Protocolo de Comunicacion Serial

Comunicacion por Xmodem y Qt serial

1st Díaz Rodríguez Andrés Felipe
Ingeniería Electrónica
Universidad Sergio Arboleda
Bogotá, Colombia
andres.diazrodriguez@correo.usa.edu.co

2nd Novoa Neisa Lorena
Ingeniería Electrónica
Universidad Sergio Arboleda
Bogotá, colombia
lorena.novoa2@correo.usa.edu.co

3th Arcila Quevedo Carlos Andrés
Ingeniería Electrónica
Universidad Sergio Arboleda
Bogotá, colombia
carlos.arcila@correo.usa.edu.co

***Resumen*—En el siguiente informe se va a mostrar cómo se implementa la comunicación serial entre dos dispositivos por medio del protocolo de xmodem, con el propósito de demostrar que la comunicación es efectiva se van a pasar documentos de texto e imágenes, para tener una comunicación exitosa es importante tener en cuenta una correcta recepción de los datos y la respuesta.**

***Abstract*—The following report will show how serial communication is implemented between two devices using the xmodem protocol, in order to demonstrate that communication is effective, text documents and images will be passed, to have a successful communication is It is important to take into account the correct reception of the data and the response.**

***Keywords*—protocolo de comunicacíon, xmodem, Qt, TeraTerm.**

I. Introducción

Para el proyecto se tomo como base de objetivos la guia de laboratorio presentada por el profesor [1]. los objetivos definidos por esta eran el de realizar una aplicacion con interfaz grafica que tuviera la capacidad de hacer comunicacion serial exitosa por medio del protocolo Xmodem. Con la utilizacion del framework Qt, que trae por defecto librerias y demas elementos graficos se pudo realizar esta aplicacion pensando en que a el usuario le sea facil de tranferir y almacenar datos y archivos.

Como base de pruebas tenemos el software Teraterm que nos permite hacer una comunicacion virutal entre dos puertos COM (en el caso de windows), la ventaja de este software es que ya trae implementado el protocolo de comunicacion Xmodem.

Este protocolo Xmodem tiene como característica, organizar los archivos lógicos en paquetes, los cuales conformarán la informacion enviada, la cual al ser recibida por ”pedazos”de archivo y el receptor podrá diferenciar de los datos erroneos y correctos, este es un protocolo de recuperacion de errores para transmision de ficheros entre dos dispositivos. Es importante recalcar sus caracteristicas:

- serial
- asíncrona
- envia 8 bits de datos
- sin paridad y un bit de stop

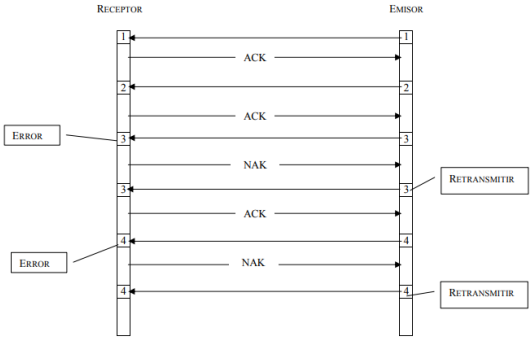

Figura 1. ARQ de envio y espera

Este protocolo retransmite un paquete cuando el receptor encuentra un error, a esto se le conoce como ARQ (Automatic Repeat Request) [2]

II. Objetivos

- Realizar una aplicacions utilizando el framework Qt para c++.
- Hacer una aplicacion con interfaz grafica que permita al usuario transferir datos por medio de la comunicacion serial utilizando el protocolo Xmodem.
- Almacenar en un archivo todos los datos recibidos por el puerto serial.

III. Metodología

Para realizar el laboratorio lo primero es crear la bases del proyecto en el IDE de Qt, en este caso como se desea hacer una interfaz grafica se utiliza un elemento especial del framework llamado widget. Ya teniendo el proyecto incial se debe implementar unos archivos de cabecera que nos permitiran implementar metodos necesarios para correr la aplicacion con la comunicacion serial.

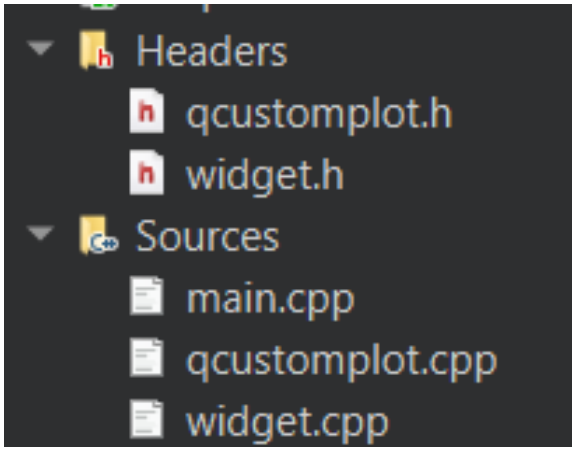

Figura 2. Headers proyecto

como parte del proceso, fue importante tomar en cuenta la documentacion del framwork Qt, ya que con la documentacion nos pudimos dar cuenta que un metodo de Qbyytearray nos retornaba valores en tipo char y esto era importante dado la logica del proyecto.

