

MONITOREO TCP/IP EN LABVIEW© DE UNA SEÑAL ULTRASÓNICA MEDIANTE UN SISTEMA IoT, UTILIZANDO EL SISTEMA ARDUINO EN MODO SERVIDOR CON EL SHIELD DE ETHERNET

MONITORING TCP/IP IN LABVIEW© OF A ULTRASONIC SIGNAL IN AN IoT SYSTEM, BY USING ARDUINO IN SERVER MODE WITH ETHERNET SHIELD

Eric Mario Silva Cruz

Instituto Tecnológico de Oaxaca
ericmsc@hotmail.com

Francisco Adolfo Silva Cruz

Instituto Tecnológico de Oaxaca
francisco.adolfo.sc@gmail.com

Carolina Rosas Huerta

Universidad Politécnica de Puebla
carolina.rosas@uppuebla.edu.mx

Franco Gabriel Caballero Julián

Instituto Tecnológico de Oaxaca
francogcaballero@gmail.com

Resumen

En todo el mundo, hay una tendencia a desarrollar sistemas basados en Internet de las cosas (*Internet of Things IoT*) con interconectividad a través del protocolo TCI/IP, lo que ha llevado a la evolución en el estándar IPV6. Las perspectivas de desarrollo incluyen una infinidad de sistemas físicos, que incluyen aplicaciones en el hogar, la oficina y los vehículos, y están destinados a fortalecer el avance en la Industria 4.0, aplicando esta filosofía a los sistemas ciberfísicos e interconectados. En el presente artículo, mostramos los elementos básicos necesarios para desarrollar sistemas de IoT y monitorización de una señal analógica, mediante la integración del sistema Arduino y su Shield de Ethernet, que está conectado a la red a través de una computadora y comparte el protocolo de Internet (IP) de forma dinámica. El Arduino está configurado como un servidor y

es responsable de recibir la señal del sensor ultrasónico y enviar los datos a la red. En lo que corresponde a la monitorización de las señales que se envían a través de TCP / IP, el software LabVIEW se usa para diseñar una instrumentación virtual, la cual accede a la dirección IP del servidor Arduino para obtener y procesar los datos transmitidos.

Palabra(s) Clave: Arduino Server, Ethernet, LabVIEW, TCP/IP.

Abstract

All over the world exist a tendency to develop systems based on the Internet of Things (IoT) with interconnectivity via the TCI / IP protocol, which has led to evolution in the IPV6 standard. The development prospects include an infinity of physical systems, which include applications in the home, office, vehicles, and are intended to strengthen the advancement in Industry 4.0, applying this philosophy to cyber-physical and interconnected systems. In the present article, are detailed the basic elements to develop IoT systems for analog signal monitoring, through the integration of the Arduino system and its Ethernet Shield, which is connected to the network through a computer and sharing the internet protocol (IP) dynamically. The Arduino is configured as a server and is responsible of receiving the signal from the ultrasonic sensor and sending the data to the network. In what corresponds to the monitoring of the signals that are sent via TCP / IP, the LabVIEW software is used to design a virtual instrumentation, which access the IP address of the Arduino server to obtain and process the transmitted data.

Keywords: Arduino Server, Ethernet, LabVIEW, TCP/IP.

1. Introducción

Protocolo TCP/IP

El protocolo de comunicación TCP/IP permite que un equipo pueda comunicarse en una red. TCP/IP provee conectividad de extremo a extremo especificando como los datos deberían ser formateados, direccionados, transmitidos, enrutados y recibidos por el destinatario. TCP/IP sigue reglas básicas basadas en el modelo de capas de comunicaciones OSI (*Open Systems*

Interconnection), figura 1, en la cual se observa que el protocolo TCP/IP utiliza únicamente 4 capas, que son: la capa de acceso a la red, la capa de internet, la capa de transporte y la capa de aplicación (Crespo, 2018).

