

Examen parcial # 2:

Controlador PID con Datalogger

Universidad de San Carlos de Guatemala
Facultad de Ingeniería
EIME - Lenguajes de Programación AIE (0991) "N"

Profesor: Iván René Morales
2º Semestre 2014

¿Qué debe hacer?

En esta ocasión deberá utilizar dos plataformas de desarrollo separadas (una computadora y un microcontrolador) y realizar la comunicación entre ambas (bluetooth o UART). El microcontrolador estará encargado de mantener una temperatura constante en el ambiente a través de un controlador PID y un par de actuadores, mientras que la computadora (o Raspberry PI) programada en Python 2.7 será quien indique al microcontrolador los coeficientes de calibración del PID y el *set point*. Asimismo, esta última almacenará la información recabada por el sensor, y la desplegará en un servidor web local, incluyendo la gráfica que muestre los valores en función del tiempo. Es posible hacer uso de hilos (*threading*), en caso de ser necesario.

- **PID:** Puede basarse en el controlador PID del ejemplo (**tiva09**). Deberá implementar un sistema de control de lazo cerrado que actúe de forma estable y rápida, limitándolo para que su respuesta no sea agresiva. Debe utilizar el microcontrolador de la tarjeta de desarrollo Tiva C (TM4C123G), siendo el lenguaje/entorno de programación cualquiera de los siguientes: Energia, Code Composer, Keil. Las constantes del sistema deben ser variables, ya que serán enviadas al inicio desde la computadora, a través de UART.
- **Interfaz de usuario y datalogger:** Desde una computadora (o Raspberry PI) se deberá ejecutar una aplicación en consola (programada en Python) para enviar los valores de inicialización al microcontrolador: K_p , K_i , K_d , Int_{max} , Int_{min} . Asimismo, deberá generar (como mínimo) 3 archivos: LOG (.csv), Página Web (.html), Gráfica (.png)

- **LOG:** Deberá almacenarse la temperatura cada 30 segundos, utilizando un archivo de texto separado por comas llamado **tempLOG.csv**. La primera columna debe ser un *Timestamp* y la segunda, la temperatura en grados Celsius con resolución de 0,1 deg C. La información debe provenir del microcontrolador. Se recomienda utilizar *handshaking* para este cometido.
- **Servidor Web:** Deberá levantar un servidor Web (Apache2, por ejemplo) para poder mostrar allí la página con la gráfica de *Temperatura vs Tiempo*, actualizándose cada 30 segundos (se sugiere el uso de la etiqueta *meta refresh*).
- **Gráfica:** Utilizando *Matplotlib* deberá crear una gráfica con las últimas 25 muestras, siendo el eje Y la Temperatura (deg C) y el eje X el *Timestamp*. Ésta no deberá mostrarse en pantalla, sino almacenarse como imagen (.png) en la misma carpeta donde se encuentra la página web.

¿Cómo implementar la electrónica con el PID?

Con el fin de mantener la temperatura en un valor constante, es necesario que el controlador PID sea capaz de *calentar* y *enfriar* el sensor de temperatura. Obviamente, esto no afectará al ambiente completo, pero con fines prácticos, el entorno alrededor del sensor basta.

El sensor puede ser un LM35 y debe estar conectado a cualquiera de las entradas analógicas de la tarjeta de desarrollo.

Para enfriar el sensor, puede utilizar el mismo motor de la tarea anterior, agregándole aspas (de cartón, papel, etc.) o conseguir un ventilador de computadora de 5V. La velocidad del ventilador se recomienda sea regulada por un MOSFET 2N7000.

Para calentar el sensor se sugiere utilizar una resistencia de 27Ω , conectada a 5V, a través de otro MOSFET 2N7000.

Siempre debe tomar en cuenta que la corriente máxima que soporta el puerto USB 2.0 de la mayoría de computadoras es de 500mA. Si tiene alguna duda de cómo calcular el consumo de todo el sistema, pregunte a su profesor. Al sobrepasar la corriente el umbral máximo, el puerto se deshabilita por protección, y su proyecto queda sin alimentación.

Restricciones y Limitaciones

Para tener derecho a calificación, deben declararse como constantes los siguientes parámetros:

- Tiempo de refresco de la página Web (por defecto, 30 segundos)
- Período de muestreo de temperatura (por defecto, 30 segundos)
- Cantidad de muestras a desplegar en la gráfica (por defecto, 25)
- Tiempo de refresco del PID dentro del microcontrolador (a su discreción)

La implementación en Python deberá realizarse con Programación Orientada a Objetos. Queda a discreción del grupo la organización de las clases. No es necesario soldar la circuitería en placas, ni tampoco entregar en protoboard. Simplemente pueden realizar las conexiones con cables, siempre y cuando éstas se mantengan firmes y que no representen un riesgo para causar corto-circuitos.

Finalmente, deberá utilizar el LED RGB de la tarjeta de desarrollo Tiva C para representar el estado actual del sistema. El objetivo es mostrar el estado actual de la temperatura, comparándola con el Set-Point en al menos dos colores (azul y rojo). La intensidad debe variar, dependiendo del error.

Los grupos que entreguen todo el trabajo en un sistema auto-contenido (por ejemplo, que quepa en la palma de una mano) de forma ordenada, utilizando la Raspberry PI obtendrá una ponderación extra de **15 puntos**.

Documentos de ayuda

- Controlador PID discreto en Energia:
<http://goo.gl/LGg2ir>
- Hoja de datos LM35:
<http://goo.gl/qBVSbd>
- Configurar servidor Web Apache en Raspberry PI:
<http://goo.gl/FokkN0>

Metodología de entrega

Con el fin de agilizar el proceso de calificación y evitar hacinamiento, la entrega será a través de un formulario en la web, pero la calificación individual se realizará por bloques (divididos por números de carnet). Esta calendarización será publicada luego de la fecha de entrega.

El proyecto se trabaja de forma individual o en parejas, y cualquier indicio de copia (parcial o total) será motivo de anulación inmediata. El límite de la entrega del proyecto es el miércoles **15/10/2014** a las **18:00 hrs**. Por ninguna razón se recibirán proyectos por otro medio, ni fuera de tiempo. Este proceso será via web: <http://goo.gl/erV7z7>. Deben enviar su examen solamente una vez, de lo contrario no será calificado. *Todo debe estar debidamente indentado y*

comentado. Al momento de la calificación deberán llevar 2 COPIAS IMPRESAS de la última página de este documento. **Por favor, lleven la computadora/Raspberry PI en la que han realizado el proyecto**, con el fin de evitar estar cambiando configuraciones a la hora de entrega.

Calificación

Carnet 1: _____ Carnet 2: _____

Descripción	Calificación
Uso adecuado de clases en Python	/20
LOG en Python	/10
Comunicación Serial/Bluetooth en Python	/5
Gráfica en Matplotlib	/15
Página web y servidor Apache	/5
Implementación de Hardware	/15
PID en microcontrolador	/10
UART en microcontrolador	/5
Representación LEDs	/5
Respuesta del PID	/10

Nota:

Calificó: