UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL PARANÁ

Trabajo Práctico II

Informática II

Parte I: Comunicación serie

Desarrolle una aplicación gráfica de escritorio que permita controlar, a través de una comunicación serie sobre USB, el encendido y apagado del led presente en la placa de desarrollo NodeMCU (ESP32). Para esto considere que:

- El tiempo en el que el led está encendido y el tiempo que está apagado se deben poder configurar de modo independiente, por lo que el led puede estar totalmente encendido (tiempo encendido diferente de 0 y tiempo apagado igual a 0), totalmente apagado (tiempo encendido igual a 0 y tiempo apagado diferente de cero), o cualquier combinación (por ejemplo 100 ms apagado y 400 ms encendido).
- Una vez aplicada la configuración de los tiempos de encendido de encendido y apagado, el led debe repetir el proceso de actualización indefinidamente.
- El máximo tiempo al que deberán responder es de 2047 [ms].
- Los valores de configuración deben ser proporcionados desde una aplicación desarrollada en Qt y que mantenga comunicación a través de un objeto de la clase QSerialPort (o QExtSerialPort) y la UART0 disponible en la placa de desarrollo.
- Se propone utilizar widgets QSlider a fin de establecer los valores de configuración, utilizando el slot ValueChanged. Cada vez que uno de los dos widgets cambie su valor deberá enviarse la información correspondiente a la placa de desarrollo a fin de reflejar en el led la configuración establecida utilizando un protocolo que incluya, como mínimo, un encabezado (o header = 0x0B), dos bytes con la configuración de tiempo de apagado, dos bytes con la configuración de tiempo de encendido, un byte con la suma de comprobación y una marca de finalización (footer 0xB0).
- Una vez aplicada la configuración, la placa de desarrollo deberá contestar con un paquete que conste, como mínimo, de un encabezado, código indicando el éxito de la operación (0xCC por ejemplo), suma de comprobación y marca de finalización. Al recibirlo y verificar su correcta estructura, la aplicación deberá presentar un mensaje en una widget QList conteniendo tanto el mensaje original como su interpretación (éxito o fracaso de la operación).
- La aplicación desarrollada para el sistema embebido deberá contar, como mínimo, con dos tareas. Una dedicada al procesamiento de los datos provenientes de la UART y otra dedicada al control del led (encendido y apagado durante el tiempo correspondiente). La comunicación entre ambas deberá realizarse utilizando una estructura de datos de tipo cola (xQueue).
- Se recomienda, en todos los casos, procesar los datos provenientes de la comunicación serie utilizando una máquina de estados finita implementada a través de un vector de punteros a funciones.

Parte II: Red de sensores

Se requiere implementar una red de sensores basada en dispositivos ESP32 utilizando el protocolo MQTT como forma de intercambio de información. Para esto realice 3 aplicaciones:

- Nodo de registro: aplicación realizada en una placa de desarrollo NodeMCU32 que, conectado a una red WiFi, se vincule a un bróker MQTT transmitiendo valores registrados a través de un sensor o un valor correspondiente a una señal senoidal. Los datos deben enviarse cada 2.5 segundos. Además, según se reciban mensajes apropiados, debe permitir encender o apagar el led presente en la placa de desarrollo. El nodo es identificado a través de un número entero en el rango 1 – 5.
- Nodo de prueba: aplicación de escritorio realizada en Qt que se comporta como un nodo de prueba, donde el valor transmitido debe poder ser establecido a través de un dial y el led debe ser representado a través de una etiqueta. En lo que respecta a la identificación del nodo, debe realizarse a través de un número en el intervalo 6 – 1024 (puede ser un valor aleatorio, pero verificando en tiempo de ejecución que no existan dos nodos con el mismo identificador).
- Panel de control: debe permitir, una vez conectado a un bróker MQTT, identificar todos los nodos conectados al sistema (NodeMCU32 como aplicaciones de software) listando su identificador asociado (número entero) en una lista. Según se seleccionen dispositivos de esa lista, debe ser posible enviar un mensaje de apagado o encendido a los leds asociados a cada nodo. Además, también debe realizarse una gráfica de los valores recibidos de un nodo seleccionado a través de un QComboBox.