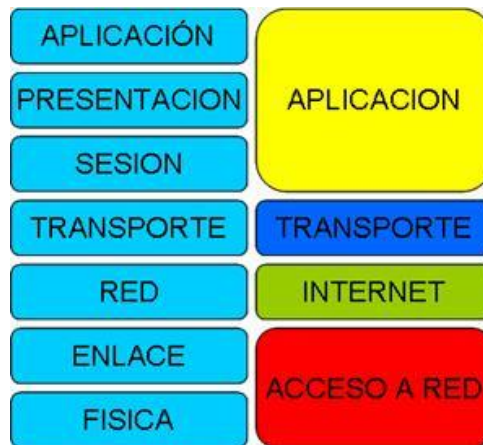


Figura 1 Modelo OSI para TCP/IP, (Crespo 2018).

Arduino

La plataforma es un sistema versátil el cual puede conectarse al protocolo TCP/IP mediante su accesorio (Shield) de Ethernet, figura 1. Este Shield nos permite acceder a la capa de acceso a red (ethernet o WiFi), internet (IP) y transporte. La capa de aplicación deberemos implementarla dentro de Arduino, ya sea directamente o mediante una librería, (Crespo, 2018).

El método de conexión se realiza mediante la plataforma Arduino conectada al Shield de Ethernet, la cual puede establecer comunicación directamente con una IP fija, que le es asignada al conectarse directamente a un concentrador (Hub) ó bien utilizando el puerto Ethernet de una computadora, esta segunda opción requiere la configuración del acceso a internet, en la cual se comparte la conexión mediante una DHCP (*Dynamical Host Configuration Protocol*).

TCP/IP utiliza un identificador o dirección IP, con una longitud de 32 bytes. La IP identifica a la red a la que pertenece un dispositivo como a él mismo dentro de dicha red y especifica la conexión entre redes, (Insam, 2003).

Se estructuran mediante cuatro octetos, escritos en formato decimal, separados por puntos. Existen ciertas direcciones IP que no están asignadas y que se

denominan privadas, que suelen ser utilizadas por los hosts que usan traducción de dirección de red (NAT) para conectarse a una red pública o por los hosts que no se conectan a Internet.

En una misma red no pueden existir dos direcciones iguales, pero sí se pueden repetir en dos redes privadas que no tengan conexión entre sí o que se conecten mediante el protocolo NAT. Las direcciones privadas son las que se muestran en la tabla 1.

Tabla 1 Direcciones privadas.

Clase	Direccionamiento	Usos
A	10.0.0.0 a 10.255.255.255 (8 bits red, 24 bits hosts)	
B	172.16.0.0 a 172.31.255.255 (16 bits red, 16 bits hosts)	16 redes clase B contiguas, uso en universidades y grandes compañías.
C	192.168.0.0 a 192.168.255.255 (24 bits red, 8 bits hosts).	256 redes clase contiguas, uso de compañías medianas y pequeñas además de pequeños proveedores de internet (ISP).

Subredes IP

Las subredes IP deben identificarse usando una máscara de subred, dicha máscara es de cuatro bytes, para obtener el número de subred se realiza una operación AND lógica entre ella y la dirección IP de algún equipo. La máscara de subred deberá ser la misma para todos los equipos de la red IP.

El enrutamiento sirve para alcanzar redes distantes y la comunicación entre subredes. Las subredes son redes físicas independientes que comparten la misma dirección IP de la red principal, de forma que las subredes comparten el mismo identificador de red, pero se sitúan en redes físicas diferentes y que podrán comunicarse usando una máscara de dirección, como puede observarse en el ejemplo de la tabla 2.

Tabla 2 Máscara de dirección.

Dirección IP	Máscara de Subred	Ubicación del equipo
148.206.257.2	255.255.255.0	148.206.257.0

Una dirección IP dinámica es una IP asignada mediante un servidor DHCP (Dynamic Host Configuration Protocol) al usuario. La IP que se obtiene tiene una duración máxima determinada. El servidor DHCP provee la dirección IP del cliente. El servidor del servicio DHCP puede ser configurado para que renueve las direcciones asignadas cada tiempo determinado.

Dirección MAC

Una parte importante de la conexión vía la plataforma Shield de Arduino es el direccionamiento MAC (*Media Access Control*), es un identificador de 48 bits conformada por 6 bloques hexadecimales, cada MAC corresponde de forma única a una tarjeta o dispositivo que se conecta a la red, se conoce también como dirección física. La configuración MAC fue establecida y configurada por el IEEE (los últimos 24 bits) y el fabricante (los primeros 24 bits) utilizando el *organizationally unique identifier*.