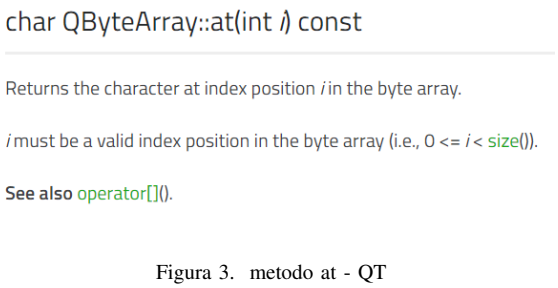

Figura 3. metodo at - QT

al ver la documentacion del framework tambien podemos ver que es importante manejar los datos de tipo unsigned, esto se debe a que por la cantidad de bits que se manejan en los archivos, el numero de paquetes puede llegar a ser superior a 128 y 256, utilizando datos de tipo unsigned no permite llegar a un limite mas alto de numero positivos y asi mitigar la posibilidad de bugs en el codigo.

luego de establecer la comunicacion utilizando el protocolo xmodem por medio de una aplicacion con interfaz grafica, se tiene que guardar los datos recibidos en un archivo, en este caso de la aplicacion los datos estan siendo almacenados en un archivo tipo txt, esto puede generar ciertos errores al enviar otro tipo de archivos (png, jpg, etc.) por lo que es importante tener en cuenta mejorar esa parte del codigo.

IV. Resultados

para esta practica se obtuvo un buen check del crc mas sin embargo, no se logro transmitir correctamente, tampoco recibir correctamente imagenes,asi como una dificultad que fue la llegada de los paquedes con valor incorrecto, o con un tamaño incorrecto como se puede apreciar en las siguiente imagenes se puede ver el funcionamiento de el codigo:

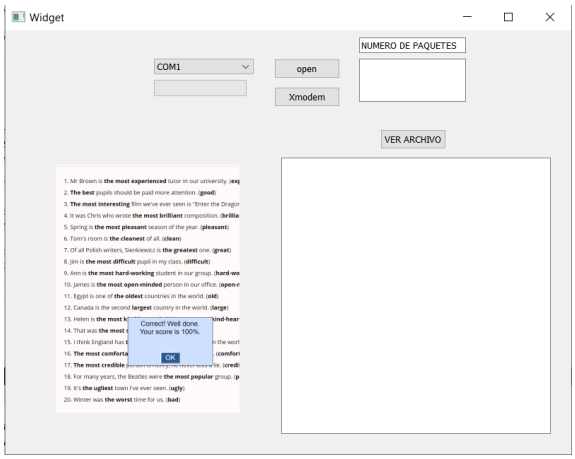

Figura 4. interfaz

En la siguiente imagen podemos ver como se ven los archivos de texto recibidos:

