

MONITOREO Y CONTROL PID DE UN SISTEMA RC USANDO IOT MEDIANTE RASPBERRY

MONITORING AND PID CONTROL OF A RC SYSTEM USING IOT THROUGH RASPBERRY

Luis Ángel Velázquez Palomino

Universidad Politécnica del Bicentenario, México
18010395@upbicentenario.edu.mx

Vanessa Pineda Casas

Universidad Politécnica del Bicentenario, México
18010390@upbicentenario.edu.mx

Hugo Antonio Méndez Guzmán

Universidad Politécnica del Bicentenario, México
hmendezg@upbicentenario.edu.mx

Juan José Martínez Nolasco

Tecnológico Nacional de México / IT de Celaya, México
juan.martinez@itcelaya.edu.mx

Recepción: 16/mayo/2020

Aceptación: 3/noviembre/2020

Resumen

Se presenta la implementación de un sistema de monitoreo y control para un circuito RC de primer orden a través de Internet de las cosas (IoT) usando la plataforma Thingspeak. El control y monitoreo se realiza a través de dos canales de la plataforma, donde una GUI en Raspberry sube los parámetros de un controlador PID a uno de los canales y otra Raspberry remotamente descarga los parámetros de control y los aplica al sistema resistivo mediante Arduino, así mismo sube a la plataforma las mediciones en tiempo real del estado actual de la salida, referencia, señal de control y error del sistema, además de enviar vía correo electrónico un gráfico del estado actual del sistema y como mensaje de texto los parámetros de control y variables del proceso. La idea principal de este trabajo es explotar la potencialidad de la plataforma para usarla en sistemas de control pese a las limitaciones que posee un usuario en modo gratuito.

Palabras Clave: *Arduino, IoT, control PID, Thingspeak, Raspberry.*

Abstract

The implementation of a monitoring and control system for a first-order RC circuit through Internet of Things (IoT) using the Thingspeak platform is presented. Control and monitoring is carried out through two channels of the platform, where a GUI made in Raspberry uploads the parameters of a PID controller to one of the channels and another remotely located Raspberry downloads the control parameters and applies them to the resistive system by using Arduino, likewise uploads to the platform the measurements in real-time of the current state of the output, reference, control signal and system error, in addition to sending via email a graph of the current state of the system and as content in a message the control parameters and variables of the process. The main idea of this work is to exploit the potential of the platform to use it in control systems despite the limitations that a user has in free mode.

Keywords: *Arduino, IoT, PID control, Raspberry, Thingspeak.*

1. Introducción

En los últimos años el uso del Internet de las cosas (IoT) ha crecido de forma radical en muchos sectores, tales como la manufactura, servicios públicos, industria electrónica, salud, energía, etc. Las principales características por las cuáles los sectores industriales han adoptado el uso de IoT para sus procesos tienen que ver con la trazabilidad de los procesos, visibilidad de la información, interoperabilidad e interacción a distancia con los procesos [Dachyar, 2019].

Un sistema completo IoT requiere conectividad de múltiples sensores para subir datos a la nube, donde el software instalado en la plataforma tiene como principal función el almacenamiento, análisis y visualización de datos a través de representaciones gráficas en páginas web, tal que dispositivos móviles puedan acceder y un usuario en base al despliegue de la información pueda tomar decisiones [Nayyar, 2018]. Entre las principales plataformas de IoT para subir y analizar información a la nube se encuentran Kaa, SiteWhere, DeviceHive, Zetta, DSA, Thingsboard.io, Thinger.io, MainFlux y Thingspeak, en los cuales los principales protocolos a los que dan soporte se encuentran MQTT, TCP y HTTP

permitiendo la conectividad con dispositivos como Arduino, ESP8266, Raspberry, Intel Edison y BeagleBone entre otros.

Este trabajo está orientado al uso de la plataforma Thingspeak bajo una licencia libre y su conectividad con Raspberry para llevar el monitoreo y control de un sistema RC de primer orden de forma remota.

Plataforma de IoT Thingspeak

Thingspeak es una plataforma que ofrece servicios para el almacenamiento de datos en la nube, su visualización y su análisis a través de MATLAB.

El número de mensajes permitido es de 3 millones por año para la licencia en modo FREE o hasta 33 millones por año por unidad en licencia de paga [Thingspeak, 2020], donde una unidad se puede considerar como la cantidad de datos generados de un solo canal con una tasa de actualización de un segundo.

Dependiendo del tipo de cuenta se tiene un límite de canales, permiso para compartir los gráficos de los canales privados y un costo anual (tabla 1). Cada canal puede monitorear 8 variables distintas, graficando los datos y etiquetándolos con fecha, hora y zona horaria en el momento que fueron captados por la plataforma.

Tabla 1 Costo de almacenamiento de datos con tasa de actualización de 1 segundo.