La mayoría de los protocolos que trabajan en la capa 2 del modelo OSI usan una de las tres numeraciones manejadas por el IEEE: MAC-48, EUI-48, y EUI-64, y son identificadores globalmente únicos. No todos los protocolos de comunicación usan direcciones MAC, y no todos los protocolos requieren identificadores globalmente únicos.

Las direcciones MAC son únicas a nivel mundial, puesto que son escritas directamente, en forma binaria, en el hardware en su momento de fabricación.

En Arduino la dirección MAC está establecida con el número: 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED. Una vez que se ha habilitado una IP para el sistema Arduino, es posible conectar una infinidad de sensores, los cuales proveen información del mundo físico hacia los usuarios, entre los cuales se encuentra sensores de distancia, de temperatura, presión, velocidad, humedad, etc.

En nuestro caso se utilizó un sensor ultrasónico para transmitir la señal obtenida, aplicando el sensor ultrasónico HC-SR04, el cual puede detectar objetos, distancia o nivel en un rango mínimo de 2 cm y un máximo de 400 cm. Se eligió usar dicho sensor ya que es ideal para trabajar con Arduino, además que puede ser también utilizado con cualquier procesador lógico que funcione a 5 V, figura 2.



Figura 2 Sensor Ultrasonico HC-SR04.

El objetivo principal es generar dispositivos para Internet de las cosas, considerando no solo variables personales, sino también la interconexión de redes de sensores establecidas en procesos industriales, los cuales requieren un seguimiento por cada área o etapa que integra una empresa.

El sistema de monitoreo se diseña en la plataforma de LabVIEW, (LabVIEW, 2015), diseñando una instrumentación virtual para obtener la información que se genera en la capa de aplicación en el cual se activa y desactiva la conexión TCP/IP, con el requerimiento de que la instrumentación virtual tenga acceso a la dirección del servidor que es el Arduino y establecer conexión mediante el puerto 8000.

2. Métodos

Conexión mediante red compartida

El primer paso para conectar el sistema Arduino a Ethernet fue configurar la conexión compartida con la computadora, de manera que el acceso se realizó mediante DHCP, (Margolis, 2011). En la figura 3, se observa la conexión del Shield de Ethernet, (Ruiz, 2013).



Figura 3 Conexión al puerto Ethernet de una computadora.

El proceso de configuración se inicia reconociendo las redes activas, en nuestro caso observamos la Red Pública en el ejemplo se observa la red Wi-Fi 3, la cual es el acceso de la computadora vía inalámbrica, como se observa la figura 4. Se observa también la red identificada como Ethernet 4, la cual es el acceso de Arduino a la computadora y su acceso a la red compartida.

Ver información básica de la red y configurar conexiones

Ver las redes activas

INFINITUMA7794C

Red pública

Tipo de acceso: Internet

Conexiones: Wi-Fi 3 (INFINITUMA7794C)

Red no identificada

Red pública

Tipo de acceso: Sin acceso a la red

Conexiones: Ethernet 4

Figura 4 Configuración de red compartida.

Se configura el modo compartido de Wifi de la red Wi-Fi 3, para dicho propósito primero se accedió a la configuración de la red, tal como se observa en la figura 5, en donde las propiedades de la red inalámbrica se activan las casillas de verificación.

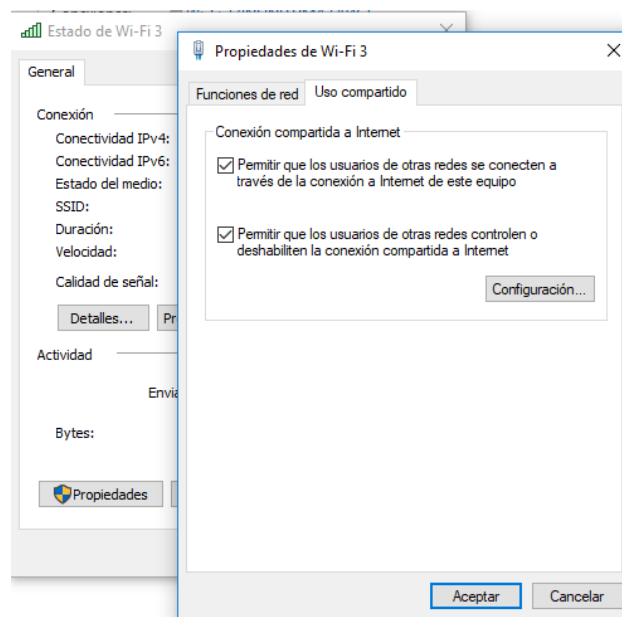


Figura 5 Configuración de la red inalámbrica disponible.