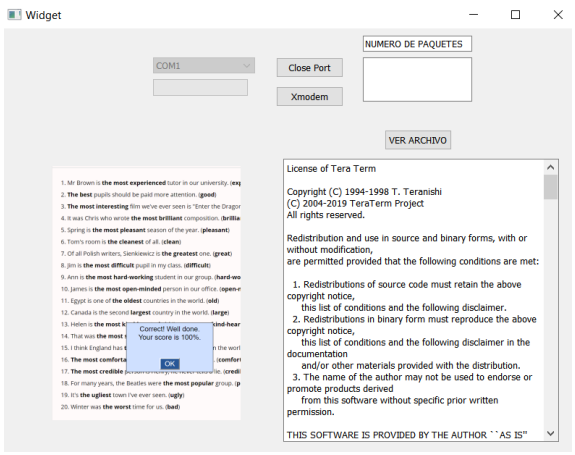

Figura 5. archivo mostrado en la interfaz

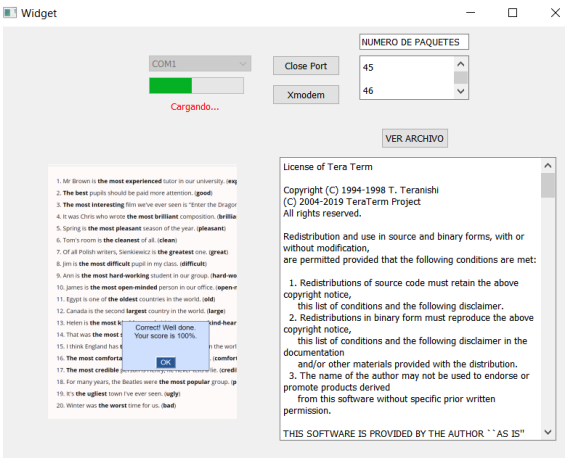

Figura 6. barra de carga para los archivos

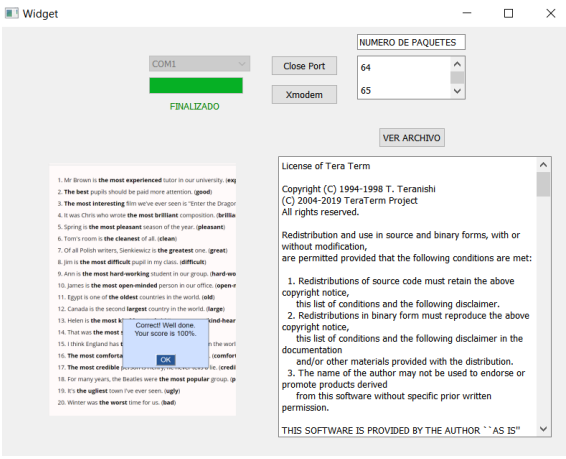

Figura 7. muestra que se finalizo la carga del archivo en tiempo de ejecucion

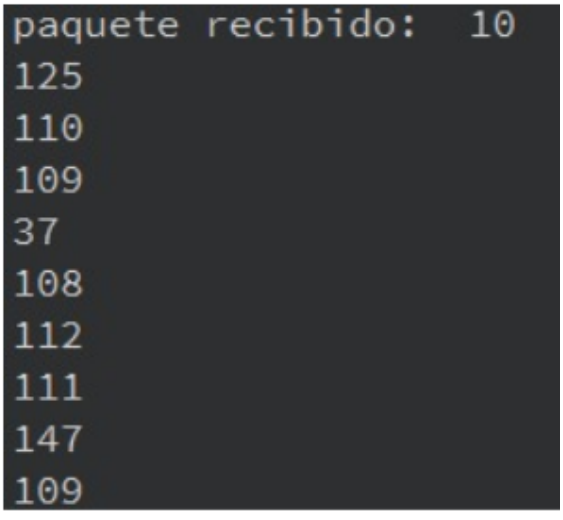

Figura 8. tamaño del paquete 10 incorrecto.

V. Análisis de resultados

- se debe hacer una mejor implementacion del protocolo Xmodem al momento de guardar las imagenes recibidas del **TERA TERM**, ya que fue posible recibir las imagenes mas no guardarlas con su respectiva extesion ya sea *.JPG* o *.PNG* .
- para los packetes cuando no se recibe su cantidad exacta en tamaño se envia una respuesta de error solicitando el ultimo packet de nuevo, con el fin de solucionar, el error cuando se envian mas de 128 packets y el tamaño es erroneo.

VI. Conclusiones

- es importante tener en cuenta que el metodo .at() del qBytearray retorna un caracter, ya que si no se compara con una variable de tipo char nos arroja error despues del packet 128.
- algunas variables como lo son quint16 y quint8 deben ser utilizadas ya que se bebe manejar limites positivos en los valores de los packets.
- se debe enfatizar mas en el uso del protocolo **XMODEM** para recibir imagenes ya que fue correcto recibir las imagenes mas no mostrarlas ya que, se guarda como un documento de texto con caracteres que no pueden ser escritos.

Referencias

[1] I. C. Camacho, “Laboratorio de xmodem,” *Universidad Sergio Arboleda*, 2020.
[2] T. Q. C. Ltd. (2020) Serializing qt data types.

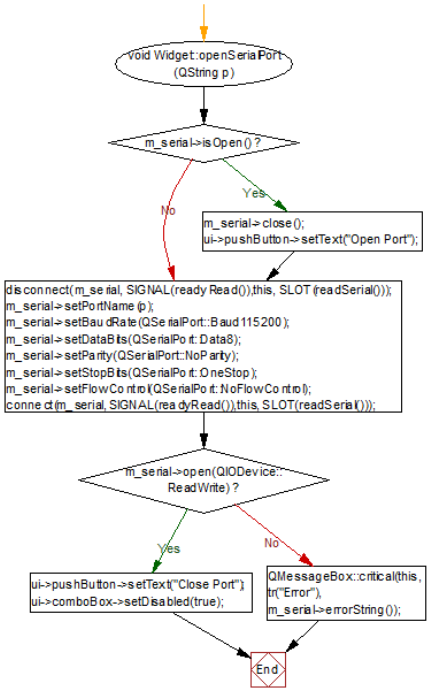

Figura 9.

