

CARACTERIZACIÓN DE LA FUNCIÓN DE TRANSFERENCIA DEL LLENADO DE UN TANQUE, APLICANDO UN SISTEMA IoT, CONFORMADO POR UN SENSOR ULTRASÓNICO, LA PLATAFORMA ARDUINO Y UNA APLICACIÓN MÓVIL

CHARACTERIZATION OF THE TRANSFER FUNCTION OF THE FILLING OF A TANK, APPLYING A SYSTEM IoT, CONFORMED BY AN ULTRASONIC SENSOR, THE ARDUINO PLATFORM AND A MOBILE APPLICATION

Francisco Adolfo Silva Cruz

Tecnológico Nacional de México/Instituto Tecnológico de Oaxaca
francisco.adolfo.sc@gmail.com

Franco Gabriel Caballero Julián

Tecnológico Nacional de México/Instituto Tecnológico de Oaxaca
francogcaballero@gmail.com

Eric Mario Silva Cruz

Tecnológico Nacional de México/Instituto Tecnológico de Oaxaca
ericmsc@hotmail.com

Carolina Rosas Huerta

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

Isaías Velásquez Cruz

Tecnológico Nacional de México/Instituto Tecnológico de Oaxaca
isaias.velasquez@itoaxaca.edu.mx

Resumen

El Internet de las cosas es la interconexión de los objetos del mundo físico a través de Internet, los cuales requieren de sensores, plataformas electrónicas y plataformas de software, enlazados mediante tecnologías de comunicación. En este artículo se presenta la caracterización de la respuesta de un sistema de nivel de un tanque de agua en una función de transferencia, lo cual se consigue mediante la plataforma Arduino con Bluetooth, enviando los datos a un dispositivo

móvil, para el cual se diseña una aplicación en la plataforma MIT App Inventor 2. Una vez obtenido las mediciones se procede a realizar el modelo matemático haciendo una estimación de la función de transferencia real.

Palabra(s) Clave: App Inventor 2, Arduino, Bluetooth, Caracterización, Función de Transferencia, Internet de la Cosas (IoT), Software Libre.

Abstract

The Internet of Things is the interconnection of the objects of the physical world through the Internet, which require sensors, electronic platforms and software platforms, linked by means of communication technologies. This article presents the characterization of the response of a water tank level system in a transfer function, which is achieved through the Arduino platform with Bluetooth, sending the data to a mobile device, for which an application is designed on the MIT app Inventor 2 platform. Once the measurements are obtained, the mathematical model is made by estimating the actual transfer function.

Keywords: App Inventor, Arduino, Characterization, Free Software, Internet of Things (IOT), Transfer Function.

1. Introducción

Hoy en día existe un consenso a nivel mundial sobre el hecho de que estamos viviendo tiempos de cambios y acelerados a nuestra vida cotidiana que están afectando de manera radical a la industria y a sus distintos sectores. Estos cambios son provocados por la revolución tecnológica o también llamada industria 4.0, con un modelo industrial en el que la innovación es colaborativa y los medios productivos están conectados, figura 1 (Íncipy, 2015).

Según investigaciones realizadas por Cisco en la actualidad solo el 1% de los objetos inteligentes están conectados, y se prevé que en el año 2020 dicho porcentaje aumente hasta llegar a los 50000 millones (Evans, 2011), debido a una mejora en la conectividad a internet y el avance acelerado que ha tenido la tecnología. Uno de los puntos más importantes es la necesidad de crear instrumentación electrónica y software en dispositivos móviles, usando la filosofía

de la industria 4.0 y través del “Internet de las Cosas”, es decir “IoT” es una plataforma para conectar personas, objetos, y entornos para informar y permitir la visibilidad, compromiso, y la innovación ya que nos permite realizar productos inteligentes los cuales se caracterizan por disponer de software embebido, electrónica y conectividad, donde podremos obtener nuevas características, funciones y capacidades, así como también el manejo de datos en los móviles y en la nube electrónica (figura 2) (Schwartz 2016).

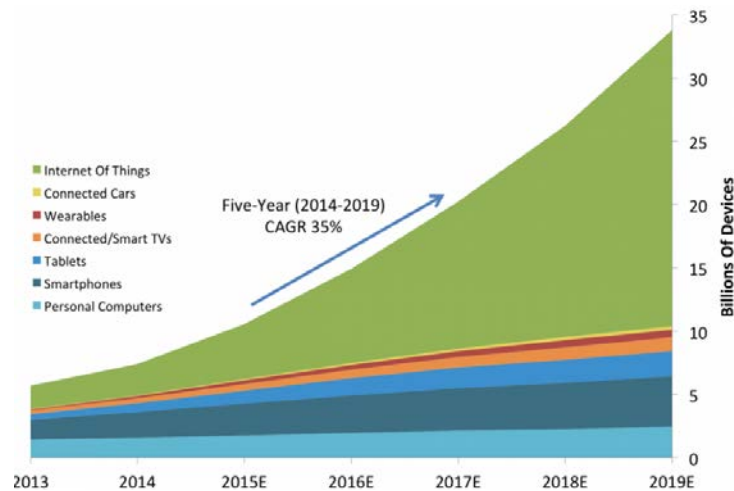


Figura 1 Número de dispositivos en el internet de las cosas.