Se procede a configurar el acceso a Ethernet mediante el acceso de la red, en nuestro caso la red Ethernet 4, en donde se accede al protocolo IPV4, y se muestra la dirección IP y la máscara de subred. En caso de no existir valores asignados se pueden emplear los valores mostrados en la figura 6.

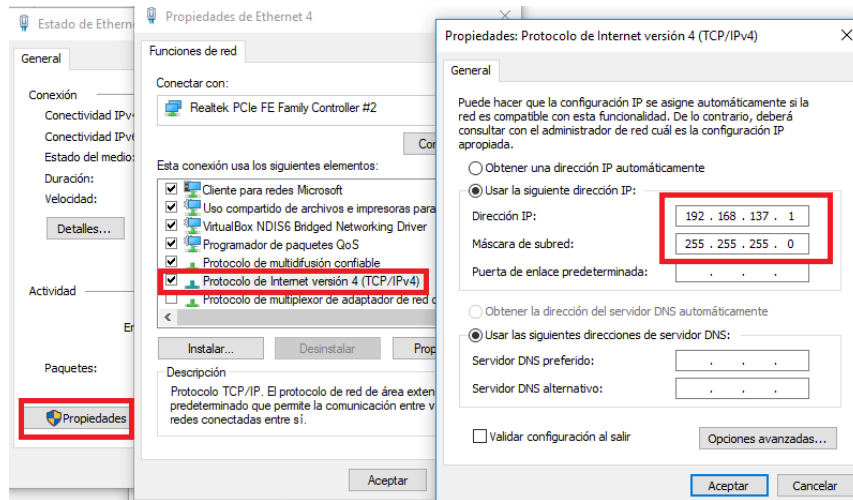


Figura 6 Configuración Ethernet para el puerto de acceso para Arduino.

Para comprobar el acceso de la red compartida se accede al símbolo de sistema (CMD), el cual nos permite ver los dispositivos conectado al teclear la siguiente configuración:

```
C:\Users\venovo 76331>ipconfig/all
```

Se debe observar que el dispositivo este habilitado para DHCP

Configuración IP de Windows

Adaptador de LAN inalámbrica Conexión de área local 2:*

Estado de los medios. : medios desconectados

Sufijo DNS específico para la conexión. . . :

Descripción : Microsoft Wi-Fi Direct Virtual Adapter #12

Dirección física. : 78-45-61-CC-56-1F

DHCP habilitado : sí

Configuración automática habilitada . . . : sí

Encontramos el adaptador de Ethernet 4, el cual es el acceso al puerto Ethernet de la computadora:

Adaptador de Ethernet Ethernet 4:

De manera particular se observa la dirección IPV4 y su Máscara de Subred

Sufijo DNS específico para la conexión. . . :

Descripción: Realtek PCIe FE Family Controller #2

Dirección física.: C8-5B-76-4B-D8-34

DHCP habilitado: no

Configuración automática habilitada . . : sí

Dirección IPv4. : 192.168.137.1(Preferido)

Máscara de subred : 255.255.255.0

Puerta de enlace predeterminada . . . :

NetBIOS sobre TCP/IP. : habilitado

De la misma manera podemos observar que el adaptador inalámbrico está habilitado:

Adaptador de LAN inalámbrica Wi-Fi 3:

Sufijo DNS específico para la conexión. . : lan

Descripción . . . : Realtek RTL8723BE Wireless LAN 802.11n PCI-E NIC #2

Dirección física.: 78-45-61-CC-56-1F

DHCP habilitado: sí

Configuración automática habilitada . : sí

Dirección IPv4. : 192.168.1.115(Preferido)

Máscara de subred : 255.255.255.0

Concesión obtenida. : viernes, 7 de septiembre de 2018 23:52:21

La concesión expira : sábado, 8 de septiembre de 2018 23:52:21

Puerta de enlace predeterminada . . . : 192.168.1.254

Servidor DHCP : 192.168.1.254

Servidores DNS. : 192.168.1.254

NetBIOS sobre TCP/IP. : habilitado

Librería de Arduino DHCPAddresPrinter

El código de Arduino DHCPAddresPrinter se usa para verificar el correcto funcionamiento de la IP con la cual está conectado el Arduino, (Margolis, 2011), esta librería se accede en el IDE de Arduino que se detalla en la figura 7.

