

Tarea 1

Universidad de Chile Facultad de Ciencias Físicas y Matemáticas Departamento de Ciencias de la computacion CC5326 – Diseño de Internet de las Cosas

Link: <https://iiot-embedded.com/iiot/tareas/t1/>

Profesor: Luciano Radrigan F. **Auxiliar:** Alberto Abarzua P.

Objetivos a cumplir:

- Generar una red WiFi a través de la Raspberry Pi (con nombre y contraseña), la cual otorgue IPs a los ESP32 mediante DHCP.
- Programar un servidor de socket TCP y UDP en la Raspberry Pi.
- Programar un cliente de socket TCP y UDP en el ESP32.
- Enviar un paquete desde el ESP32 con datos generados en el mismo dispositivo, el cual será recibido por el servidor.
- Almacenar la información en una base de datos PostgreSQL utilizando Docker (se proporcionará todo lo necesario para esto).
- Posibilitar el cambio del protocolo de envío de datos desde la base de datos.
- Permitir el cambio del tipo de conexión (TCP o UDP) desde la base de datos.
- Los cambios en la base de datos deberán reflejarse en el ESP32 (por ejemplo, si se cambia el protocolo de envío, el ESP32 deberá enviar los datos con el nuevo protocolo).
- Deberá ser posible trabajar con ambos ESP32 al mismo tiempo; es decir, se debe poder recibir datos de ambos ESP32 de forma simultánea y almacenarlos en la base de datos.

Indicaciones

Flujo de los datos

1. El microcontrolador se inicia a través de una conexión TCP.
2. Realiza una consulta a la base de datos para obtener el valor de las variables ID_protocol y Transport_Layer. Con estos valores, determinará qué protocolo de paquete y tipo de conexión (TCP o UDP) utilizará.
3. Con esta información, procederá al envío de paquetes según las siguientes especificaciones:
 - En caso de usar TCP: enviará el paquete de datos y entrará en modo Deep Sleep por 60 segundos, tras lo cual repetirá el proceso.

- En caso de usar UDP: enviará los datos de forma continua hasta que el valor de Transport_Layer cambie. Este valor deberá poder ser cambiado manualmente (por ejemplo, apretando un botón).
4. Los paquetes recibidos por la Raspberry Pi deberán ser descompuestos y almacenados en una base de datos SQL.
 5. En cada envío de datos, se deberá verificar la configuración de la base de datos. Si ha habido algún cambio, se deberá ajustar el protocolo de envío o el tipo de conexión, según corresponda.

Encabezados de los paquetes

Cada paquete deberá contar con su propio encabezado, que incluirá la siguiente información. Este será común para todos los protocolos de envío:

Header	Data
<2 bytes>	ID (<i>ID unico del mensaje</i>)
<6 bytes>	Device MAC
<1 bytes>	Transport Layer (<i>TCP o UDP</i>)
<1 bytes>	ID Protocol
<2 bytes>	Length (<i>Headers + Body</i>)

Protocolos de envío

Los protocolos de envío determinan la forma y el contenido que deben seguir los paquetes enviados por los ESP32. Estos paquetes se enviarán a través de TCP o UDP, dependiendo de la configuración asignada a la ESP32, la cual se encuentra en la base de datos.

Valores	Protocolo 0	Protocolo 1	Protocolo 2	Protocolo 3
Headers
<1 bytes> Data 1	Batt level	Batt level	Batt level	Batt level
<4 bytes> Data 2		Timestamp	Timestamp	Timestamp
<1 bytes> Data 3			Temp	Temp
<4 bytes> Data 4			Press	Press
<1 bytes> Data 5			Hum	Hum
<4 bytes> Data 6			Co	Co

Valores	Protocolo 0	Protocolo 1	Protocolo 2	Protocolo 3
<4 bytes> Data 7				RMS
<4 bytes> Data 8				Amp x
<4 bytes> Data 9				Frec x
<4 bytes> Data 10				Amp y
<4 bytes> Data 11				Frec y
<4 bytes> Data 12				Amp z
<4 bytes> Data 13				Frec z
Valores	Protocolo 4			
Headers	...			
<1 bytes> Data 1	Batt level			
<4 bytes> Data 2	Timestamp			
<1 bytes> Data 3	Temp			
<4 bytes> Data 4	Press			
<1 bytes> Data 5	Hum			
<4 bytes> Data 6	Co			
<8000 bytes> Data 7	Acc x - Array 2000 floats			
<8000 bytes> Data 8	Acc y - Array 2000 floats			
<8000 bytes> Data 9	Acc z - Array 2000 floats			
<8000 bytes> Data 10	Rgyr x- Array 2000 floats			
<8000 bytes> Data 11	Rgyr y- Array 2000 floats			
<8000 bytes> Data 12	Rgyr z - Array 2000 floats			

En resumen, cada protocolo de envío tiene la estructura Headers + Body. El “Body” contiene los datos que se enviarán, los cuales están especificados en las tablas anteriores. Este “Body” depende del protocolo de envío que se haya configurado en la base de datos.