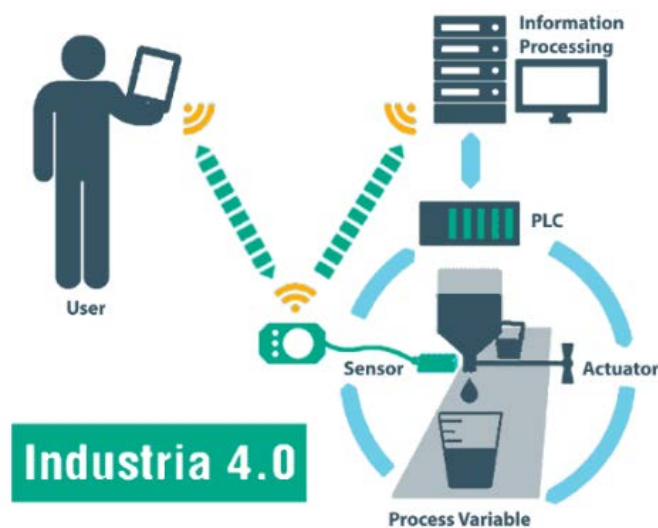


Figura 2 Industria 4.0.

2. Métodos

Control del sistema

Diseñamos un sistema de adquisición de datos en la plataforma de hardware libre Arduino Nano 3.0, figura 3, la cual ha tenido un incremento en su popularidad en años recientes, principalmente porque existe una gran comunidad desarrollando librerías y hardware que facilitan enormemente el desarrollo de un sistema mecatrónico o embebido, (Jecrespom 2018).

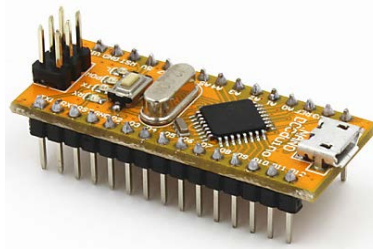


Figura 3 Arduino Nano V3.0 ATmega328P.

Para medir el nivel del tanque se decidió utilizar un 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 4.



Figura 4 Sensor Ultrasónico HC-SR04.

También se usará un led (*light-emitting diode*) para indicar al usuario cuando inicia a recolectar los datos y en qué momento dejará de guardarlos a través de la aplicación móvil, figura 5.



Figura 5 LED (*light-emitting diode*).

Para la comunicación con el celular se usó un Módulo Bluetooth HC-06, figura 6, se ha elegido este, por su bajo costo y es fácil su integración a Arduino.



Figura 6 Modulo Bluetooth HC-06.

Arduino provee su propia IDE de código abierto, disponible para extensión por programadores experimentados. El lenguaje puede ser expandido mediante librerías C++, y la gente que quiera entender los detalles técnicos pueden hacer el salto desde Arduino a la programación en lenguaje AVR C en el cual está basado [Rafael, 2009]. La programación de la placa de control se realizó en la misma IDE, el código desarrollado es el siguiente:

```
1 const unsigned int Trigger = 2;
2 const unsigned int Echo = 3;
3 const unsigned int BAUD_RATE = 9600;
4 const int LED = 13;
5 boolean statusLED = false;
6
7 void setup() {
8   pinMode(LED, OUTPUT);
9   digitalWrite(LED, statusLED);
10  Serial.begin(BAUD_RATE);
11 }
12
13 void loop(){
```

```
14 pinMode(Trigger, OUTPUT);
15 digitalWrite(Trigger, LOW);
16 delayMicroseconds(2);
17
18 digitalWrite(Trigger, HIGH);
19 delayMicroseconds(5);
20 digitalWrite(Trigger, LOW);
21
22 pinMode(Echo, INPUT);
23 const unsigned long duration = pulseIn(Echo, HIGH);
24 if (duration == 0){
25   Serial.println("Warning: We did not get a pulse from sensor.");
26 } else{
27   Serial.print("#");
28   Serial.println(microseconds_to_cm(duration));
29 }
30 delay(100);
31 }
32
33 unsigned long microseconds_to_cm(const unsigned long microseconds){
34   return microseconds / 29 / 2;
35 }
36
37 void serialEvent() {
38   while (Serial.available()) {
39     char comando = (char)Serial.read();
40     switch (comando) {
41       case 'a':
42         cambiarLEDon();
43         break;
44       case 'b':
45         cambiarLEDooff();
46         break;
47       default:
48         Serial.print(comando);
49         Serial.println(": Comando no reconocido");
50         break;
51     }
52   }
53 }
```

```
54
55 void cambiarLEDon() {
56   statusLED = true;
57   digitalWrite(LED, statusLED);
58   Serial.println("ON");
59 }
60
61 void cambiarLEDooff() {
62   statusLED = false;
63   digitalWrite(LED, statusLED);
64   Serial.println("OFF");
65 }
```

Para interconectar nuestro Sensor Ultrasónico, el Bluetooth y el LED a nuestra placa Arduino Nano, se realizó la conexión que se muestra en la figura 7.

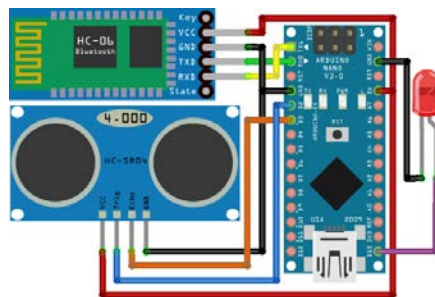


Figura 7 Diagrama de conexión.

Aplicación Móvil

Se desarrolló una aplicación para este proyecto bajo el sistema operativo Android, realizada con la herramienta de programación MIT App Inventor 2, esta plataforma es de código abierto y basada en bloques.