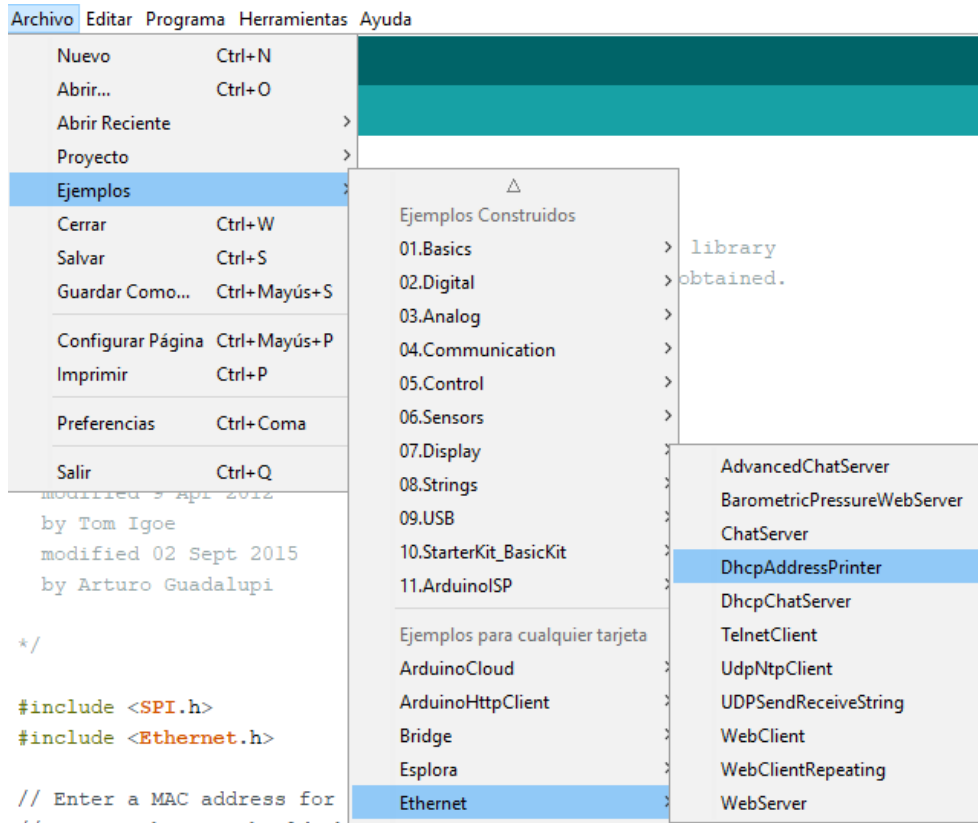


Figura 7 Comprobación de que el Arduino tiene acceso a la red.

Finalmente, la ejecución de dicho código refleja la IP dinámica que se le asigna a la tarjeta Arduino, por ejemplo, en nuestro caso es: 192,168,137,205.