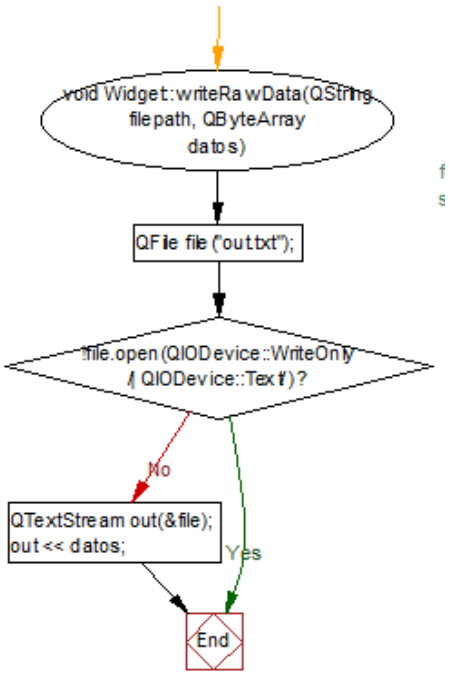

Figura 10.

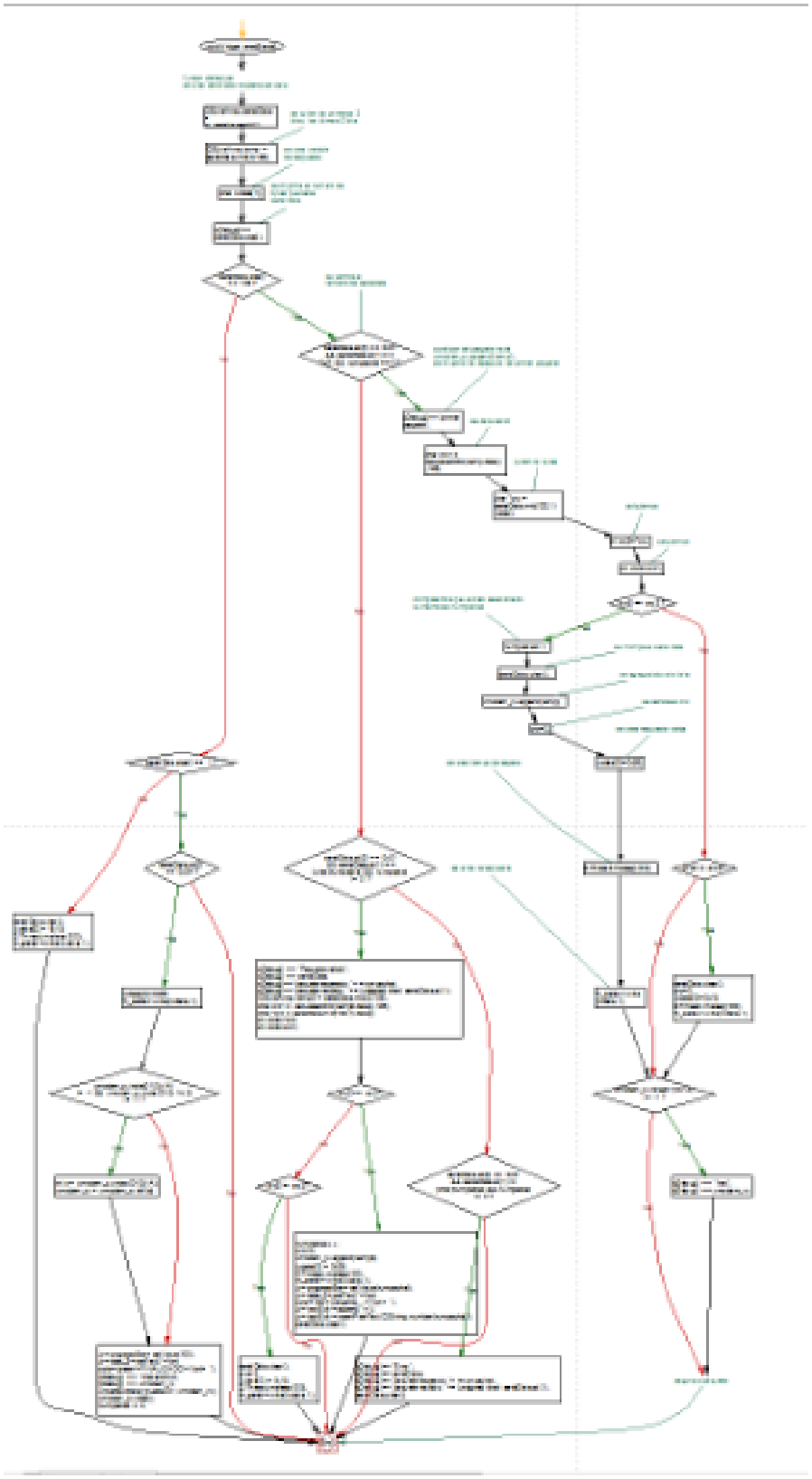

Figura 11.