El objetivo principal de la aplicación es recibir datos por medio del módulo Bluetooth con el que cuenta nuestro dispositivo. El segundo objetivo es guardar dichos datos y exportarlos en un archivo.txt para utilizarlos posteriormente en un software matemático (e.g. MATLAB). Adicionalmente se agregó un display de la respuesta en tiempo real con respecto a los datos que se reciben.

Para el desarrollo de esta aplicación se accedió al sitio web <http://ai2.appinventor.mit.edu/>, donde iniciamos sesión con una cuenta de Gmail y

en la pantalla principal creamos un nuevo proyecto (*Start new Projec*), y colocamos nuestro nombre del proyecto, figura 8. Ya que creamos nuestro proyecto, aparece la página principal donde desarrollamos nuestra aplicación, ver figura 9.

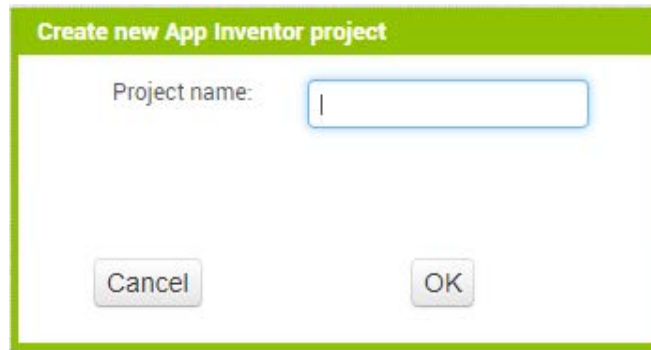


Figura 8 Nombre del proyecto de la nueva App.

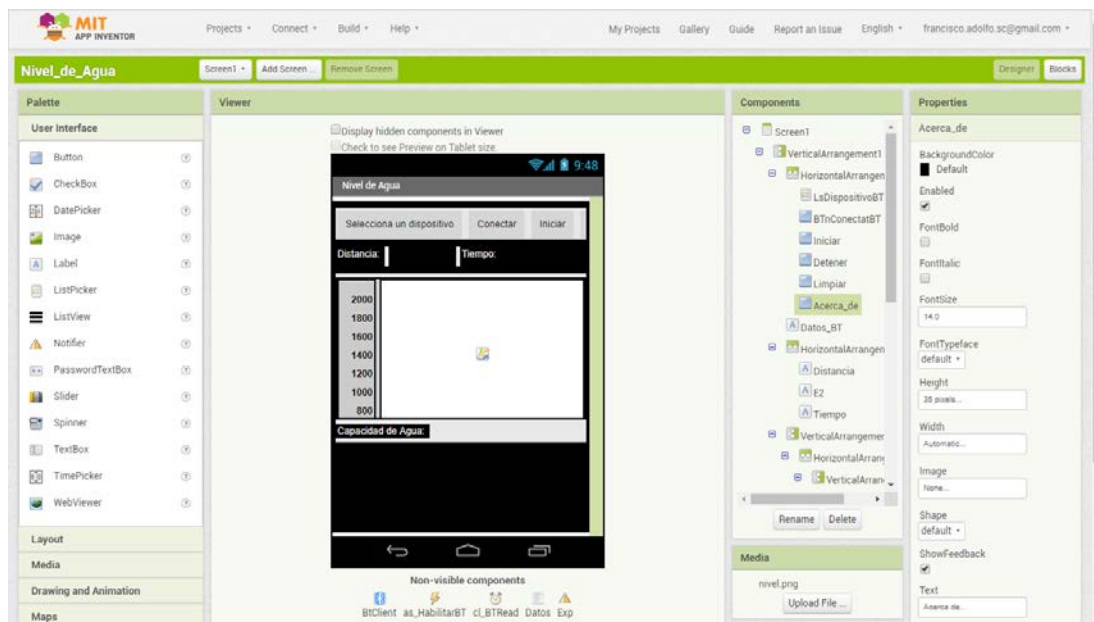


Figura 9 Ventana de diseño de Apps.

En esta ventana, nos encontramos en la pestaña de Designer en la cual se agregan todos los elementos necesarios para construir el aspecto que el usuario podrá visualizar desde su dispositivo Android, en la siguiente figura se puede observar el resultado final de esta aplicación, figura 10.



Figura 10 Aspecto inicial de la App.

A continuación, se muestran cuatro bloques principales del programa que corresponden para establecer la comunicación, el primer bloque crea una variable llamada MAC, figura 11a, donde guardaremos la MAC de nuestro Modulo Bluetooth. El segundo bloque nos permite seleccionar el dispositivo que deseamos conectar, figura 11b. El tercer bloque nos mostrar que dispositivo hemos seleccionado. El cuarto bloque establece la comunicación mostrando por medio de un botón si este se conectó satisfactoriamente.

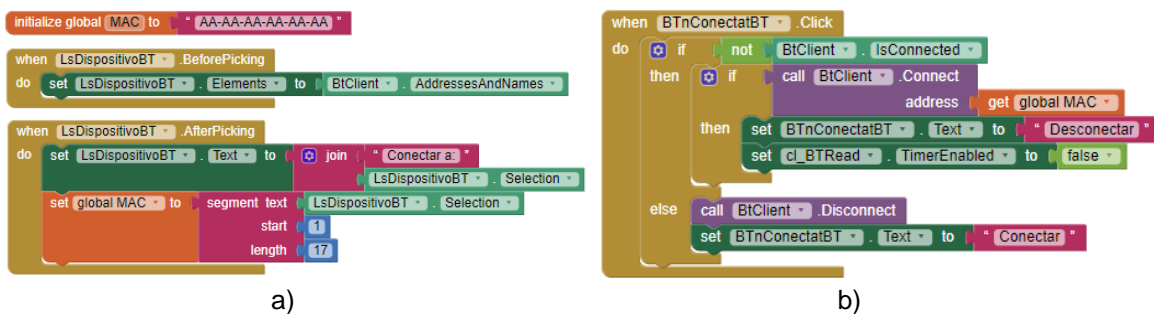


Figura 11 Bloques para establecer la comunicación por Bluetooth.