La obtención de la IP dinámica nos permite configurar la transmisión de datos hacia el puerto Ethernet aplicando el siguiente código para Arduino, (Seneviratne, 2015):

```
#include <SPI.h>
#include <Ethernet.h>
// sensor ultrasónico
const unsigned int PTRG = 7; PECHO = 8;
const unsigned int BAUD_RATE = 9600;
unsigned long k=0;
byte mac [] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED }; // MAC de Arduino
IPAddress ip(192,168,137,205); // dirección IP dinámica
EthernetServer server(8000); // Puerto de acceso
void setup() {
```

```
Ethernet.begin(mac, ip);           // Inicialización de TCP/IP
server.begin();                    // Configuración servidor
pinMode(3,OUTPUT);                // Valor de salida en la terminal 3
digitalWrite(3,LOW);              // Se asigna un valor bajo en la terminal 3
analogReference(INTERNAL);
Serial.begin(BAUD_RATE);
}
void loop()
{
EthernetClient client=server.available(); // Se evalúa la disponibilidad del servidor
if (client)
{
// Si el cliente existe
while (client.connected())
{
// mientras el cliente este activo
if (client.available())
{
// Se configura el sensor para recibir distancia
// Se emite un pulso de 2us en low y 5us en High en la terminal 7
pinMode(PTRG, OUTPUT);
digitalWrite(PTRG, LOW);
delayMicroseconds(2);
digitalWrite(PTRG, HIGH);
delayMicroseconds(5);
digitalWrite(PTRG, LOW);
pinMode(PECHO, INPUT);
unsigned long duration = pulseIn(PECHO, HIGH); // detección del pulso
k= duration / 29 / 2;                          // conversion a distancia
}
char c = client.read();
if (c == 'A') digitalWrite(3,HIGH);
if (c == 'a') digitalWrite(3,LOW);
}
server.println(k);
client.stop();                               // Fin de la conexión
}
```

Instrumentación Virtual para TCP/IP en LabVIEW

El último dispositivo que se diseñó fue la instrumentación virtual en LabVIEW para obtener los datos del sensor ultrasónico vía TCP/IP, (Vizcaíno y Sebastián, 2007), que están ubicados en la dirección IP del Arduino. En la figura 8 se muestra el código virtual de LabVIEW, en donde se especifica el diagrama de bloques de cada componente son:

- 1 String control del panel String & Path.
- 1 String indicator del panel String & Path.
- 2 Numeric control del panel Numeric.
- 1 Select del panel Comparison
- 2 String constant del panel String.
- 1 Push button del panel Boolean.
- 1 Stop button del panel Boolean.
- 1 Decimal string to Number del panel Number/String conversion.
- 1 Multiply del panel Numeric.
- 1 Waveform Chart del panel Graph.
- 1 Error Out del panel Dialog & User Interface.
- 1 TCP Open connection del panel TCP.
- 1 TCP Read.
- 1 TCP Close connection.
- 1 TCP Write.

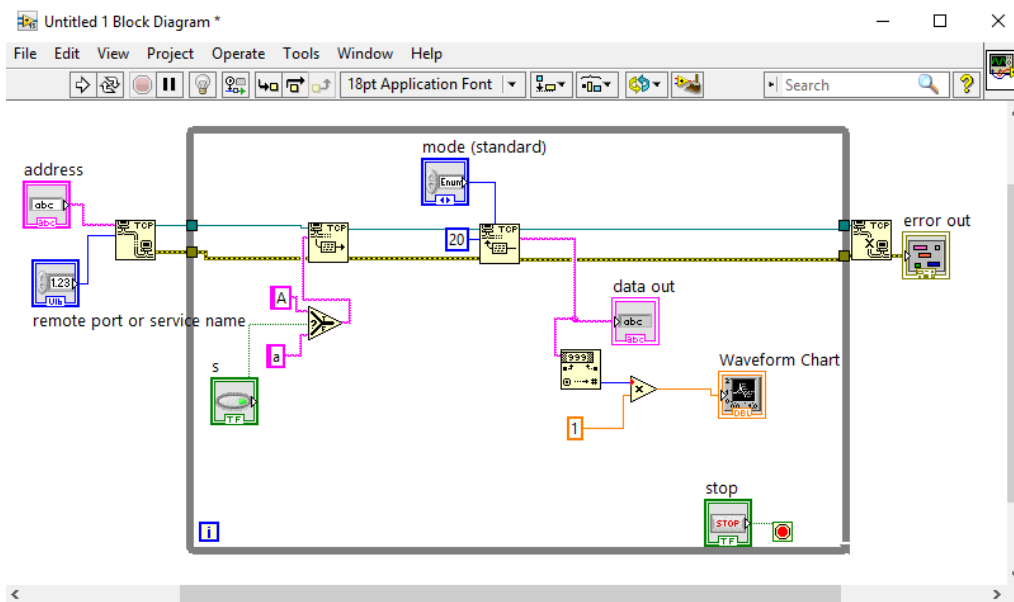


Figura 8 Diagrama de Bloques de la instrumentación virtual para conexión TCP/IP.

