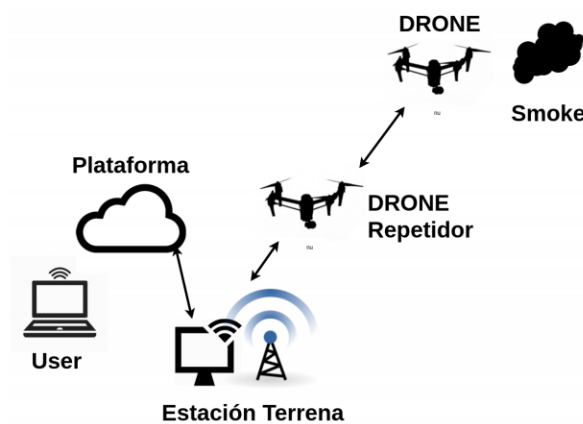
Adicionalmente se integrará un segundo Drone el cual actuará como nodo repetidor, una vez el primer Drone no esté al alcance de la estación terrena, esto permitirá a través de una red inteligente de comunicaciones poder aumentar el rango de medición del smoke y su visualización en tiempo real a través de un portal web.

El trabajo que se realizará comprende las fases o actividades descritas en la tabla 1. Cronograma. Se pretende obtener la cantidad de smoke que está presente en la ciudad a diferentes alturas, para esto se integrará un Drone con un sistema de sensor, que se comunicara con una estación remota, a través de un protocolo de comunicaciones inteligente que permita incluir si es necesario un segundo Drone que actuará como repetidor y aumentar el rango de medidas del Drone sensor.

Una vez instalado este sistema, la estación terrena estará conectada con una plataforma en la nube como servicio, la cual permitirá al usuario poder monitorear el smoke a través de una aplicación web. En la figura 1, (a) y (b) se presenta la aproximación gráfica de lo que se quiere lograr con el proyecto.



(a) Sistema planteado con 1 Drone.



(b) Sistema planteado con 2 Drones, 1 repetidor y el drone sensor.

3. Objetivos

3.1. General

- Visualizar a través de un servicio web, datos tomados en "tiempo real" de polución desde un Drone, basado en un sistema de comunicaciones wireless y conectado a una estación terrena.

3.2. Especificos

- Diseño e implementación de sistema de adquisición de datos y configuración del sensor de monóxido de carbono CO.
- Configuración y despliegue de sistema basado en la nube para adquisición de los datos del sensor y publicación en internet.
- Implementación de sistema de comunicaciones wireless con capacidad de nodo repetidor y transmisión de información en "tiempo real".
- Pruebas del sistema integrado, en tierra y si es posible de acuerdo a la disponibilidad de drones en el aire.
- Configuración de plataforma en la nube para visualización de datos.

4. Recursos Necesarios

A continuación se presenta la lista de los materiales y recursos necesarios para realizar el proyecto.

- Por lo menos un Drone para realizar el experimento de sensor smoke y comunicar a estación terrena.
- Sensor especializado para medir contaminación, específicamente monóxido de carbono.
- Sistema embebido para adquisición de datos y procesamiento de la información del sensor.
- Módulos de comunicaciones inalámbricos, con la posibilidad de instalar varios métodos de comunicación.
- Interfaz para conectar los módulos de comunicaciones al computador que hará las veces de estación terrena y subir los datos del sensor a internet.
- Plataforma en la nube, amazon web services o servicio en el CECAD.



(a) Drone disponible para pruebas



(b) Sensor de monóxido de carbono-MQ-7



(c) Modulo de comunicaciones Xbee

Figura 2: Recursos dentro de la Universidad Distrital.

5. Cronograma

En el cuadro 1 se presenta el cronograma propuesto para el desarrollo del proyecto, está planteado para desarrollarse en 10 semanas, cumpliendo con 8 actividades descritas en la cuadro 1, estas actividades son secuenciales y en muchas no se puede avanzar a la siguiente si no se tiene completada la actividad anterior.

| Actividades | Semana | | | | | | | | | |
|--|--------|----|----|----|----|----|----|----|----|----|
| | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 |
| Formulación, alcance y limitaciones del proyecto. | X | | | | | | | | | |
| Estudio de los sensores de Smoke. | | X | X | | | | | | | |
| Estudio de las plataformas de hardware para el proyecto. | | | X | X | X | X | X | X | | |
| Integración del sistema completo, Nodo Sensor de Smoke en drone. | | | | | X | X | X | X | | |
| Integración sistema de comunicaciones inteligente. | | | X | X | X | X | X | X | X | |
| Configuración servidor y plataforma como servicio. | | | | X | X | X | X | X | X | X |
| Pruebas de funcionamiento. | | | | X | X | X | X | X | X | X |
| Entrega Final del proyecto. | | | | | | | | | | X |