	FREE	Home	Academic	Student	Standard
Máximo número de canales permitido	4	10	250	10	250
Compartición de Canales Privados	3	-	-	-	-
Costo anual por canal (USD)	-	\$95	\$250	\$79	\$650

Thingspeak permite la elección del tipo de licencia y la tasa de actualización de datos en el momento de crear la cuenta y dado que la plataforma cobra por la cantidad de datos almacenados, algo a resaltar es que el costo de tener un solo canal con una tasa de actualización de 1 segundo es el mismo que si se quisiera dos canales con tasa de actualización de 2 segundos. Sin embargo, para un usuario con licencia gratuita solo puede subir o descargar datos con una tasa de actualización de 15 segundos y con un límite de 4 canales.

Consideraciones para muestreo y control mediante Thingspeak

Considerando el uso de la plataforma para un usuario con licencia FREE se debe tener en cuenta la limitación de los sistemas de control que se podrán implementar, ya que al subir los datos se está muestreando el proceso y esto implica una pérdida de información. Para un sistema de primer orden con función de transferencia, tenemos ecuación 1.

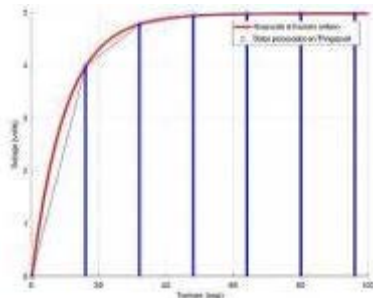
$$\frac{C(s)}{R(s)} = \frac{1}{Ts + 1} \quad (1)$$

Donde T representa el momento en el cuál el sistema llega al 63.2% de su valor final y su estabilización se da al transcurrir 5T con un error del 1% [Ogata, 2003]. Para darse una idea clara de cómo se visualizaría un proceso controlado externamente subiendo datos hacia Thingspeak con una tasa de actualización de 15 segundos, se puede simular a detalle el comportamiento del sistema con un controlador On-Off y la respuesta al escalón unitario, haciendo uso de su equivalente en el dominio digital (ecuación 2) proveniente de su discretización con la instrucción c2d (ecuación 3) de Matlab u Octave y con un tiempo de muestreo de 0.1 segundos.

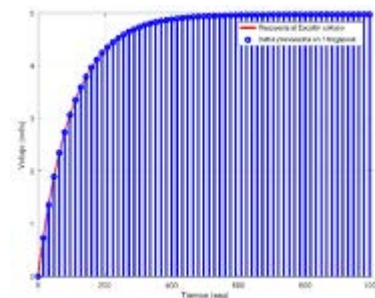
$$\frac{C(Z)}{R(Z)} = \frac{a}{Z - b} \quad (2)$$

$$c[k] = r[k]a + c[k - 1]b \quad (3)$$

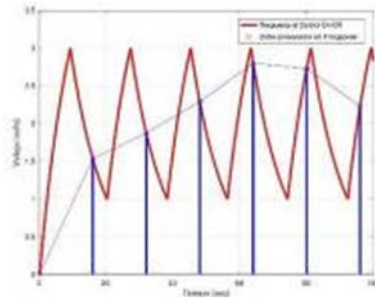
La figura 1 muestra como ante la respuesta al escalón unitario del sistema, Thingspeak alcanzaría a representar adecuadamente en la plataforma la respuesta para ambos casos, un sistema de primer orden con T igual a 10 y a T igual 100 (figuras 1a y 1b), sin embargo, para un control on-off con T igual a 10 no alcanzaría a visualizarse adecuadamente, mientras con T igual a 100 alcanzaría representar de 5 a 6 muestras de subida o de bajada, es decir que para llevar un sistema de control con un tiempo de muestreo igual a 15 segundos (tasa de actualización para un usuario FREE) usando como muestreador a Thingspeak, es necesario que T del sistema de primer orden sea por lo menos 5 veces mayor a la tasa de actualización de Thingspeak.



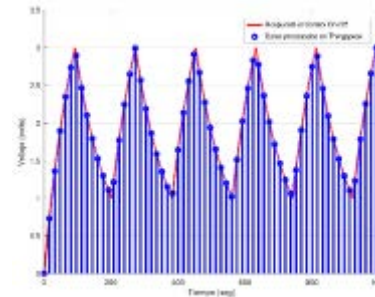
a) Respuesta escalón unitario con $T=10$.



b) Respuesta al escalón unitario con $T=100$.



c) Respuesta al control On-Off con $T=10$.



d) Respuesta al control On-Off con $T=100$.

Figura 1 Simulación de la visualización de un sistema de primer orden.

2. Métodos

Tomando las consideraciones para trabajar con la plataforma Thingspeak en modo gratuito y para la generación de un controlador PID con comunicación IoT se adoptó un modelo conformado por dos partes (figura 2), el Panel del Supervisor donde un usuario tiene acceso a visualizar el estado del proceso en la plataforma Thingspeak y a través de una GUI en Raspberry puede modificar y enviar los parámetros de *referencia*, K_p , K_i y K_d a la nube a un canal 1 de Thingspeak, y por otra parte el Sistema de Monitoreo y Control que descarga los parámetros de control del canal 1, calcula la señal de control, la aplica y sube los datos recolectados de nuevo a Thingspeak a un canal 2, además de enviar por correo electrónico una imagen del estado del proceso, así como los parámetros actuales del proceso en forma textual.