Generacion de datos

Acceloremeter_kpi: Representa un sensor de vibraciones que mide en los tres ejes y calcula su promedio mediante la fórmula de la raíz cuadrada de la media (RMS). Los valores son float y se generan con las siguientes fórmulas:

- Ampx: Valores aleatorios entre 0.0059 y 0.12
- Freqx: Valores aleatorios entre 29.0 y 31.0
- Ampy: Valores aleatorios entre 0.0041 y 0.11
- Freqy: Valores aleatorios entre 59.0 y 61.0
- Ampz: Valores aleatorios entre 0.008 y 0.15
- Freqz: Valores aleatorios entre 89.0 y 91.0
- RMS: $\sqrt{\text{Ampx}^2 + \text{Ampy}^2 + \text{Ampz}^2}$

Acceloremeter_Sensor: Un medidor de aceleración que genera un vector de 2000 datos por eje (X,Y,Z) para sus dos parámetros: aceleración y velocidad angular. Los datos son float y se generan con las siguientes fórmulas:

- Acc_X: Valores aleatorios entre -16.0 y 16.0
- Acc_Y: Valores aleatorios entre -16.0 y 16.0
- Acc_Z: Valores aleatorios entre -16.0 y 16.0
- Rgyr_X: Valores aleatorios entre -1000 y 1000
- Rgyr_Y: Valores aleatorios entre -1000 y 1000
- Rgyr_Z: Valores aleatorios entre -1000 y 1000

THPC_Sensor (Temperatura-Humedad-Presión-CO): Representa un sensor de cada uno de estos aspectos. Los valores son enteros, excepto para CO que es float.

- Temp: Valor aleatorio entre 5 y 30
- Hum: Valor aleatorio entre 30 y 80
- Pres: Valor aleatorio entre 1000 y 1200
- CO: Valor aleatorio entre 30 y 200

Batt_Sensor: Representa el nivel de batería del aparato. El valor es un u_int8 aleatorio entre 1 y 100.

Base de Datos

La base de datos debe contener las siguientes tablas:

- **Datos:** Esta tabla almacenará todos los datos recibidos, incluyendo un sello de tiempo (timestamp) y el identificador del dispositivo (Id_device y MAC).

- **Logs:** Esta tabla registrará la información de cada conexión recibida por el servidor, incluyendo el ID_device, el tipo de capa de transporte (Transport_Layer), el protocolo utilizado y su timestamp.
- **Configuración:** Esta tabla contendrá las variables ID_protocol y Transport_Layer, que se utilizarán para configurar el envío de datos al iniciar la conexión del ESP32. Estas variables deben poder modificarse de alguna manera.
- **Loss:** Esta tabla almacenará el tiempo de demora en la comunicación (tiempo desde el momento de envío hasta el momento de escritura en la base de datos, calculado usando la diferencia entre esos timestamps) y la cantidad de pérdida de paquetes (packet loss) en bytes para cada comunicación que haya ocurrido.

Para la creación del servicio de base de datos, se proporcionará un servicio de Docker, el cual debe iniciarse con el siguiente comando en el directorio base del proyecto:

```
docker-compose up -d
```

Entrega

1. Una demostración en vivo del funcionamiento del sistema el día Viernes 26 de Mayo.
2. Subir el código a U-cursos (Plazo en U-cursos). Si es posible, incluya un enlace a un repositorio de Git donde se trabajó en la tarea.

Recomendaciones

- Para la gestión y visualización de la base de datos, considere usar un servicio de [Adminer](#) y un [ORM](#) como SQLAlchemy o Peewee (Peewee siendo el mas simple de los dos). Adminer permite acceder y modificar la base de datos desde una interfaz web, mientras que un ORM facilita la creación de tablas y la realización de consultas.
- Dado que la tarea involucra componentes físicos, habrá desafíos para probar el software en grupo. Se recomienda coordinarse para realizar pruebas en persona, manteniendo el código en un repositorio (por ejemplo, en GitHub). Si no pueden reunirse, consideren utilizar la extensión de VSCode Live Share.
- Al trabajar con conexiones de pines a través de cables, conéctelos con cuidado para evitar daños a la ESP32. Un mal manejo de los pines de alto voltaje o corriente podría dañar el dispositivo, especialmente si se coloca sobre una superficie metálica.
- Ante cualquier problema con el funcionamiento de los componentes, primero verifique si el problema es físico (por ejemplo, cables mal conectados o defectuosos) antes de investigar posibles errores de software. Consulte con el equipo docente si no puede resolver o identificar el problema.
- Si tiene dudas acerca de la tarea o cómo manejar los componentes, publique su consulta en el foro o comuníquese con el equipo docente.