En el diagrama a bloques podemos observar los dispositivos conectados, iniciando con el bloque donde se integran los elementos, en una estructura *While loop*, la cual permanecerá activa hasta que suceda un error, detectada por el componente *error out*, o bien mediante el botón *Stop*. El bloque *Address* es aquel donde se registra la dirección IP al cual nos conectaremos de manera dinámica y el bloque *remote port* es el puerto de acceso definido como 8000.

En lo que respecta a la comunicación TCP/IP el bloque externo es *TCP Open Connection*, el cual abre la conexión y evalúa la validez de la dirección IP y el puerto de conexión. Una vez abierta la comunicación se inicializa la escritura y lectura de los datos mediante *TCP Write* y *TCP Read*, los cuales obtienen los datos del sensor mediante la configuración *Standard, CRLF, Imme o Buffer*.

Los datos son reflejados en el *Waveform Chart* mediante la conversión *Decimal String a Number*, en donde podemos observar la distancia obtenida del sensor que está conectado al Arduino en modo de servidor.

El panel frontal de la instrumentación en LabVIEW se presenta en la figura 9, el cual es el resultado de la programación en el diagrama a bloques.

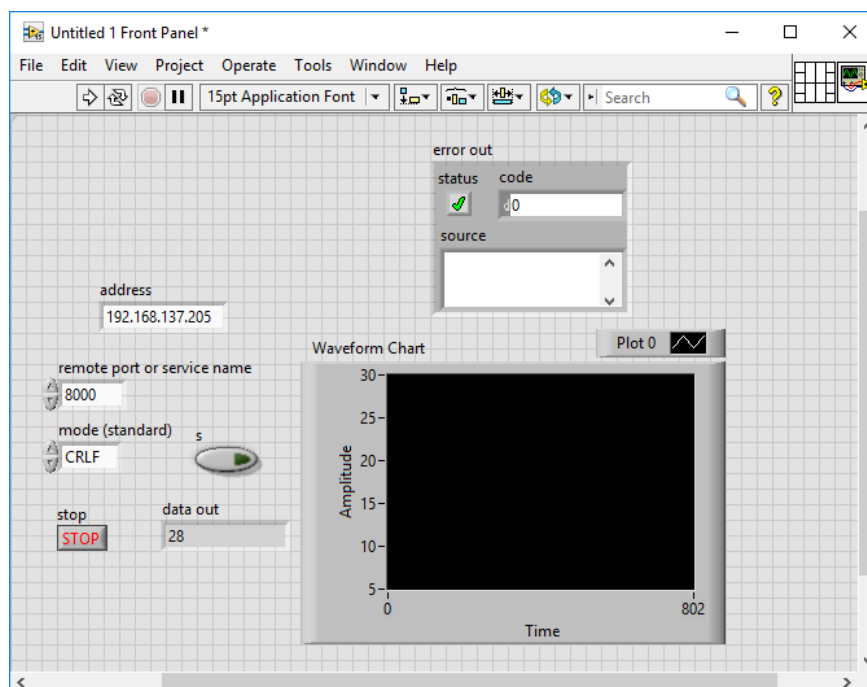


Figura 9 Diagrama Frontal de la instrumentación virtual en LabVIEW.

3. Resultados

Los resultados se muestran en la figura 10, en donde se ve la conexión del sensor ultrasónico con el Arduino y su Shield de Ethernet. La medición es en tiempo real con el acceso TCP/IP habilitado para transmisión de datos.



Figura 10 Conexión del sensor a la plataforma Arduino.

La finalidad de este sistema IoT fue comprobar la interacción de los componentes de adquisición de datos con el Software LabVIEW el cual permitió la manipulación y el procesamiento de los datos. Así mismo conocer el proceso de conectividad entre redes IP y la compartición del acceso a la red aplicando el protocolo TCP/IP, el cual ha sido ampliamente extendido a nivel mundial, y esto se logró gracias a la interconexión entre la plataforma Arduino en modo server y su Shield de Ethernet, comprobamos el direccionamiento dinámico para una red y una subred, figura 11.