Ecuación del PID discreto

Para poder aplicar un controlador PID con comunicación a la plataforma Thingspeak es necesario conformar su ecuación en el dominio digital considerando como t_s la tasa de actualización de Thingspeak siempre y cuando el T del sistema

de primer orden sea por lo menos 5 veces mayor a esta. Para el sistema de monitoreo y control se adopta el equivalente de la transformada Bilineal para los equivalentes discretos de los componentes PID [Srivastava, M, et al 2009], donde controlador PID discreto se define con ecuación 4.

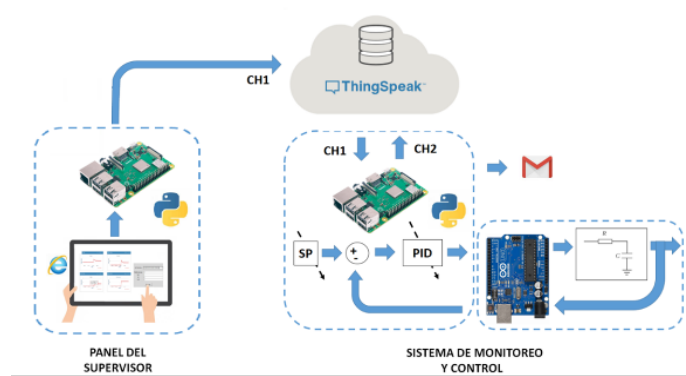


Figura 2 Sistema de monitoreo y control mediante Thingspeak.

$$D(z) = K_p + K_i \frac{t_s}{2} \left[\frac{z+1}{z-1} \right] + K_d \left[\frac{z-1}{t_s z} \right] \quad (4)$$

Cuya ecuación en forma canónica puede adoptar la forma de la ecuación 5.

$$D(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}} \quad (5)$$

Y donde cada coeficiente queda definido por ecuaciones 6 a 10.

$$a_0 = K_p + \frac{K_i t_s}{2} + \frac{K_d}{t_s} \quad (6)$$

$$a_1 = -K_p + \frac{K_i t_s}{2} + \frac{-2K_d}{t_s} \quad (7)$$

$$a_2 = \frac{K_d}{t_s} \quad (8)$$

$$b_1 = -1 \quad (9)$$

$$b_2 = 0 \quad (10)$$

A partir de la función de transferencia en dominios de la transformada Z, se obtiene ecuación 11, controlador en el dominio digital.

$$y[n] = x[n]a_0 + x[n - 1]a_1 + x[n - 2]a_2 - y[n - 1]b_1 - y[n - 2]b_2 \quad (11)$$

Con $y[n]$, $x[n]$ y $y[n - 1]$, $y[n - 2]$, $x[n - 1]$ y $x[n - 2]$ como la salida del controlador, el error del sistema actuales y sus muestras pasadas.

Hardware del Panel del Supervisor

El Panel del Supervisor, como lo muestra la figura 3 es un GUI realizado a través de Tkinter en una tarjeta de desarrollo Raspberry, el cual consta de entradas de texto y un botón para envío de los parámetros de control a Thingspeak con su respectiva API_key para acceso al canal 1. Si un usuario tiene acceso a visualizar el comportamiento del sistema a través de Thingspeak, puede usar esta GUI para cambiar la referencia del sistema o bien adecuar los parámetros del PID para mejorar la respuesta del sistema.

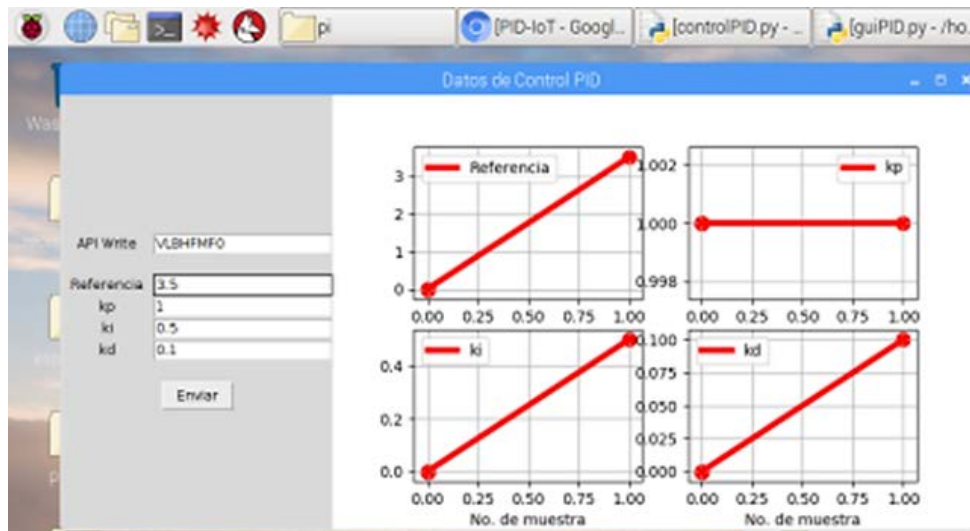


Figura 3 GUI para envío de parámetros PID a la nube.