Cuadro 1: Cronograma de actividades

6. Desarrollo del Proyecto

A continuación se describe el desarrollo del proyecto planteado, presentando cada uno de los retos presentados durante el transcurso del mismo. Se describirá en cada subsección la fase de desarrollo de acuerdo al cronograma, describiendo la actividad desarrollada, describiendo el funcionamiento del mismo y presentando los debidos soportes del proceso.

En la Figura 3 se muestra la arquitectura general del sistema, y el desarrollo, funcionamiento y explicación del proyecto se mostrara a en las siguientes secciones documentando el proceso realizado, adicionalmente se adjuntan a esto documento, anexos los códigos fuentes documentados, así como las hojas de datos de los dispositivos usados.

El sistema propuesta esta compuesto por un sensor de CO, una plataforma Arduino UNO, un Drone IRIS, modulos de comunicaciones Xbee, un script desarrollado en processing, un sistema operativo debian 8, una plataforma en la nube Dweet.io para la adquisición de datos y una plataforma en la nube freeboard.io para la visualización en la nube.



Figura 3: Arquitectura del sistema

6.1. Sensor de Polución

De acuerdo a los niveles actuales de contaminación y acumulación de gases de combustibles para el desarrollo del proyecto se usa un sensor de monóxido de carbono, referencia MQ-7.

| MQ-7 Carbon Monoxide (CO) [3] | |
|--------------------------------------|--|
| Detección de la concentración de Gas | 10-1000 ppm CO |
| Voltaje de alimentación | 5V±0.2V |
| Sensibilidad | $R_s(\text{in air})/R_s(100\text{ppmCO})5$ |
| Gradiente de concentración | 0.6 |
| Condiciones ambientales Temperatura | 20±2 |
| Condiciones ambientales Humedad | 65%±5%RH |

Figura 4: Especificaciones técnicas Sensor

El monóxido de carbono es una sustancia tóxica que ingresa al cuerpo a través de la respiración. Es altamente peligroso porque no es detectable a través de los sentidos, es indispensable mantener bien ventilados los ambientes y la correcta instalación de los artefactos. Se produce debido a que todo material combustible rico en carbono (gas, petróleo, carbón, kerosén, nafta, madera, plásticos) necesita oxígeno para quemarse. cuando la cantidad de oxígeno es insuficiente, la combustión es incompleta y se forma monóxido de carbono(CO).

El MQ7 es un sensor de monóxido de carbono (CO) de uso simple, adecuado para la detección de concentraciones de CO en el aire. Puede detectar desde 10 a 1000 ppm(Partes por Millón). Cuenta con dos señales de salida, una digital a través de un comparador con umbral ajustable y otra analógica para poder conectarlo a cualquier microcontrolador y realizar la medición.En la siguiente imagen se muestran las características técnicas principales del sensor seleccionado para el proyecto.

El proyecto resultará exitoso si el sistema es capaz de medir la cantidad de CO en el ambiente a través de un Drone y registrarlo en “tiempo real” en una plataforma web. El monóxido de carbono presenta su medida mas alta cuando se presenten quema de combustible y poco oxígeno, esto indica la necesidad urgente de realizar las medidas de la variable en una ciudad debido al alto flujo de vehículos que usan combustibles.

6.2. Plataforma de hardware para el proyecto y sistema de adquisición de datos integrado

El sistema embebido seleccionado para el proyecto es una plataforma de prototipado rápido llamada Arduino UNO. En la siguiente imagen se describen sus características técnicas principales, el software desarrollado para el funcionamiento del proyecto está basado en el lenguaje de programación propio de Arduino que está basado en C. Para la programación se uso el IDE de Arduino, una interfaz basada en Java para desarrollar software.