Para esto, se propone trabajar, como mínimo, con el siguiente diccionario:

- Tópico /ej02/cmd: a este tópico se subscriben todos los nodos. Al recibir el mensaje getId publican en el tópico /ej02/id con el valor asociado a su identificación como mensaje.
  El panel de control escribe el mensaje getId en este tópico a fin de que los nodos publiquen su identificación.
- Tópico /ej02/[id]/cmd: aquí id es la identificación del nodo. Los nodos se subscriben a este tópico. Al recibir los mensajes ledOn o ledOff se debe obrar en consecuencia encendiendo o apagando el led correspondiente o su representación en el caso de aplicaciones de escritorio.
  Desde el panel de control se publica en este tópico a fin de enviar un comando a un nodo particular.
- Tópico /ej02/[id]/sensor: a intervalos de 2.5 segundos, si el nodo se encuentra conectado al bróker, publica un número real que representa el valor proveniente de un sensor.
  La aplicación que funciona como panel de control se encuentra subscrita a este tópico (/ej02/+/sensor) y grafica los datos de aquel que se ha seleccionado. Idea: utilice el método split de la clase QString a fin de dividir el tópico en sus partes e interpretar a que nodo corresponde el valor recibido.

Como bróker puede utilizar localmente la aplicación emqx (https://www.emqx.io/), un bróker de su preferencia o el servidor de prueba mosquito.org (https://test.mosquitto.org/). En caso de utilizar este último, considere cambiar los tópicos utilizados a fin de garantizar que su aplicación no interactuará con mensajes publicados por otro grupo (por ejemplo reemplazar en todos los tópicos el fragmento ej02 por una palabra/código de su elección).
A modo de ejemplo se incluyen implementaciones de un nodo y un panel de control realizado en Qt.

Entrega:

- Las aplicaciones desarrolladas deben presentarse a través de un repositorio creado en gitHub a tal fin.
- La dirección del repositorio debe ser incluida en un informe que será entregado a través del campus virtual. Este informe deberá incluir:
    - Descripción del problema a resolver en cada caso.
    - Descripción del desarrollo realizado en cada ejercicio.
    - Dificultades encontradas.

Criterios de Evaluación:

1. Entendimiento del Problema:
    - 1: No comprende el problema ni los requisitos.
    - 2: Comprende parcialmente el problema y los requisitos.
    - 3: Comprende el problema y los requisitos, pero con ciertas lagunas.
    - 4: Comprende bien el problema y los requisitos.
    - 5: Demuestra un profundo entendimiento del problema y los requisitos.
2. Planificación y Organización:
    - 1: No hubo planificación ni organización evidentes.
    - 2: La planificación y organización fueron inadecuadas.
    - 3: Se hizo una planificación y organización aceptable.
    - 4: Se planificó y organizó de manera efectiva.
    - 5: La planificación y organización fueron excepcionales.
3. Desarrollo e Implementación:
    - 1: No se completó ninguna parte del trabajo práctico.
    - 2: La implementación es insuficiente y con errores graves.
    - 3: La implementación es adecuada, pero con algunos errores menores.
    - 4: La implementación es sólida con errores menores.
    - 5: La implementación es excelente sin errores importantes.
4. Testeo y Depuración:
    - 1: No se realizaron pruebas ni depuración.
    - 2: Se hicieron pruebas, pero no se depuraron los errores.
    - 3: Se realizaron pruebas y se depuraron algunos errores.
    - 4: Se realizaron pruebas exhaustivas y se depuraron la mayoría de los errores.
    - 5: Se realizaron pruebas exhaustivas y se depuraron todos los errores.
5. Documentación:
    - 1: No hay documentación o es incomprensible.
    - 2: La documentación es mínima y confusa.
    - 3: La documentación es adecuada pero podría ser más detallada.
    - 4: La documentación es buena y proporciona información útil.
    - 5: La documentación es excepcional, detallada y clara.
6. Presentación:
    - 1: La presentación es incoherente y difícil de seguir.
    - 2: La presentación es confusa y poco atractiva.
    - 3: La presentación es aceptable pero podría mejorarse.
    - 4: La presentación es atractiva y facilita la comprensión.
    - 5: La presentación es excepcional, visualmente atractiva y fácil de seguir.