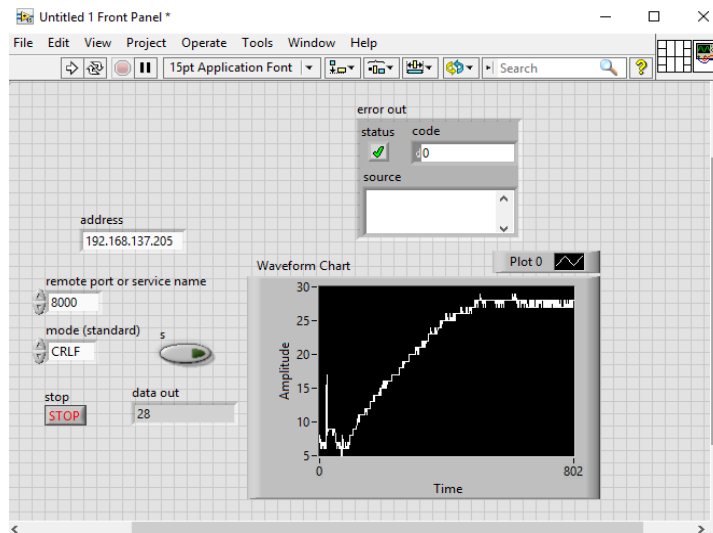


Figura 11 Medición realizada mediante la conexión TCP/IP.

Las mediciones obtenidas con el sensor tuvieron un cierto retraso debido al procesamiento con la plataforma de Arduino y consideraciones en la configuración del programa que permite conectar mediante TCP/IP.

Es posible cambiar el monitoreo de los datos cuando se cambia el modo de recepción estándar, que está definido por el botón *mode*.

4. Discusión

La configuración IoT obtenida de la interacción entre el Arduino y el acceso TCP/IP mediante la interfaz de LabVIEW permite observar un panorama de las diversas complicaciones que se presentan al enlazar más de un sistema vía TCP/IP y la necesidad de realizar sistemas que se integren de una mejor manera. En lo que respecta al sensor ultrasónico se tuvo cierta complejidad al momento de enviar los datos debido a que al programar Arduino se deben considerar primeramente la conexión y escritura de TCP/IP, y ubicar el código fue uno de los desafíos.

La conexión compartida de la computadora hacia el Shield de Ethernet no siempre es sencilla de conseguir, esto debido a falla en los controladores de la red Wifi y que el DHCP no esté habilitado, por lo cual existen procesos para limpiar y reconfigurar los registros.

LabVIEW nos permitió ver las ventajas que se tienen al enlazar sistemas de adquisición de datos y su reconfigurabilidad con respecto a los bloques TCP/IP.

Es posible realizar modificaciones en cuanto a cada instrumento utilizado en esta investigación, una de ellas es modificar la plataforma Arduino por tarjetas con mejor rendimiento, tal es el caso de las tarjetas de Raspberry o las tarjetas MyDAQ de National Instruments.

La presente información es un referente para los interesados en realizar configuraciones básicas de IoT, con la finalidad de que conozcan cada componente y su funcionalidad, poder conectar sensores a los distintos servidores de datos y así acceder a ellos de manera remota mediante el protocolo TCP/IP, para después procesar los datos en software especializado, tal es el caso de LabVIEW.

5. Bibliografía y Referencias

- [1] Crespo, E. (2018). Aprendiendo Arduino. España. Accedido el 30 de Enero, 2018: <https://aprendiendoarduino.wordpress.com/>.
- [2] Isam, E. (2003). TCP/IP Embedded Internet Applications. Great Britain Editorial Newnes.
- [3] LabVIEW (2015), LabVIEW© [Software], trial versión: <http://www.ni.com/download/labview-development-system-2015/5314/en/>.
- [4] Margolis M. (2011) Arduino Cookbook, United States of America. O' Reilly Publishing.
- [5] Ruiz Gutiérrez, J. M. (2013). Arduino + Ethernet Shield. Creative Commons Attribution 3.0: <http://josemanuelruizgutierrez.blogspot.com>.
- [6] Seneviratne, P. (2015) Internet of Things with Arduino Blueprints, Develop interactive Arduino-based Internet projects with Ethernet and Wi-Fi, Birmingham – Mumbai, Packt Publishing.
- [7] Vizcaíno, J., Sebastián J. (2007). LabVIEW: Entorno gráfico de programación. Editorial Alfaomega.