Hardware del Sistema de monitoreo y control

El sistema de monitoreo y control consta de una tarjeta de desarrollo Raspberry y un Arduino UNO (figura 4a). Raspberry implementa la comunicación con la plataforma Thingspeak para descargar los parámetros de control *referencia*, K_p , K_i , y K_d de la nube, y se comunica con Arduino para recibir la adquisición analógica medida sobre el capacitor para realizar con ellos los cálculos del error del sistema y

la señal de control (ecuación 11) que se envía a Arduino, ambos usando comunicación serial en Python.

Arduino por su parte se encuentra conectado a un puerto USB de Raspberry (figura 4b) funcionando sólo como interfaz para lectura de canales analógicos e inyección de la señal de control a través de PWM al sistema resistivo a través de la paquetería ArduinoIO instalada en él y proporcionada por MATLAB.

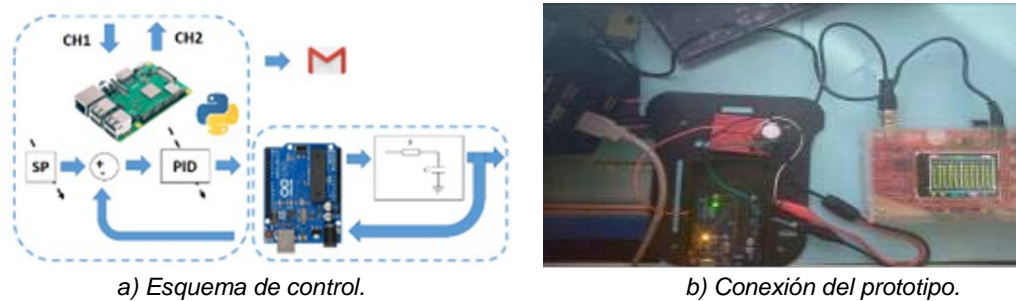


Figura 4 Sistema de Monitoreo y Control.

Calculada la salida del controlador es enviada al sistema RC aplicando una señal PWM proporcional en ancho de pulso a la salida del controlador PID. En la figura 5 se muestra parte del código implementado. Como se puede apreciar, el programa realizado en Python usa las instrucciones para controlar Arduino provenientes de las paqueterías de ArduinoIO.

```
v[k]=(5.0/1023)*a.analogRead(lecturapin)
x[k]=s[k]-v[k]
y[k]=a0*x[k]+a1*x[k-1]+a2*x[k-2]-b1*y[k-1]
if y[k]>5:
    y[k]=5
if y[k]<0:
    y[k]=0
pwm=int((255/5)*y[k])
a.analogWrite(controlpin,pwm)
print("vp=",v[k],"error=",x[k],"control=",y[k],"pwm=",pwm)
```

Figura 5 Código Python para control PID.

Descripción del Software

En la figura 6 se muestra un diagrama de flujo que representa las funcionalidades de software ejecutadas por cada Raspberry. Como se aprecia el Panel del

Supervisor (figura 6a) siempre se encuentra en espera de la presión de un botón, cuando es presionado lee los datos que el usuario haya colocado en las entradas de texto y las envía al canal 1 de Thingspeak como parámetros de control.