En la figura 12 se aprecia la interfaz donde se elige con que dispositivo bluetooth nos vamos a conectar, previamente el dispositivo ya debe de estar vinculado con el dispositivo móvil. En figura 13 se muestra el diseño principal, pero con el dispositivo ya conectado. Los tres primeros bloques de la figura 14, definen que variables vamos a utilizar para ir graficando punto por punto, el cuarto bloque define la escala (estando el smartphome de manera vertical) sin embargo hemos colocado el quinto bloque sirven para establecer la misma escala de nuestra grafica

además de configurarla para que esta funcione correctamente al colocar el dispositivo Android de manera horizontal, como último bloque, realiza la grafican punto por punto, (Augusto, 2016).

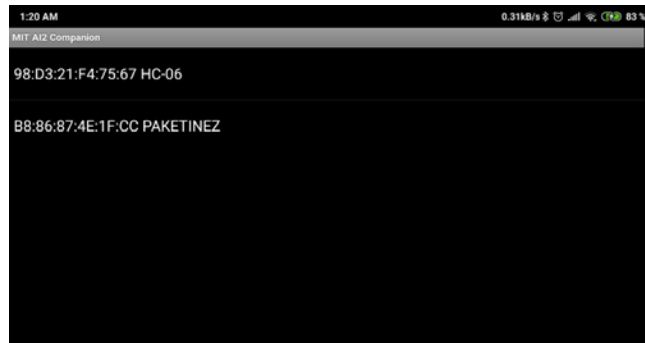


Figura 12 Lista de Dispositivos que se encuentran cerca.



Figura 13 Dispositivo seleccionado y conectado al Modulo Bluetooth.

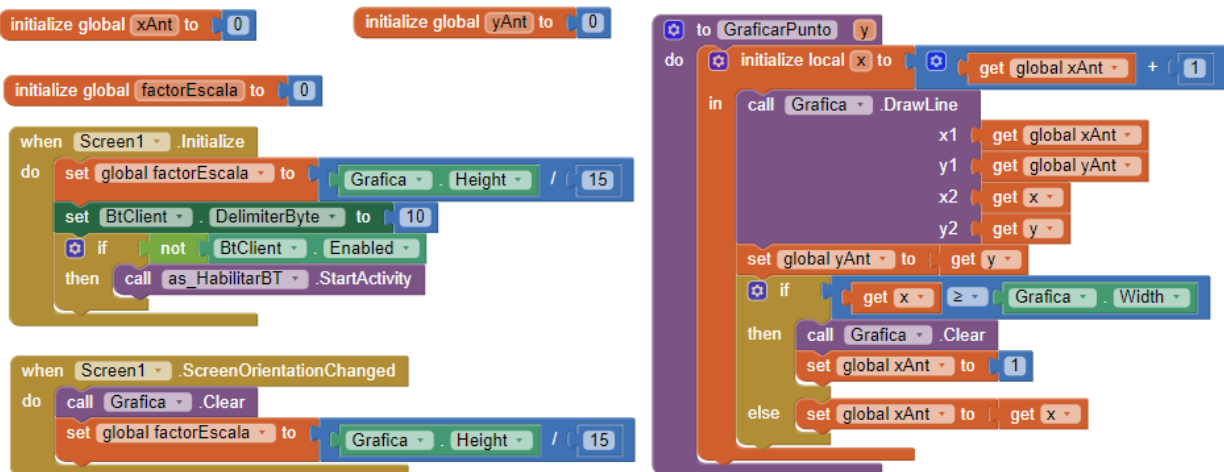


Figura 14 Bloques para la configuración de la gráfica.

En la figura 15, se muestran los bloques para realizar una determinada acción para los botones “Iniciar”, “Detener” y “Limpiar”, el primer bloque realiza la tarea de que al momento que el botón “Iniciar” sea tocado, iniciara un reloj para medir el tiempo, además de borrar un archivo con nombre Exportar.txt donde se almacenan los datos, además de enviar por Bluetooth una letra “a”, la cual avisara al Arduino para que encienda el LED y avise de esta manera que se ha iniciado la captura de datos, además de crear el archivo Exportar.txt antes borrado para almacenar los nuevos datos (María 2016).

