UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Departamento de Ingeniería Electrónica

Técnicas Digitales III

Guía de Trabajos Prácticos

Segundo cuatrimestre

Ciclo lectivo 2023 **Plan 95A**

1. INTERACCIÓN CON EL KIT "BEAGLEBONE BLACK"

a) Familiaricese con el hardware de la BeagleBone Black (BBB). Siga los pasos documentados en la Wiki de la cátedra
http://wiki.electron.frba.utn.edu.ar/doku.php?id=td3:bbx

b) Pruebe los ejemplos de código provistos por la cátedra. Interactúe con la misma mediante SSH y SCP.

2. EJERCICIO INTEGRADOR

Objetivo

El presente desarrollo tiene por objeto obtener, por encuesta y en forma remota, lecturas de un sensor y visualizarlas en el navegador de una PC.

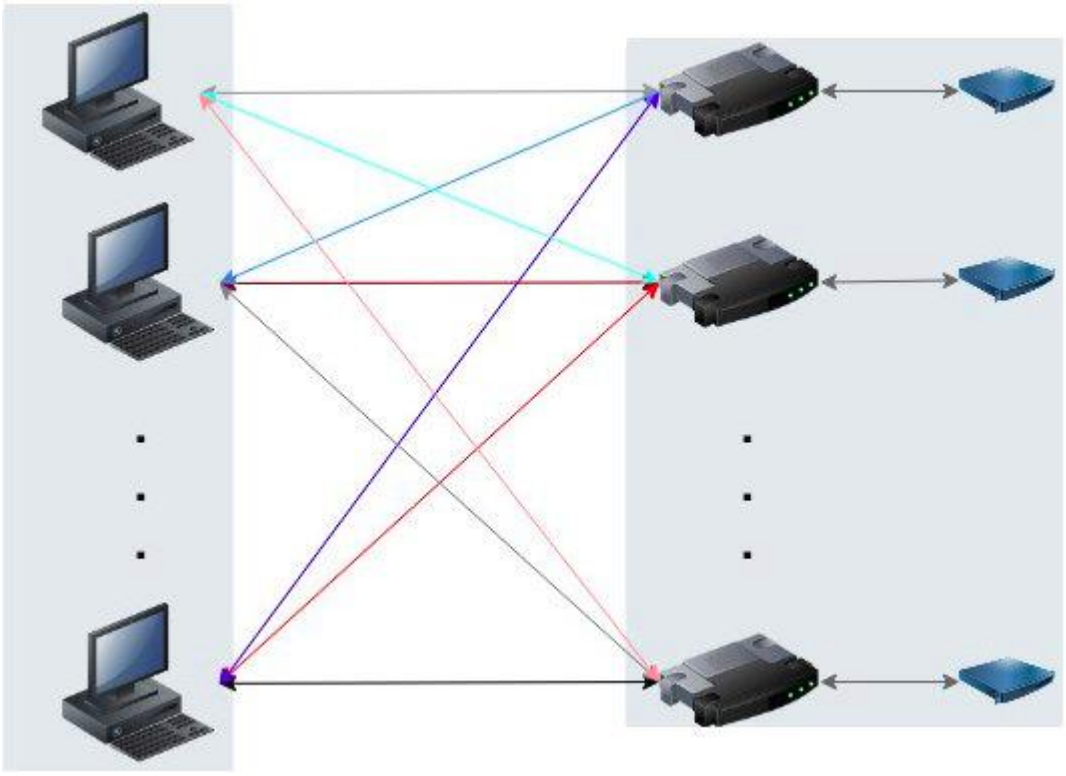
Requisitos

Se debe adquirir un dispositivo (sensor) que cumpla con los siguientes requisitos:

- Poseer **interface I2C**, operando en modo esclavo.
- Disponer de al menos un registro de configuración.

Arquitectura

El sistema a implementar debe estar compuesto por uno o varios dispositivos de adquisición (BeagleBone Black) junto al respectivo sensor elegido, que serán accedidos desde cualquier terminal (navegador web), según se detalla en la siguiente imagen:



Estación de visualización **Dispositivo de adquisición**

Estación de visualización

Esta parte del proyecto solo se limita a hacer solicitudes periódicas de contenido HTML al servidor y presentarlo en pantalla. A tal fin se utilizará el navegador de una PC, ingresando como URL la IP correspondiente al dispositivo de adquisición a interrogar.