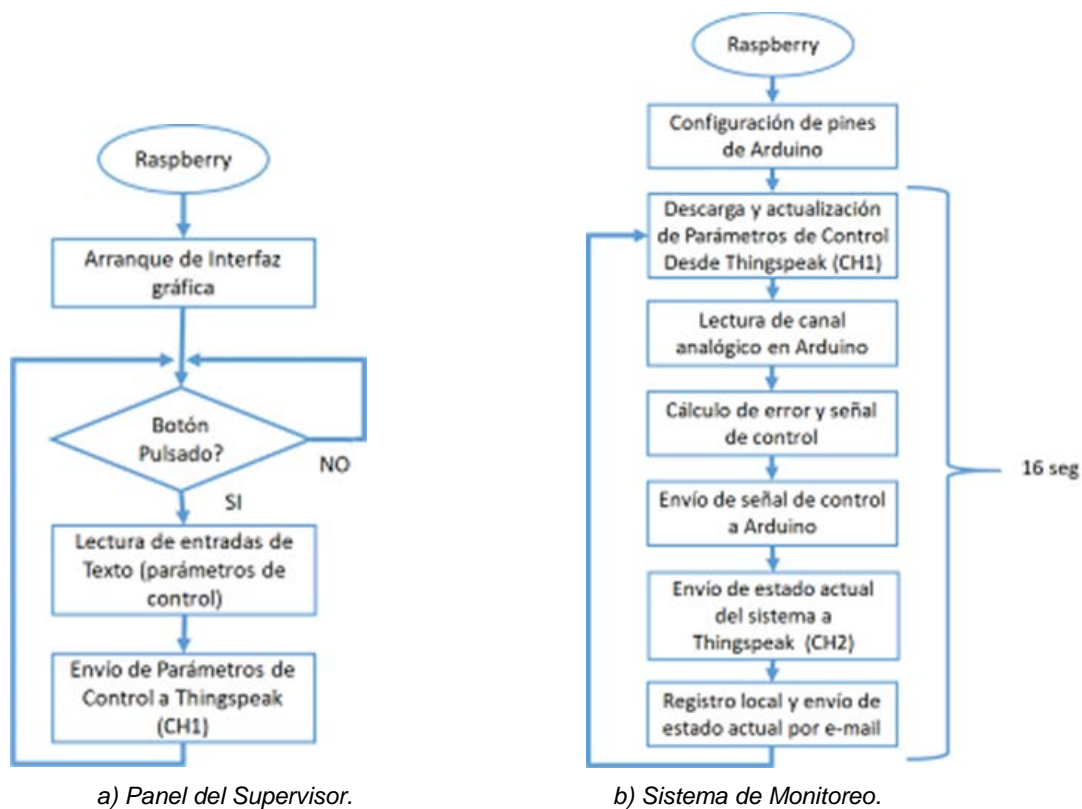


Figura 6 Software de cada dispositivo.

Por otra parte, en el sistema de Monitoreo y Control (figura 6b), una vez que ha configurado la tarjeta Arduino entra en un bucle infinito, donde en cada iteración descarga los parámetros de control del canal 1 de Thingspeak, realiza la lectura analógica del estado del capacitor a través de Arduino y calcula el error del sistema y la señal de control. Posteriormente envía el resultado del cálculo a Arduino para que este inyecte la señal de control al sistema RC por PWM y envía la información a Thingspeak para que pueda ser registrada y monitoreada en la plataforma. Finalmente hace un registro local del estado del proceso y envía por correo electrónico una imagen del proceso y los valores actuales del sistema en forma textual, todo este procedimiento en el lapso de 16 segundos.

3. Resultados

El sistema de control por IoT fue probado considerando como tiempo de muestreo una tasa de actualización de datos en Thingspeak de 16 segundos y para el sistema RC un resistor de 100 k Ω y un capacitor de 1000 μ F para generar un sistema de primer orden con un T de 100. La primera prueba fue verificar el funcionamiento del controlador en forma de proporcional puro, enviando como parametros de control 3.0, 1.0, 0, 0 como referencia, control proporcional, control integral y control derivativo respectivamente (figura 7).

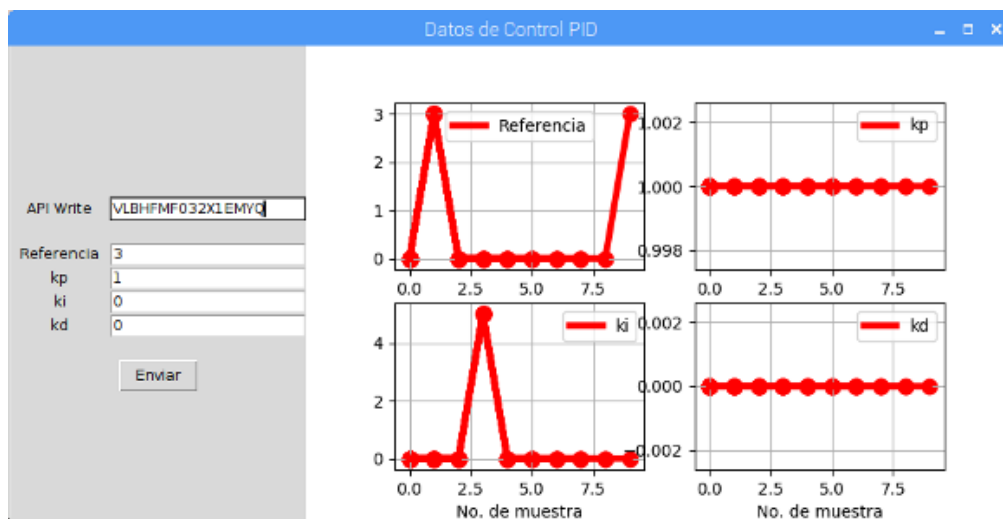


Figura 7 Envío de parámetros de control como proporcional.

Las figura 8 y 9 muestran como es efectuado el registro en la plataforma Thingspeak, mientras la figura 10 muestra el registro local del comportamiento del proceso por el sistema de monitoreo y control, el cuál envía esta imagen via correo electrónico, así como el estado actual de las variables del proceso.

La figura 9 muestra los efectos del controlador proporcional con las órdenes de mando del canal 1 de Thingspeak (figura 8).

Tras la primera prueba se modifíco el controlador via remota como controlador proporcional integral, enviando como parametros de control 3.5, 1.0, 0.5, 0 como referencia, control proporcional, control integral y control derivativo respectivamente (figura 11). Las figuras 12 y 13 muestran como es efectuado el registro en la

plataforma, mientras figura 14 muestra el registro local del comportamiento del proceso por el sistema de monitoreo y control. Finalmente se procedió a sintonizar el sistema via remota, enviando como parametros de control 3.5, 1.0, 0.05, 0 como referencia, control proporcional, control integral y control derivativo respectivamente.

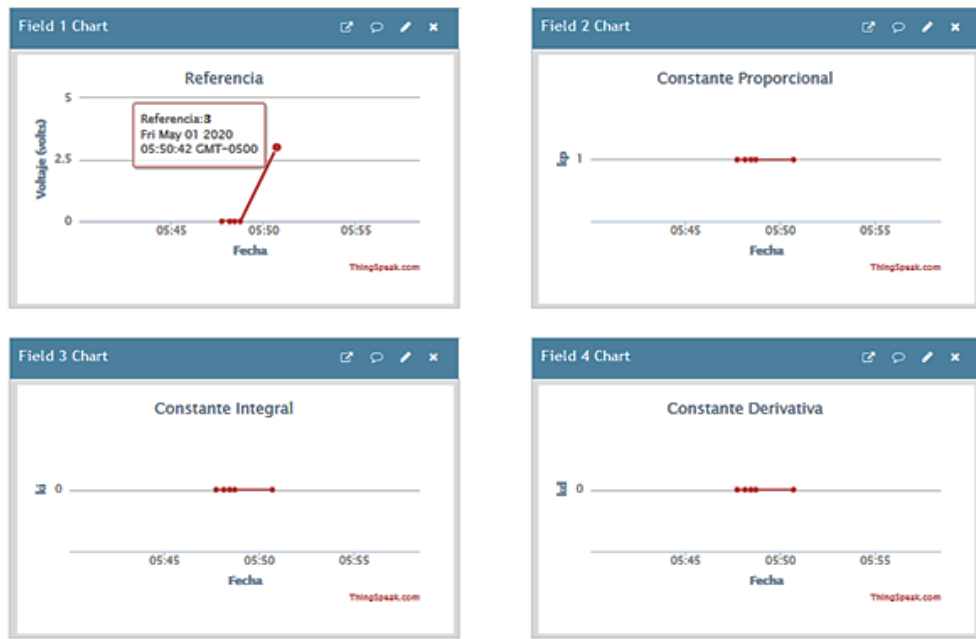


Figura 8 Registro de parámetros de control como proporcional en canal 1 de Thingspeak.



Figura 9 Registro de variables del proceso en canal 2 de Thingspeak.

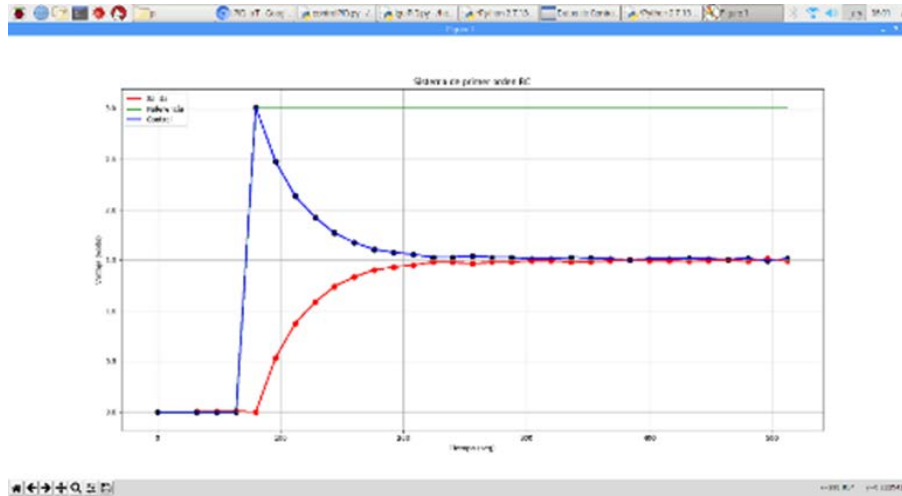


Figura 10 Registro local de variables.

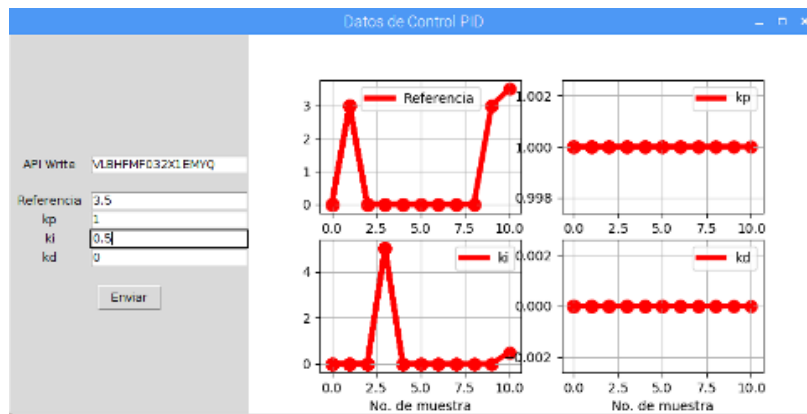


Figura 11 Envío de parámetros como controlador PI.