El sistema de adquisición de datos depende directamente del sensor y la forma de conexión al Arduino, el sensor cuenta con 4 cables de los cuales 2 son para el sistema de alimentación de energía y tierra, las otras dos conexiones corresponden a una salida analógica y una digital que se conectan en diferentes puertos del

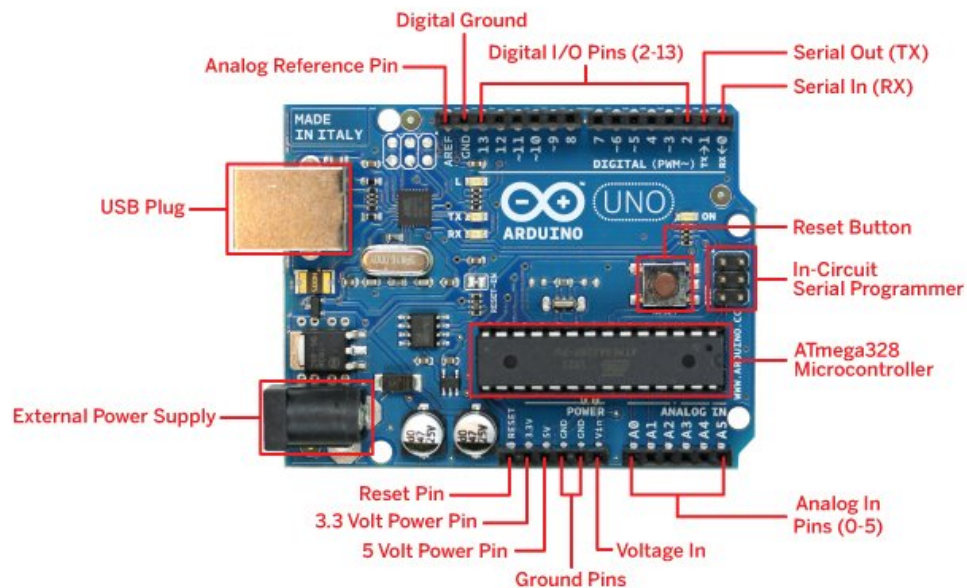


Figura 5: Sistema embebido Arduino UNO V2

Arduino de acuerdo al tipo, la analógica al puerto A0 y la digital al puerto D8.

A continuación se muestra el software realizado para el funcionamiento del sistema, se basa en la lectura de CO en ppm, la función principal del software es adquirir la información del sensor y comunicar a través de un puerto serial al computador para su visualización o un puerto de comunicaciones serial de otro dispositivo. El software se encuentra documentado para mayor entendimiento.

```

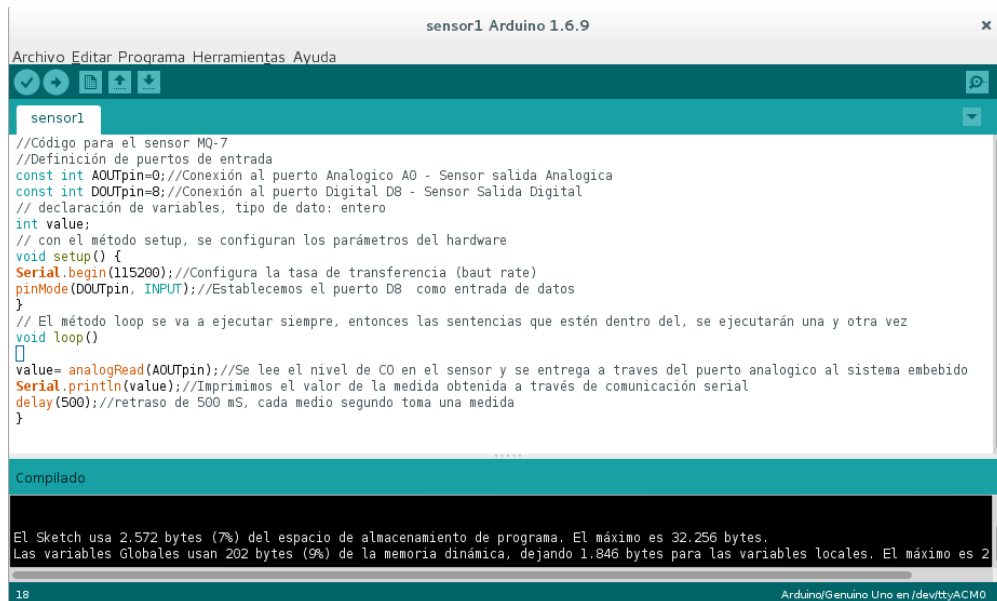
1  const int AOUTpin=0;
2  const int DOUTpin=8;
3  int value;
4  void setup()    {
5      Serial.begin(115200);
6      pinMode(DOUTpin, INPUT);
7  }
8  void loop()    {
9      value= analogRead(AOUTpin);
10     Serial.println(value);
11     delay(500);
12 }

```

6.3. Drone o Vehículo Aéreo No Tripulado VANT

Actualmente los VANT o UAV, por sus siglas en inglés (Unmanned Aerial Vehicle) hacen parte de la solución a diversos problemas industriales, comerciales y educativos, esto ha impulsado en el mercado la compra y venta de estos (“drones”). La utilización de estos productos depende del tipo de aplicación y necesidades del usuario final las cuales se pueden ver limitadas por las características del equipo adquirido y la clase del mismo.