Dispositivo de adquisición

El dispositivo de adquisición deberá satisfacer las siguientes funcionalidades y requerimientos:

1. Gestionar el dispositivo de medición (sensor), mediante un controlador (*driver*) que satisfaga el Linux Kernel Driver Model (LKDM). El mismo deberá satisfacer la API de un dispositivo de caracteres (open, close, read, write, ioctl), ser un módulo de kernel (*.ko*), gestionar el acceso y configuración del sensor elegido mediante la interfaz I2C en modo maestro e inicializar el periférico correspondiente, empleando parámetros de configuración establecidos en el *device tree*. El módulo deberá poder instalarse y desinstalarse en forma dinámica.
2. Recolectar, procesar y almacenar los datos del dispositivo de medición.
3. Mantener activo un servidor TCP concurrente mediante el cual se recibirán las solicitudes periódicas desde las estaciones de visualización y se enviarán los datos como paquetes cuyo contenido responde al protocolo HTTP. El servidor deberá cumplir los siguientes requisitos / funcionalidades:

3.1. ***Control de conexiones y pedidos de conexión***:

Limitar la cantidad de pedidos de conexiones simultáneos (en cola o *backlog*), así como la cantidad máxima de procesos que se creen para atender a los clientes que requieran datos (dicho de otra forma, la cantidad de conexiones activas).

En tal sentido el servidor dispondrá de un archivo de configuración de texto plano, en donde a razón de un parámetro por línea se configuren los parámetros de trabajo.

Si no existe el archivo de configuración, por defecto el *backlog* es de 2, y la cantidad máxima de conexiones activas es 1000.

3.2. ***Adquisición de datos***

Adquirir lecturas del dispositivo, cuyo controlador (*driver*) se desarrolló en el ítem anterior, en forma periódica con una frecuencia establecida como parámetro en la tercer línea del archivo de configuración. Si en el arranque no existe tal archivo adoptará un valor por defecto de 1 seg.

3.3. ***Estadísticas.***

Mantener una media móvil de una cantidad de lecturas configurada en la cuarta línea de texto del archivo de configuración. De no existir el archivo adoptará un valor por defecto de 5.

3.4. ***Refresco de configuración***

Los valores de *backlog*, cantidad máxima de procesos/hilos, frecuencia de lectura y media móvil, podrán modificarse editando el archivo de configuración en cualquier momento de operación sin reinicio del server. Para ello una vez alterados los parámetros en el archivo de configuración, se enviará una señal **SIGUSR1** al server en cuyo handler se leerán del archivo de configuración los nuevos valores y los establecerá como parámetros de trabajo hasta recibir otra vez la señal de refresco.