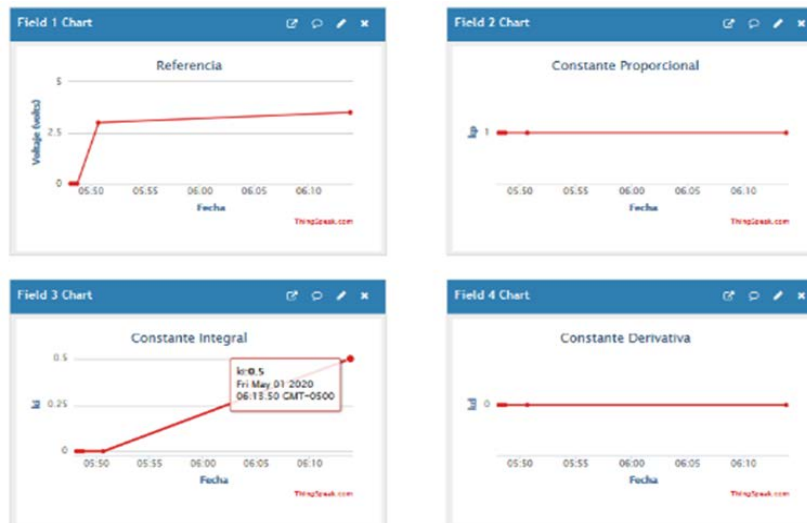


Figura 12 Registro de parámetros como controlador PI en canal 1 de Thingspeak.



Figura 13 Registro de variables del proceso en canal 2 de Thingspeak.

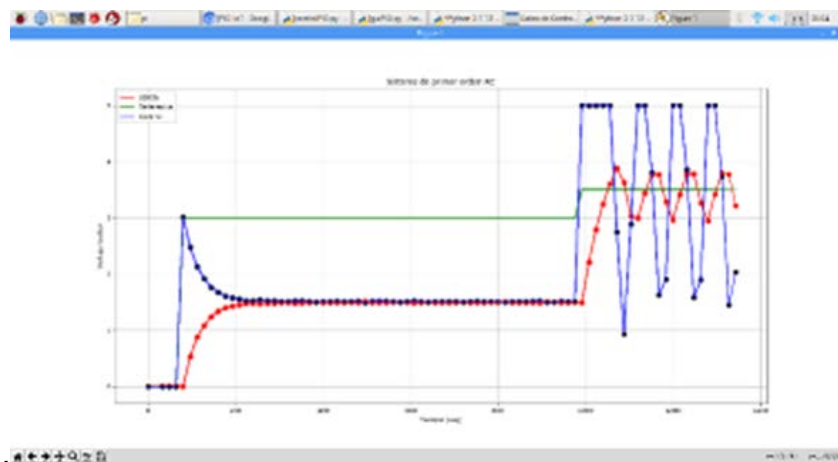


Figura 14 Registro local de variables en sistema de monitoreo y control.

La figura 15 muestra como es efectuado el registro en la plataforma, mientras la figura 16 muestra el registro local del comportamiento del proceso por el sistema de monitoreo y control, observandose un sobreimpulso de 0.35 y un tiempo de asentamiento de 334 segundos para un criterio de error del 1%.

En la figura 17a se muestra el registro de correos electrónicos enviados por el sistema de monitoreo de control, mientras en la figura 17b se aprecia uno de los mensajes enviados, cuyo contenido es el estado actual de las variables del proceso y una imagen del registro local realizado por el sistema de monitoreo y control.



Figura 15 Registro de variables del proceso sintonizado en canal 2 de Thingspeak.

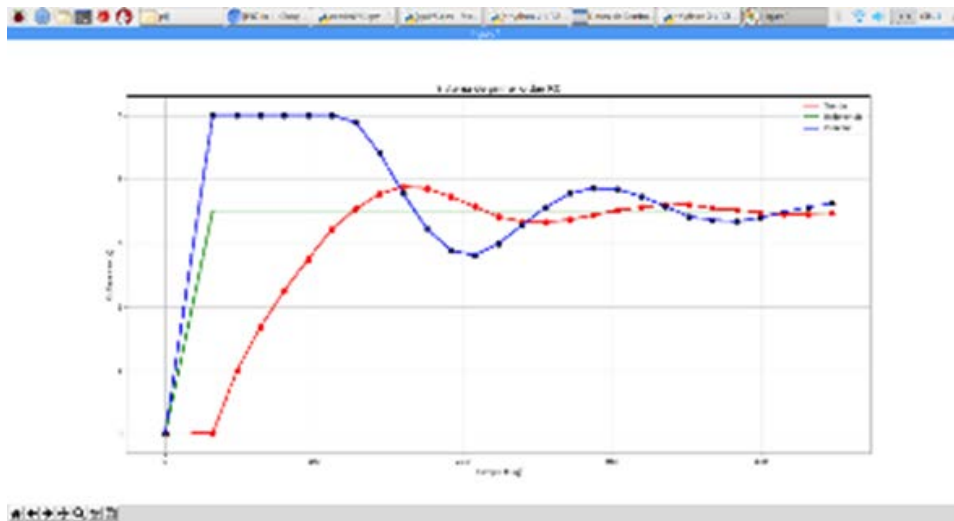
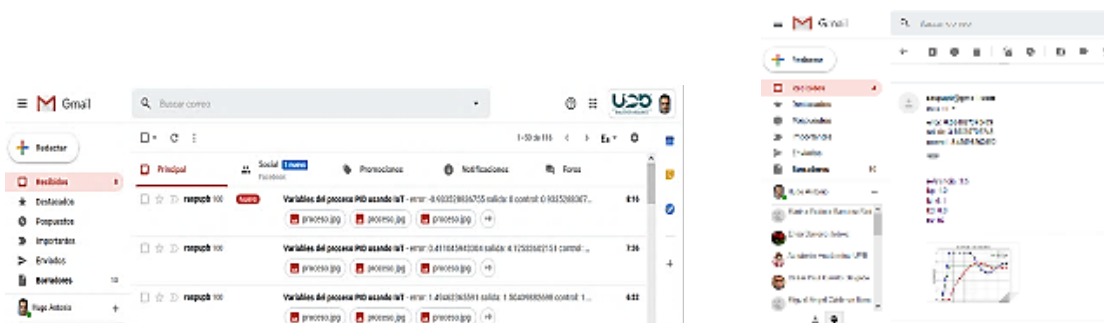