Los VANT, han evolucionado en el mundo desde la Primera Guerra Mundial, en anteriores décadas se encontraban muy centrados en entornos militares. Entonces, se utilizaban como una forma de volar más segura ya que no requerían de piloto. Por esta razón, se ha buscado hacer que las aeronaves fueran más pequeñas, aerodinámicas, con menos consumo de combustible y a la vez más ágiles. Hoy, todas esas características



```
sensor1 Arduino 1.6.9
Archivo Editar Programa Herramientas Ayuda
sensor1
//Código para el sensor MQ-7
//Definición de puertos de entrada
const int AOUTpin=0;//Conexión al puerto Analógico A0 - Sensor salida Analógica
const int DOUTpin=8;//Conexión al puerto Digital D8 - Sensor Salida Digital
// declaración de variables, tipo de dato: entero
int value;
// con el método setup, se configuran los parámetros del hardware
void setup() {
  Serial.begin(115200);//Configura la tasa de transferencia (baud rate)
  pinMode(DOUTpin, INPUT);//Establecemos el puerto D8 como entrada de datos
}
// El método loop se va a ejecutar siempre, entonces las sentencias que estén dentro del, se ejecutarán una y otra vez
void loop()
{
  value= analogRead(AOUTpin);//Se lee el nivel de CO en el sensor y se entrega a través del puerto analógico al sistema embebido
  Serial.println(value);//Imprimimos el valor de la medida obtenida a través de comunicación serial
  delay(500);//retraso de 500 mS, cada medio segundo toma una medida
}

Compilado

El Sketch usa 2.572 bytes (7%) del espacio de almacenamiento de programa. El máximo es 32.256 bytes.
Las variables Globales usan 202 bytes (9%) de la memoria dinámica, dejando 1.846 bytes para las variables locales. El máximo es 2

18 Arduino/Genuino Uno en /dev/ttyACM0
```

Figura 6: Compilación del código en el IDE de Arduino

siguen siendo extremadamente útiles, pero además se ha expandido su campo de acción.

Un VANT, con mayor o menor grado de inteligencia, puede comunicarse con su controlador para devolverle datos de una imagen (térmica, óptica, etc.), así como también información referente a su estado, posición, velocidad del aire, orientación, altitud o cualquier parámetro de su telemetría. Generalmente, estos vehículos están armados con sistemas que en caso de fallo de cualquiera de sus componentes o programas toman medidas correctivas o alertan al operador. En los casos más inteligentes, son capaces de tomar medidas “imaginativas” para afrontar un problema no esperado. Como en todos los campos, se pueden clasificar los VANTs utilizando multitud de características. Tamaño, autonomía, forma de sustentación o utilidad.

El Drone usado para este proyecto es el Iris fabricado por la empresa 3DRobotics [9], es un quad-copter autónomo que podemos manejar de manera remota o bien simplemente dibujar en nuestro smartphone la ruta que queremos que haga y él solo se encarga de seguir el patrón dibujado en nuestro smartphone o incluso de seguir un objeto que se esté moviendo.

Se puede controlar vía un dispositivo Android, móvil o tablet, y prometen aplicación para iOS además de con un mando de nueve canales vía RF, incluido en el kit. Si se maneja desde la aplicación podrás despegar y aterrizar con un solo botón en pantalla y puedes dibujar una ruta en el mapa que ves en pantalla e Iris comenzará a seguirla.

Este quad-copter integra una cámara entre sus brazos frontal y muestra una vista en tiempo real al usuario. Además, para los usuarios más exigentes dispone de un soporte para montar una cámara GoPro Hero3. Tiene una autonomía de entre 9 y 14 minutos y un peso de 1,2 kg.

el Drone integra un sistema de piloto automático gracias al procesador Cortex-M4 que ejecuta un sistema operativo que controla todos los parámetros de vuelo basándose en los datos de los sensores de ST Microelectronics. El código de vuelo software libre que integra es el fruto del desarrollo y prueba de una comunidad de más de 40.000 usuarios.

