Tarea 1

Profesor: Luciano Radrigan F.
Auxiliar: Ricardo Aravena P.

Cada equipo contara con una Raspberry Pi3 y una ESP32

Objetivos

1. A través de la Raspberry generar una red WiFi (con nombre y contraseña)
2. En la Raspberry programar un socket **TCP y UDP server**
3. En el ESP32 programar un socket **TCP y UDP client**
4. Se enviara un paquete desde ESP32 con datos generados dentro de la misma, el cual el servior debe guardar en una base de datos.
5. El paquete enviado desde el ESP32 deberá cumplir las siguientes características:

Tabla 1: Forma de los paquetes

Header					**Data**			
2 bytes	6 bytes	1 byte	1 byte	2 bytes	-	-	-	-
ID Device	MAC	Transport Layer	ID Protocol	Leng Msg	Data 1	Data 2	Data ...	Data N

- Los protocolos de envio que deberán probar son:

Tabla 2: Protocolos 0,1,2 y 3 .

Tamaño de los Datos ->		1 bytes	1 bytes	4 bytes	1 bytes	4 bytes	1 bytes	4 bytes	4 bytes	4 bytes	4 bytes	4 bytes	4 bytes	4 bytes	4 bytes
ID Protocol	**Leng Msg (bytes)**	**Data 1**	**Data 2**	**Data 3**	**Data 4**	**Data 5**	**Data 6**	**Data 7**	**Data 8**	**Data 9**	**Data 10**	**Data 11**	**Data 12**	**Data 13**	**Data 14**
0	6	Val: 1	Batt level	Timestamp											
1	16	Val: 1	Batt level	Timestamp	Temp	Press	Hum	Co							
2	20	Val: 1	Batt level	Timestamp	Temp	Press	Hum	Co	RMS						
3	44	Val: 1	Batt level	Timestamp	Temp	Press	Hum	Co	RMS	Amp x	Frec x	Amp y	Frec y	Amp z	Frec z

Tabla 3: Protocolo 4

Tamaño de los datos ->		1 bytes	1 bytes	4 bytes	1 bytes	4 bytes	1 bytes	4 bytes	6400 bytes	6400 bytes	6400 bytes
ID protocol	**Leng msg (bytes)**	**Data 1**	**Data 2**	**Data 3**	**Data 4**	**Data 5**	**Data 6**	**Data 7**	**Data 8**	**Data 9**	**Data 10**
4	24016	Val: 1	Batt Level	Timestamp	Temp	Press	Hum	Co	Acc X	Acc y	Acc z

Indicaciones

Flujo de datos

1. Microcontrolador inicia a través de conexión TCP
2. Realiza una consulta a la base de datos sobre valor de la variable $ID\_protocol$ y $Transport\_Layer$, con las cuales sabrá cual protocolo de paquete y tipo de conexión (TCP o UDP) deberá trabajar.
3. Con esto obtenido realizara el envió de paquetes con lo indicado, con la siguiente especificación:
    - **En caso de usar TCP:** enviara el paquete de datos y entrara en modo DeepSleep por 60 segundos, tras lo cual repetirá el proceso.
    - **En caso de usar UDP:** Se enviara los datos sin para hasta que el valor de $Transport\_Layer$ cambie, este deberá poder ser cambiado manualmente (por ejemplo apretando un botón).
4. Los paquetes recibidos por la Raspberry deberán ser abiertos y guardados dentro de una base de datos SQL.
5. Si desea puede programar un mecanismo para que el envió de datos se detenga completamente, apretando un botón por ejemplo.

Base de datos

La base de datos debe contener las siguientes tablas:

- **Datos**: Esta tabla tendrá todos los datos recibidos con su timestamp y el identificador del dispositivo (Id_device y MAC).
- **Logs**: Esta tabla tendrá la información de cada conexión que haya resivido el servidor, teniendo el ID device, el tipo de transport layer, cual protocolo y timestamp de este.
- **Configuración**: Esta tabla contendrá las variables $ID\_protocol$ y $Transport\_Layer$ con las cuales se configurara el envió de datos al iniciar la conexión de la ESP32. Estos deben poder cambiarse de alguna forma.

Generación de datos

Para generar los datos deberá implementar funciones que emulen el funcionamiento de los sensores, usando las siguientes indicaciones:

- **Acceloremeter_Sensor**: Un medidor de aceleración para este se generara un vector de 2000 datos por eje. Estos serán floats generados por la siguiente formula (n es un número cualquiera, deben ser distintos entre datos del mismo eje):
    - $Acc_x = 2 \cdot sin(2\pi \cdot 0.001 \cdot n)$
    - $Acc_y = 3 \cdot cos(2\pi \cdot 0.001 \cdot n)$
    - $Acc_z = 10 \cdot sin(2\pi \cdot 0.001 \cdot n)$

- **THPC_Sensor** (Temperatura-Humedad-Presión-CO_2): representa un sensor de cada uno de estos aspectos, genere su medición usando los siguientes valores:
  - $Temp$ = Valor aletorio entre 5.0 y 30.0
  - Hum = Valor aletorio entre 30 a 80
  - $Pres$ = Valor aletorio entre 1000 y 1200
  - CO_2 = Valor aletorio entre 30.0 y 200.0
- **Batt_sensor**: representando el nivel de bateria del aparato, debera ser un valor entre 1 y 100.
- **Acelerometer_kpi**: representa un sensor de vibraciones, midiendo en los tres ejes y sacando su promedio (RMS, root mean square), para sus valores use:
  - Amp_x = valor aletorio entre 0.0059 y 0.12
  - $Frec_x$ = valor aletorio entre 29.0 y 31.0
  - Amp_y = valor aletorio entre 0.0041 y 0.11
  - $Frec_y$ = valor aletorio entre 59.0 y 61.0
  - Amp_z = valor aletorio entre 0.008 y 0.15
  - $Frec_z$ = valor aletorio entre 89.0 y 91.0
  - $RMS = \sqrt{(Amp_x^2 + Amp_y^2 + Amp_z^2)}$

Entrega

Esta consistira de:

1. Una demostración en vivo del funcionamiento del sistema, el Viernes 14/10
2. Subir el código a U-cursos (misma fecha de entrega que la demostración). De ser posible entregue un link a un repositorio Git donde se trabajo la tarea.