Figura 16 Registro local de variables del proceso sintonizado.



a) Registro de correos electrónicos.

b) Contenido de uno de los mensajes enviados.

Figura 17 Envío de alertas periódicas vía correo electrónico.

Una característica a resaltar del uso de Thingspeak es la existencia de aplicaciones gratuitas para su monitoreo a través de dispositivos móviles como Thingview, la cuál permite visualizar en tiempo real la información subida a la nube (figura 18).

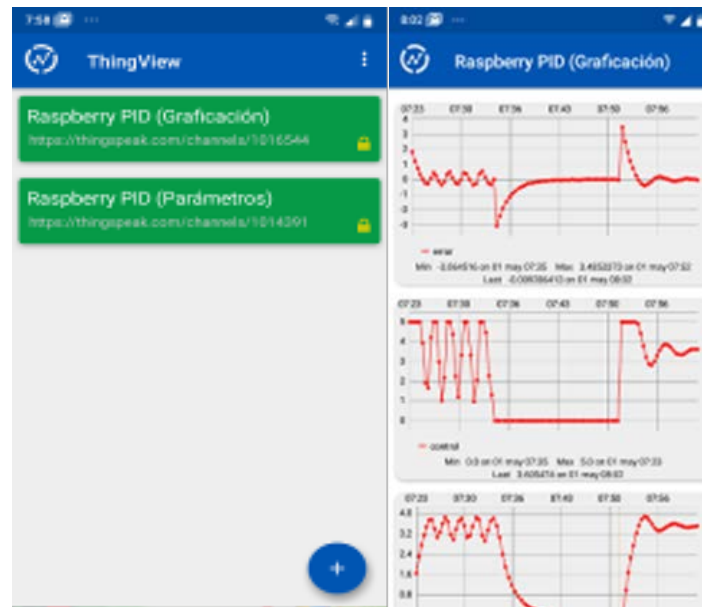


Figura 18 Monitoreo de canales de Thingspeak a través de la aplicación Thingview.

4. Conclusiones

A través de este trabajo se mostró la implementación de un sistema de control PID para controlar un circuito RC de primer orden, considerando las restricciones en el muestreo de la señal al trabajar en modo gratuito con la plataforma Thingspeak y con una tasa de actualización de 16 segundos.

Así mismo se demuestra que si se tiene una aplicación con capacidad de enviar datos a Thingspeak, en un lugar remoto pueden ser descargados por otro dispositivo para aplicarlos en el control supervisado de un proceso.

Entre las cualidades a resaltar de la plataforma Thingspeak es la existencia de aplicaciones gratuitas en Android para monitoreo y envío de datos a través de dispositivos móviles, aplicaciones como ThingView, IoT Thingspeak Monitor y ThingSpeak (IoT) entre otros.

Por otra parte el uso de tecnologías como Raspberry permite incorporar a sistemas remotos registros y análisis de información de forma local, sistemas de alerta en la

modificación de procesos e inclusive la intercomunicación con más sistemas electrónicos como lo son tarjetas de adquisición como lo es Arduino, dando una potencialidad mucho mayor al uso la plataforma Thingspeak pese a las restricciones para un usuario en modo gratuito.

5. Bibliografía y Referencias

- [1] Dachyar, M.; Zagloel, T.; Saragih, L. (2019). Knowledge growth and development: internet of things (IoT) research, 2016-2018. Department of Industrial Engineering, Universitas Indonesia. Heliyon.
- [2] Nayar, A. (2018). The Top Open Source IoT Platforms for Developers: <https://opensourceforu.com/2018/10/the-top-open-source-iot-platforms-for-developers/>.
- [3] Thingspeak. License Options (abril de 2020): <https://thingspeak.com/prices/>.
- [4] Ogata, K. (2003). Ingeniería de Control Moderna. Prentice Hall. México. (pags 221-222).
- [5] Srivastava, M.; Sristava, M. C.; Bhatnagar, S. (2009). Control Systems. Tata McGraw-Hill. New Delhi. (pag 225).