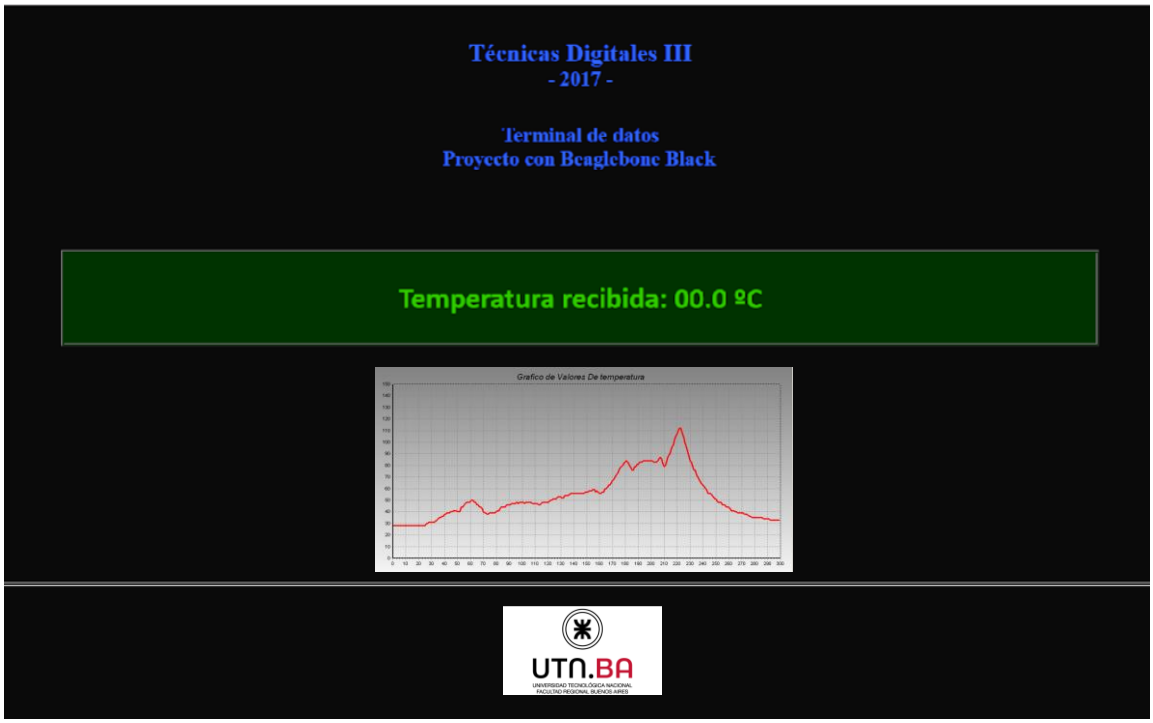
3.5. Transmisión de datos

Dado que este servidor se accede de manera remota con cualquier navegador de internet estándar, los paquetes deben cumplir con el protocolo HTTP y además se deberá esperar requerimientos de conexión por el puerto 80.

El contenido de estos paquetes HTTP será HTML plano (como cadena de caracteres o *string*). La Cátedra proveerá una función con el código necesario para darle el formato adecuado a los paquetes, la misma solo se deberá invocar con los argumentos necesarios para insertar los valores adquiridos.

La interacción entre los mensajes que envía el navegador y las respuestas a cada uno por parte del server se detalla en el Apéndice I.

Se muestra en la siguiente imagen el formato de la página a visualizar, a modo de referencia, debiéndose ajustar a los parámetros que se adquieran según el dispositivo de medición.



Opcional: Incluir el código necesario para poder enviar un gráfico con las lecturas históricas. Para ello, se sugiere utilizar un *wrapper* en C de gnuplot. En caso de optar por esta librería, preste atención a la documentación y procure instalar gnuplot via apt.

3. PRIMER RECUPERATORIO

Agregar al código del driver un contador del número de lecturas del parámetro medido realizadas desde que se instaló el módulo. El driver deberá retornar el valor del contador a través de ioctl. La media móvil del servidor debe reemplazarse por el valor medio, utilizando el contador indicado precedentemente.

4. SEGUNDO RECUPERATORIO

En caso que un proceso intente abrir el dispositivo cuando éste ya está siendo utilizado, el mismo deberá dormir hasta que se libere el recurso.

Apéndice I

Noción de la comunicación HTTP:

En el marco del protocolo HTTP, el navegador (cliente) enviará una de las siguientes solicitudes GET (***request*** en términos HTTP), cuyas respuestas (***status*** en términos HTTP) por parte del server se especifican en cada caso:

1. Solicitud de contenido:
   Request:
   GET / HTTP/1.1

   Respuesta:
   Si el método es correcto:
   HTTP/1.1 200 Ok
   Content-Type: text/html; charset=UTF-8\n
   Content-Length: %d\n El número es: strlen(HTML)
   \n
   … ***acá escribir el código HTML …***

   Si el método solicitado no es GET:
   HTTP/1.1 400 Bad method
   Si el recurso no es /
   HTTP/1.1 404 Bad resource

2. Solicitud de elemento gráfico:
   Request:
   GET /Archivo-grafico HTTP/1.1

   Respuesta:
   Si el método es correcto:
   HTTP/1.1 200 Ok
   Content-Type: image/png\n
   Content-Length: %d\n El número es: sizeof(buffer_imagen)
   \n
   … acá copiar el gráfico en binario …

   Si el método solicitado no es GET:
   HTTP/1.1 400 Bad method
   Si el recurso no es /Archivo-grafico
   HTTP/1.1 404 Bad resource

3. Para errores 400 o 404:

Content-Type: text/html; charset=UTF-8\n
Content-Length: %d\n El número es: strlen(HTML)
\n
… acá escribir el código HTML que muestre el error en pantalla …