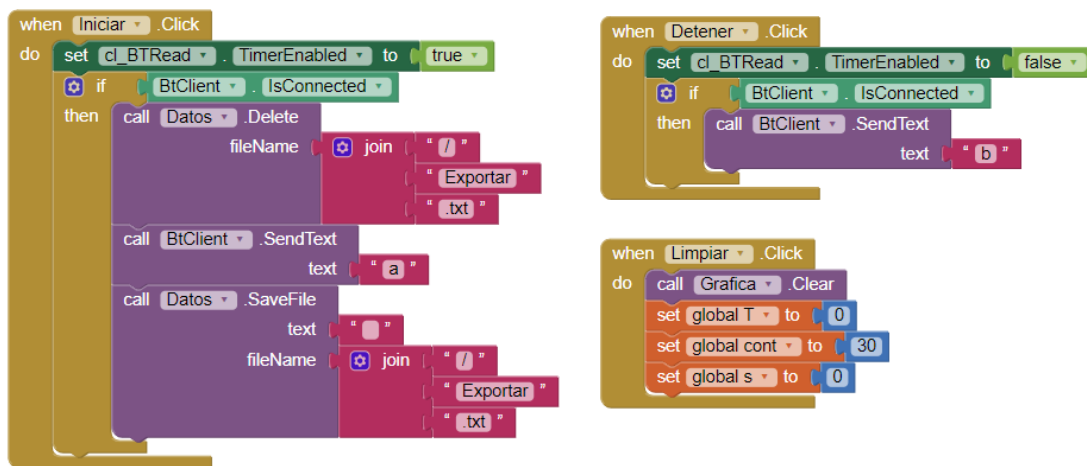


Figura 15 Bloques para los botones “Iniciar”, “Detener” y “Limpiar”.

El bloque dos se activa al tocar el botón “Detener” el cual termina el reloj y además de enviar esta es una letra “b” para que el Arduino en este caso apague el LED e indique que se ha detenido la captura de datos.

El último bloque de la figura 15 por medio del botón “Limpiar” realizara una limpieza de la gráfica, además de restablecer las variables usadas para llevar el tiempo durante la lectura de los datos.

Como último bloque tenemos los mostrados en las figuras 16, 17, 18 y 19, donde el reloj al ser iniciado, este crea una variable donde se guarda el mensaje que fue mandado por el Arduino, además de convertir este dato en tipo de dato entero, de esta manera poder manipularlo, además en esta parte del bloque se muestra que, dependiendo un determinado valor, la gráfica comenzara en un determinado punto.

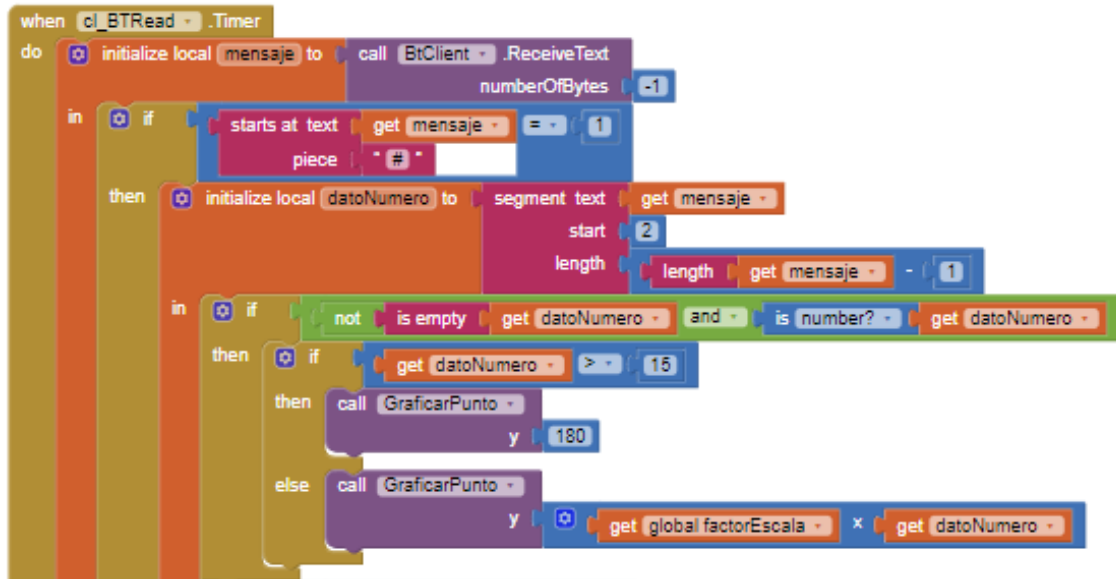


Figura 16 Bloque tipo Timer para realizar la lectura de datos.

En la siguiente parte del bloque Timer (figura 17), se realiza la conversión de la distancia a la capacidad en ml además de mandar a mostrar al usuario que datos estamos recibiendo en cm y en ml respectivamente.