Se puede programar una ruta basándonos en puntos de control situados mediante su posición GPS que permitirá realizar un seguimiento preciso para crear mapas de zonas, investigación científica e incluso para tomas para cine.



Figura 7: Drone Iris de la empresa 3DRobotics

6.4. Sistema de comunicaciones

El sistema de comunicaciones propuesto tiene como objetivo comunicar el sistema de adquisición de datos ubicado en el Drone y la estación terrena ubicada en un computador personal y con conectividad a internet, se deben entregar los datos de las mediciones realizadas en el Drone, para esto se usan módulos Xbee [], estos modulos se configuran con un protocolo basado en topología mesh, y usan un protocolo de comunicaciones propio del fabricante llamado Digimesh.

Una red de comunicaciones digital Digimesh, se basa en autogestión, su topología es mesh como su nombre lo indica, esto nos demuestra que la red es inteligente, permite que se añadan nuevos nodos a la red si están configurados debidamente, si algún nodo que compone la red falla rápidamente se remplazan las rutas de envío de información, esto lo realiza automáticamente.

Para el proyecto es necesario contar con este tipo de red debido a los requerimientos en campo de comunicaciones y los posibles eventos que se produzcan. La primera red que se configura es de 3 nodos, gracias a los nodos sensores fabricados y un nodo central. Cada nodo de comunicaciones está compuesto por un módulo de comunicaciones Xbee 900 Hp el cual se comunica a través de la red digimesh con sus vecinos. **Estos módulos son configurados desde el IDE propietario X-CTU**, En la siguiente imagen se muestra el software cuando se configuran los módulos de comunicaciones.

Para la configuración de la red Digimesh es necesario realizar el siguiente procedimiento:

- Identificación de la Red.

El ID de la red es fundamental para agrupar los nodos en una respectiva red. Los módulos están por defecto de fábrica configurados con el ID 0x7FFF, Se debe utilizar un ID diferente por motivos de seguridad, ya que esta dirección viene configurada por defecto en los módulos de comunicaciones Xbee y podría ocasionar que ingrese un nodo extraño o que no esté autorizado a la red. Para la red inicialmente la ID Seleccionada es 0x1FFF.

- Canal – Secuencia de salto.

El comportamiento del canal en Digimesh depende de la tecnología usada. En el caso del Proyecto se usaran módulos de comunicaciones Xbee 900HP, estos módulos funcionan a la frecuencia de 900 MHz. La opción de ajuste que se debe configurar en cada uno de los módulos es HP, este ajuste define el patrón, todos los módulos de la red deben tener el mismo ajuste. Estos dispositivos siempre están saltando entre frecuencias, algunos canales con ruido o malos no rompen la comunicación del sistema. La configuración de un xbee 900Hp. Para la red la configuración ID HP = 1.