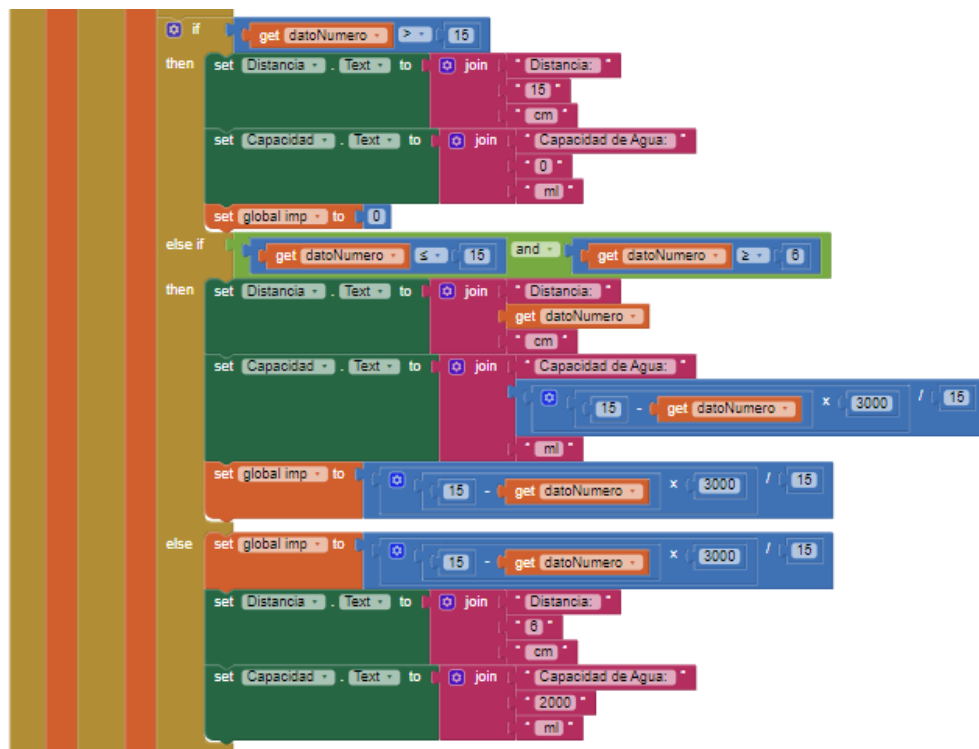


Figura 17 Bloque tipo Timer para convertir los datos e imprimirlos en pantalla.

La parte tres del bloque realiza una comparación con los datos recibidos con datos establecidos, de esta manera mostrar de una manera gráfica la cantidad de agua que lleva el tanque, figura 18.

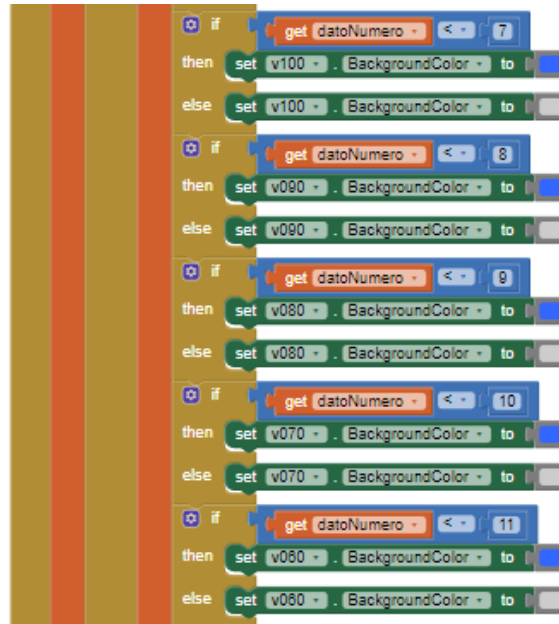


Figura 18 Bloque Timer para comparar datos y mostrar el nivel del agua en el tanque.

En esta última parte del bloque Timer (figura 19), abre el archivo Exportar.txt para guardar los datos cada segundo aproximadamente.

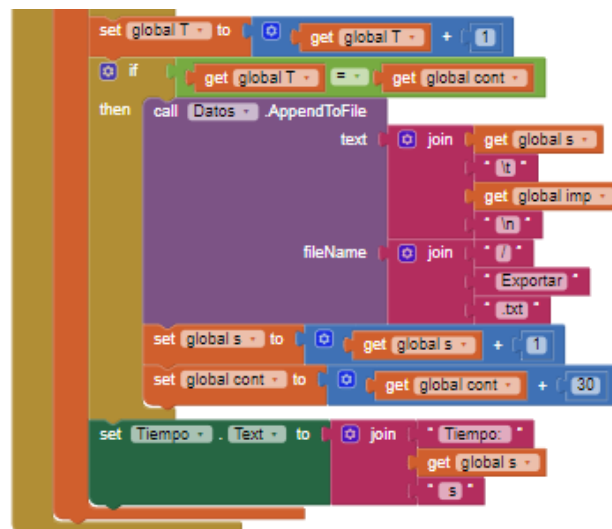


Figura 19 Bloque Timer para guardar los datos en un archivo .txt.

Caracterización de la función de transferencia

Principalmente realizamos la medición del nivel de agua en el tanque con nuestro dispositivo de adquisición de datos como se muestra en la figura 20.



Figura 20 Dispositivo en funcionamiento.

Se ha considerado un método simple para la respuesta frente a un escalón para este sistema, es importante la relación con los datos del sistema, donde no siempre será posible obtener una respuesta completa del escalón, pero sin embargo una parte de la respuesta estará disponible y con estos datos será suficiente implementar dicho método (Niemann, 2014). La caracterización de la función de transferencia se realiza mediante el método de la pendiente, (Hernández, 2010), el cual consiste en obtener una estimación de la respuesta del sistema en una función de transferencia aproximada a la respuesta real, tal como se observa en la figura 21.

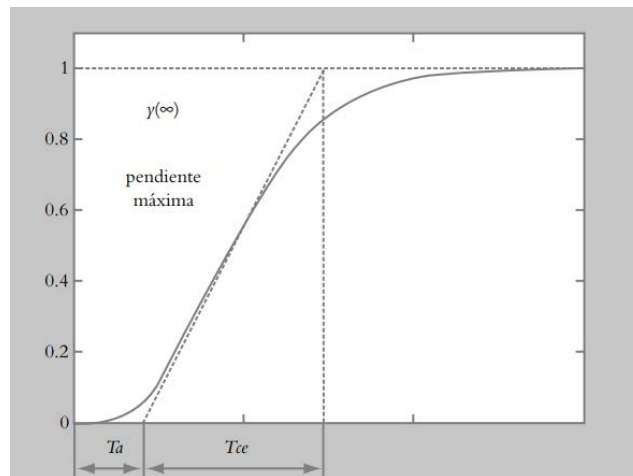


Figura 21 Estimación de la función de transferencia de un sistema.

El método consiste en tomar las mediciones reales de un sistema, en las cuales se observa un comportamiento exponencial creciente, y representa la respuesta del sistema, debido a que las mediciones representan un sistema no lineal, el objetivo de este método es linealizar el sistema, considerando las mediciones reales y aplicando la ecuación 1.

$$T(s) = \left[\frac{\gamma(\infty)}{(\tau s + 1)} \right]^n \quad (1)$$

A la curva de la respuesta real de la medición se le asigna su pendiente máxima, ecuación 2.

$$m = \frac{y_2 - y_1}{x_2 - x_1} \quad (2)$$

En donde el tiempo de atraso T_a es el intervalo del tiempo de elevación, el cual es el inicio de la respuesta al escalón y se proyecta hasta que alcanza el punto en el que la pendiente máxima cruza al eje t .

El tiempo de crecimiento exponencial T_{ce} es el intervalo de tiempo que transcurre desde el cruce de la pendiente máxima, con el eje t , hasta el punto en que dicha pendiente máxima alcanza el valor final $\gamma(\infty)$. Una vez realizada la medición en tiempo real, se obtienen de la gráfica los números T_a y T_{ce} , así como el valor de τ , considerando un retraso de la señal debido al tiempo muerto, aplicamos la ecuación 3.

$$G(s) = \frac{K e^{-T_a s}}{\tau s + 1} \quad (3)$$

3. Resultados

El sistema de nivel de llenado está compuesto por un tanque de agua de dimensiones 19 cm de ancho, 28 cm de largo y 30 cm de altura. El sensor se colocó en la parte superior del tanque para realizar la adquisición del nivel de agua, con un tiempo de muestreo en segundos, tal como se observa en la figura 22, donde se aprecia la medición realizada en tiempo real, los cuales al final de la medición se realiza el envío de los datos vía la tarjeta de bluetooth.

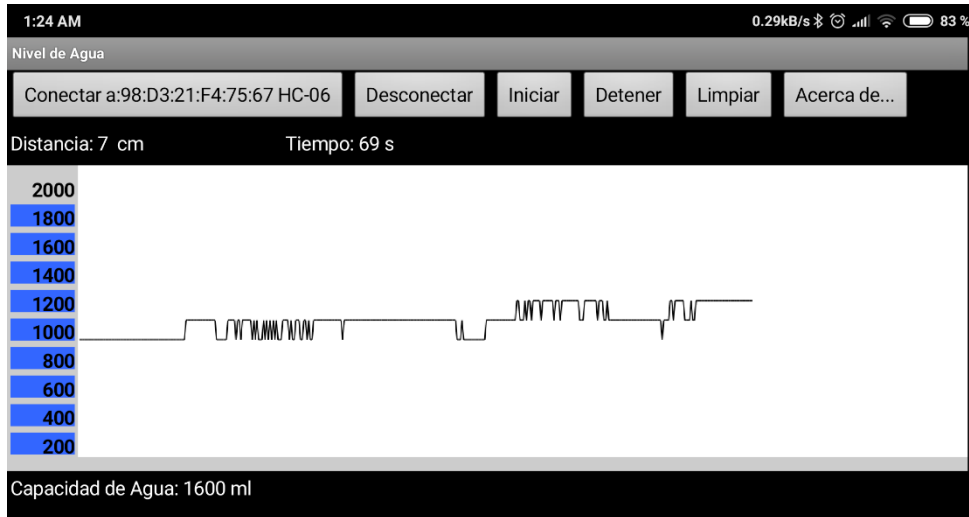


Figura 22 Medición en tiempo real del nivel de agua del Sistema.

Existen diferentes métodos de caracterizar un sistema, entre ellos existe la función “ident” de MATLAB, (Lennart 2018) el cual es un toolbox que permite caracterizar una respuesta de un sistema, en el presente trabajo se implementa un código basado en el método de la pendiente para la caracterización de los datos obtenidos. El matemático es el siguiente. Se define los siguientes valores con respecto a los datos que se obtuvieron por medio de la aplicación:

$$Ta = 10$$

$$x2 = 78$$

$$y1 = 0$$

$$y2 = 2000$$

Donde ta y $y1$ es el punto inicial de la pendiente y $x2$ y $y2$ el punto final de esta (figura 21). Si procedemos a calcular el valor de la pendiente m utilizando la ecuación 2:

$$m = \frac{2000 - 0}{78 - 10} = 29.411$$

Se calcula el valor de τ :

$$\tau = \frac{y2}{m} = \frac{2000}{29.411} = 68$$

Se define la ganancia $K = 2000$ y surtimos los valores en la ecuación 3 y desarrollando, se obtiene ecuación 4.

$$G(s) = \frac{2000}{680 s^2 + 78 s + 1} \quad (4)$$

Este mismo proceso se realiza en la plataforma matemática MATLAB versión estudiantil (Matlab 2012), los datos obtenidos de la medición los procesamos mediante un código que diseñamos para la ecuación 3. De manera que podamos realizar la caracterización de la respuesta del sistema en una función de transferencia, y obteniendo la misma ecuación 11.