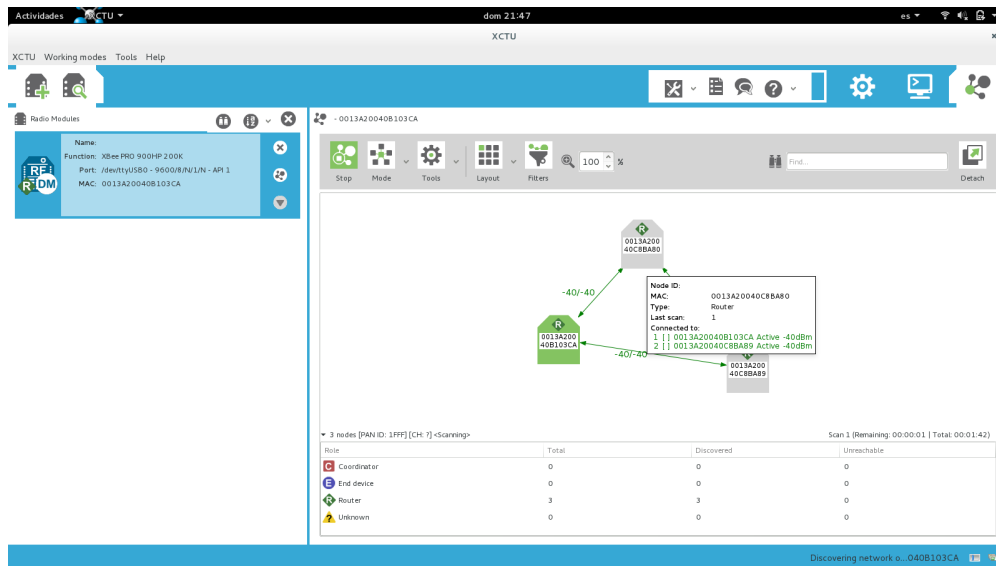


Figura 8: X-CTU programador de los módulos Xbee

- Modo sueño (SM). Debido a las características del proyecto es necesario realizar un consumo mínimo de potencia, para esto es necesario que los módulos entren en etapa de sueño cuando sea necesario. Actualmente no es necesario configurar este parámetro en el módulo de comunicaciones debido a que se encuentra en fase de pruebas y desarrollo. El parámetro SM configurado temporalmente es SM=0. En el futuro se implementará esta configuración por medio de software o por intervención del microprocesador central.

Se configuraron exitosamente 3 módulos de comunicaciones los cuales son encargados de establecer la red en topología mesh y enviar los datos del sensor a tierra.

6.5. Configuración de sistema PaaS

Para la visualización de los datos del sensor CO en internet se usan la plataforma **Dweet.io** que ofrece un servicio gratuito para enviar y representar datos en la nube, adicionalmente se usa la plataforma **freeboar.io** para representar los datos mediante diagramas.

Para subir los datos se realiza de la siguiente manera mediante protocolo http.

$http : //dweet.io/dweet/for/ajmorenom?CO = 50$

Para visualizar los datos lo podemos hacer en el siguiente enlace:

$http : //dweet.io/follow/ajmorenom$

Gracias a la plataforma el proceso y protocolo de comunicación es transparente para el usuario.

Para actualizar los datos constantemente se realiza un Script en el software para desarrollo basado en Arduino llamado Processing, Processing es un lenguaje de programación y entorno de desarrollo integrado de código abierto basado en Java, de fácil utilización, y que sirve como medio para la enseñanza y producción de proyectos multimedia e interactivos de diseño digital. El script desarrollado se muestra a continuación.

```
2 import processing.serial.*;
3 int lf = 10;
4 Serial myPort;

6 void setup()
7 {
8     println(Serial.list());

10     myPort = new Serial(this, Serial.list()[4], 115200);
11 }

12 void draw()
13 {
14     while (myPort.available() > 0)
15     {
16         String lectura = myPort.readStringUntil(lf);
17         if (lectura != null)
18         {
19             println(lectura);
20             loadStrings("https://dweet.io/dweet/for/ajmorenom?CO="+lectura);
21         }
22     }
23 }
24 }
```