El proceso de caracterización se observa en el siguiente código diseñado en MATLAB:

```
1 clc;clear all;close all
2 [x,y]=textread('Exportar.txt');
3 figure(1)
4 plot(x,y,'r');
5 hold on;
6 line([10 78],[0 2000])
7 hold on;
8 x2=78;
9 Ta=10;
10 y2=2000;
11 y1=0;
12 m=(y2-y1)/(x2-Ta);
13 bmax=2000;
14 tau=bmax/m;
15 K=2000;
16 numGs=[K];
17 f1=[tau 1];
18 f2=[Ta 1];
19 denGs=conv(f1,f2);
20 step(numGs,denGs,'m'),
21 title('Medición real vs Caracterización')
22 axis([0 350 0 2200]),
23 xlabel ('tiempo'),
24 ylabel('Altura(Diatancia-Sensro-')
25 grid on;legend('Medición','Pendiente','Caracterización')
26 Gs=tf(numGs,denGs)
```

La caracterización de la función de transferencia del Sistema resultante es la que se observa en la ecuación 4.

En la figura 23 se observa el resultado de las mediciones realizadas con la plataforma Arduino, el método matemático de caracterización utilizado, así como la caracterización de la función de transferencia obtenida.

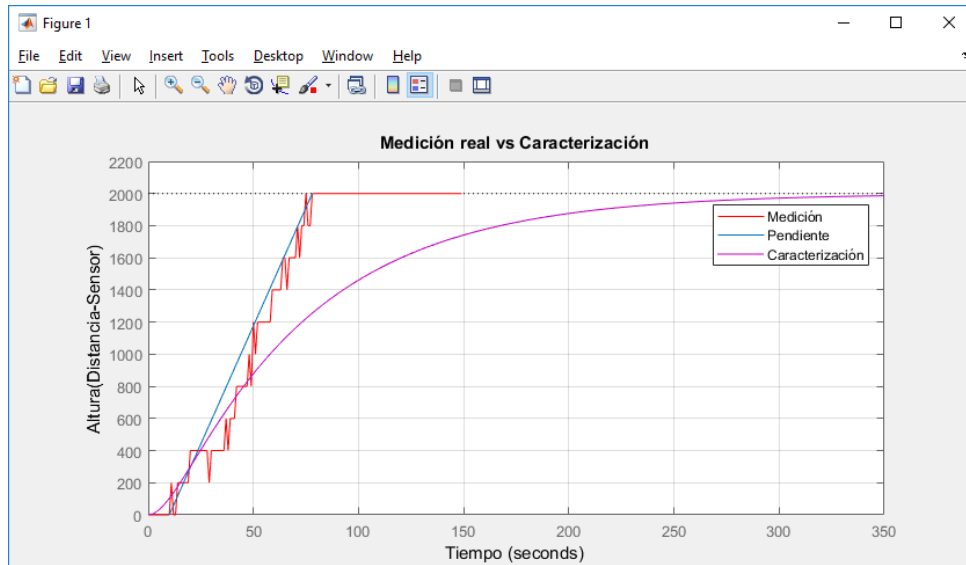


Figura 23 Caracterización de la función de transferencia.

4. Discusión

En el presente artículo se presenta un diseño IoT de bajo costo para ser aplicado en la industria 4.0, haciendo uso de las plataformas Arduino, y MATLAB. Hemos caracterizado la respuesta de un sistema de llenado de un tanque mediante la plataforma Arduino, aproximando la respuesta real del sistema, lo cual nos permite utilizar la información para procesar los datos y modelar el sistema de control en un software para diseñar los controladores PID para mejorar la respuesta al valor deseado, reduciendo los tiempos de procesamiento de señales. Es posible rediseñar el sistema para ser aplicado en la medición de diferentes variables, como puede ser la velocidad del aire, temperatura, humedad, etc. En futuros trabajos se pretende mejorar la adquisición de datos mediante tarjetas Raspberry y MyDAQ o MyRIO, aplicando comunicaciones Ethernet y Wi-Fi, y acceder estos datos a la nube electrónica (iCloud).

5. Bibliografía y Referencias

- [1] Augusto Valdez, (2016), Graficar señal de Arduino en Android por Bluetooth con App Inventor 2: <https://www.youtube.com/watch?v=tztGm6H0xC8>.
- [2] Dave Evans, (2011), How the Next Evolution of the Internet Is Changing Everything. Cisco Internet of Things White Paper.
- [3] Íncipy, (2015), Internet of things (IoT) en la transformación digital de las empresas: <https://www.fundacionseres.org/Lists/Informes/Attachments/987/150923%20internet-of-things.pdf>.
- [4] Hernández Gaviño, R., (2010), Introducción a los sistemas de control. 1st ed. México: Pearson Educación de México, SA de CV, pp.226-227.
- [5] Jecrespom, (2018), Aprendiendo a manejar Arduino en profundidad: <https://aprendiendoarduino.wordpress.com/>.
- [6] Lennart Ljung, (2018), System identification Toolbox User's Guide.
- [7] Maria Salud Mota Aguilar, (2015), Como Exportar una TinyDB a Excel | App Inventor: <https://www.youtube.com/watch?v=tvzGDRFvwXU>.
- [8] Marco Schwartz, (2016), Internet of Things with Arduino Cookbook, Packt Publishing.
- [9] Matlab, Student Version. (2012).
- [10] Niemann, H., and Miklos., R., (2014), A Simple Method for Estimation of Parameters in First order Systems, European Workshop on Advanced Control and Diagnosis, Journal of Physics, Conference Series 570, 01200.
- [11] Rafael Enríquez Herrador, (2009), Guía de Usuario de Arduino: http://www.uco.es/aulasoftwarelibre/wp-content/uploads/2010/05/Arduino_user_manual_es.pdf.