La plataforma funcionando se puede observar en la figura 9

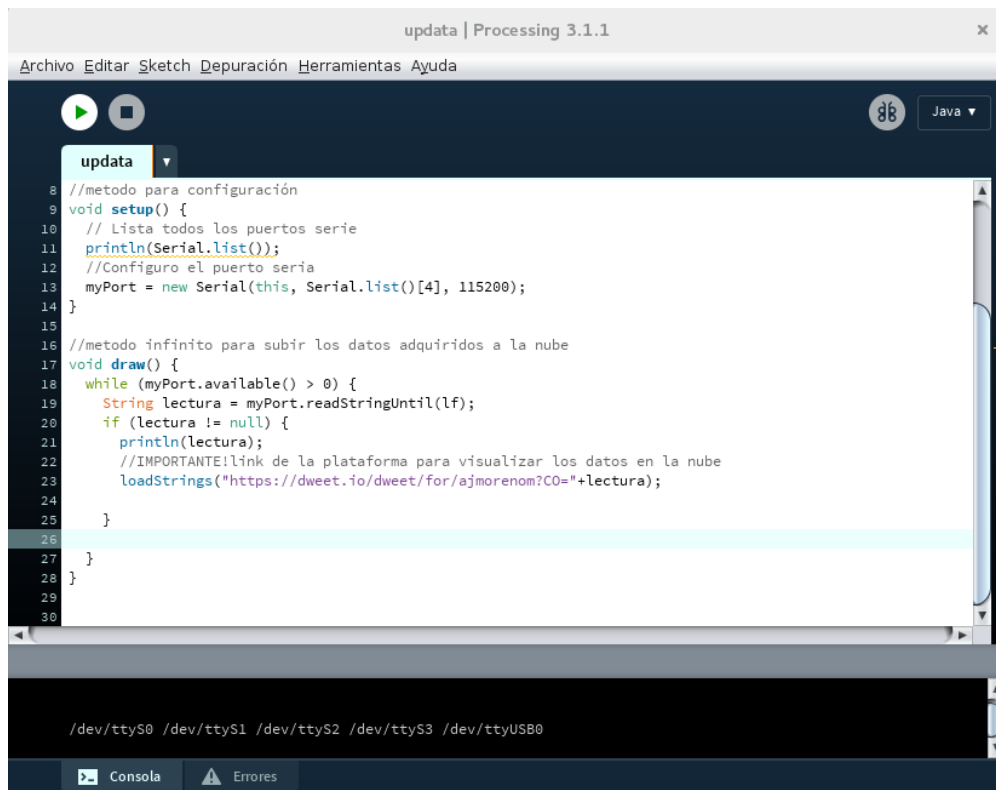


Figura 9: Processing script para subir datos a la nube

7. Pruebas del sistema

Una vez se cuenta con todo el sistema configurado e instalado, se procede a realizar las pruebas de funcionamiento, si en la plataforma observamos que se miden en el sensor, en la figura 10 se muestra el resultado obtenido, obteniendo las mediciones que realiza el sistema y logrando visualizarlas a través de internet gracias las plataformas de IoT, Dweet.io y freeboard.io, estas plataformas permiten subir los datos desde un terminal y visualizarlos a través de internet, son plataformas libres, sin embargo si se desea acceder al historial de los datos o opciones avanzadas se debe pagar por usar estas características.

En la figura 10, se observan los datos obtenidos, y visualizados en "tiempo real".

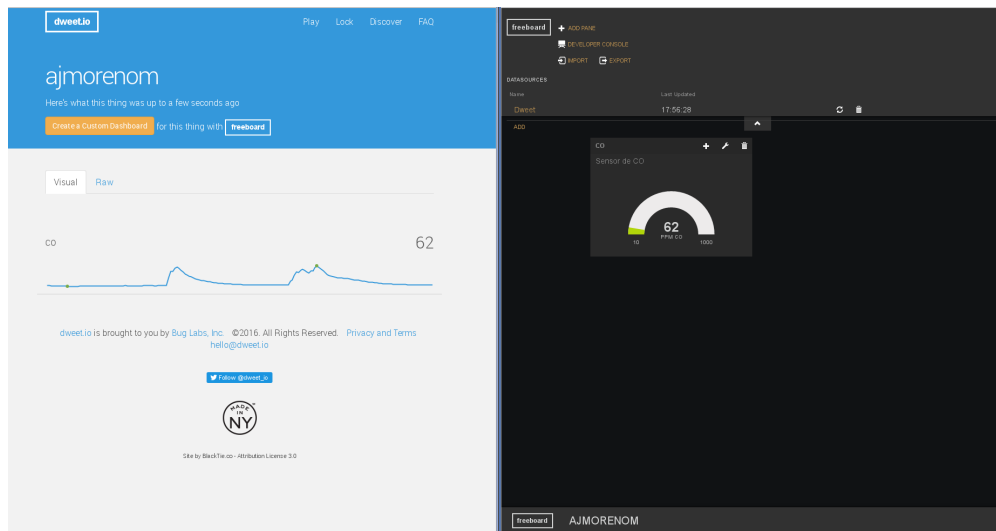


Figura 10: Resultados visualizados sobre plataforma en internet

8. Resultados Esperados

Para que el proyecto sea un éxito, se debe cumplir con los siguientes resultados:

- Monitoreo en tiempo real del smoke, a través de un Drone y una plataforma en la nube.
- Configurar un sistema que permita al Drone aumentar la distancia y altura de sus medidas.
- Establecer comunicación desde la estación terrena con el Drone a través de un Drone repetidor de comunicaciones.

9. Análisis Dinámico de Datos

Para trabajar con documentos dinámicos en Knitr, primero se cargan las siguientes librerías en RStudio.

```
library(curl)
library(httr)

##
## Attaching package: 'httr'
## The following object is masked from 'package:curl':
##
##   handle_reset

library(jsonlite)
```

```
## Loading required package: methods
```

```
library(knitr)
library(stringr)
```

En el anterior chunk de R, se observa las respuestas de RStudio al cargar las librerías necesarias, a continuación se obtiene el archivo JSON de la plataforma Dweet.io, se observa la función GET Usada e información del archivo obtenido de la plataforma en la nube.

```
r <- GET("https://dweet.io:443/get/dweets/for/ajmorenom")
headers(r)
```

En el siguiente chunk, se observa el contenido del archivo JSON.

```
headers(r)
str(content(r))
```

Como se trata de un documento dinámico la variable CO y created date, varían de acuerdo a la fecha que se ejecute el chunk de código, esto quiere decir que los datos obtenidos de la plataforma serán diferentes de acuerdo al avance del tiempo.

En el siguiente chunk se guardan los datos del archivo descargado JSON en una variable, se convierte a una tabla de datos, y se obtiene la información organizada del contenido para su posterior análisis exploratorio.

```
library(jsonlite)
urldew<- "https://dweet.io:443/get/dweets/for/ajmorenom"
urldew<- paste(urldew, sep = "")
dwet<- fromJSON(urldew)
dwet

## $this
## [1] "succeeded"
##
## $by
## [1] "getting"
##
## $the
## [1] "dweets"
##
## $with
##      thing              created CO
## 1 ajmorenom 2016-07-06T14:21:01.353Z 37
## 2 ajmorenom 2016-07-06T14:21:00.343Z 37
## 3 ajmorenom 2016-07-06T14:20:58.338Z 37
## 4 ajmorenom 2016-07-06T14:20:57.324Z 37
## 5 ajmorenom 2016-07-06T14:20:55.334Z 37
```

Una vez obtenidos los datos de CO, se procede a realizar un análisis exploratorio de la variable seleccionada o de interés, en este caso las partes por millón de contaminación en monóxido de carbono CO, se realiza una captura de los últimos 5 datos de la plataforma, en los últimos 5 segundos, esto debido a las características de la información que permite descargar la plataforma en la nube, se realiza un análisis de la información y se calcula, el valor mínimo, máximo, primer cuartil, tercer cuartil y media.

```
datos<-dwet$with$content
dim(datos)
```

```
## [1] 5 1

head(datos)

##      CO
## 1 37
## 2 37
## 3 37
## 4 37
## 5 37

names(datos)

## [1] "CO"

str(datos)

## 'data.frame': 5 obs. of  1 variable:
##  $ CO: int  37 37 37 37 37

summary(datos)

##           CO
##  Min.      :37
## 1st Qu.:37
##  Median :37
##   Mean  :37
## 3rd Qu.:37
##   Max.   :37
```

Se debe tener en cuenta que siempre que se ejecute el chunk, procesa los últimos 5 datos obtenidos por el Drone de polución.

10. Conclusiones

- Se logra realizar el monitoreo en tiempo real del smoke, a través de un drone y una plataforma en la nube, se visualizan las medidas de CO en ppm.
- Se logra configurar un sistema integrado capaz de aumentar el alcance del Drone, y entrega las medidas a la estación terrena.
- Se logra establecer comunicación desde la estación terrena con el Drone a través de un Drone repetidor de comunicaciones, esto basado en el sistema propuesto basado en topología mesh.
- el uso del drone facilita realizar este tipo de mediciones.

11. Trabajo futuro

Posterior al proyecto desarrollado se debe hacer un análisis de las medidas para obtener los datos de polución de acuerdo al vuelo del Drone, además de integrar una nueva variable al sistema la cual es la posición en el espacio a través de un GPS y la altura. con esto se puede estructurar mucho mejor los datos obtenidos de polución en una ciudad o ambiente determinado.

12. Bibliografía

Accesos WEB

- Software para configuración de módulos Xbee
- Hoja de Datos del Sensor de CO MQ-7
- Arduino UNO
- Software de desarrollo Processing
- Fabricante de Drones 3DRobotics
- Instalación de Arduino en Debian 8
- Plataforma en la nube Dweet
- Plataforma en la nube Freeboard
- Instalación XCTU en Debian 8

Referencias

- [Barrientos, 2009] *Vehículos aéreos no tripulados para uso civil. Tecnología y aplicaciones..* Pages 1–29.
- [Zhou, 2005] *Control of an inspection robot for 110KV power transmission lines based on expert system design methods..* Proceedings of 2005 IEEE Conference on Control Applications, 2005. CCA 2005.
- [Mejías, 2007] *COLIBRI: A vision-guided UAV for surveillance and visual inspection..* Proceedings - IEEE International Conference on Robotics and Automation, (April), 2760–2761.
- [Jones, 2009] *Power line inspection an UAV concept. .* 23rd International Conference Image and Vision Computing New Zealand, IVCNZ, 1–6