



Protocolos de Comunicación y Sistemas Embebidos

Tarea 3: “Envío de mensajes con LoRa con capas 4 y 3 simplificadas”

Objetivos

Crear un dispositivo capaz de conectarse a la red “Protocolo V1” basada en LoRa en donde todos los nodos tendrán una comunicación P2P e implementada en capas de red, enlace y física.

Grupos de Trabajo

Se debe trabajar en grupos de 2/3 personas donde todos deberán trabajar en misma medida.

Requisitos

Para la fácil comprensión de su código es necesario comentar por secciones, describiendo qué es lo que se está llevando a cabo. En caso de usar funciones, describir qué hace, sus salidas y entradas.

La tarea debe incluir un archivo Makefile para compilar ambos programas. En caso de que los programas no compilen, **la tarea no será revisada**.

Los programas deben ejecutarse en los Kits suministrados por el docente.

Programas:

Se pide desarrollar **Dos Programas principales (con sus respectivos módulos)** uno denominado “**Modem**” el cual se ejecutará en la ESP32, El programa debe ser capaz de enviar y recibir mensajes dentro de la red LoRa en donde las IP de origen y destino corresponde al **número del KIT** con ayuda del hardware integrado en el dispositivo Heltec LoRa 32, a su vez comunicarse por medio de **UART** y el protocolo **SLIP** con un Segundo Programa llamado “**Nodo**”, debe contar con un menú que permita al usuario enviar los diferentes mensajes al “**Modem**”, además de recibir los mensajes de este último. En la **figura 1** se presenta el hardware a utilizar en donde la Raspberry se encuentra conectada directamente por medio de un cable USB al dispositivo Heltec LoRa 32, este dispositivo internamente cuenta una Radio LoRa (SX1276) conectada directamente a la ESP32.

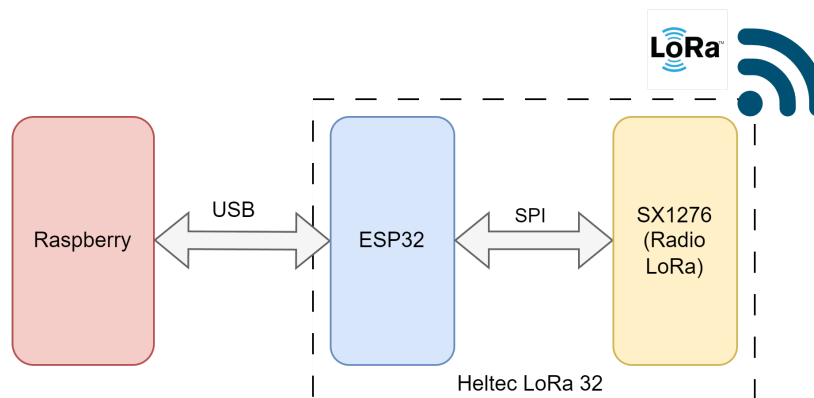


Figura 1: Esquema general del hardware a utilizar.



Protocolos de Comunicación y Sistemas Embebidos

1. Modem:

Este dispositivo compuesto por el hardware **Heltec LoRa 32** ya implementado y el software a desarrollar, debe contar con la capacidad de recibir y/o enviar los mensajes por medio de UART encapsulados por SLIP para interactuar correctamente con el Programa “**Nodo**” y intercambiar mensajes a través de LoRa con otros nodos de la red “Protocolo V1”.

1.1. Mensajes intercambiados con **Nodo**:

Estos mensajes enviados y/o recibidos por UART tienen las características mostradas en la Figura 2 en donde los mensajes deben estar encapsulados desde capa 4 hasta SLIP, en capa 4 se observa el uso del **protocolo propio** el cual debe ser el mismo utilizado en tareas 1 y 2, además del protocolo **Datos de Red** los cuales serán algunos de los comandos enviados por el protocolo Propio pero adicionalmente se agregan 4 tipos de mensajes mostrados en [tabla 2](#), los cuales se conocerán como ACK, mensaje Unicast (para comunicación p2p entre nodos), mensaje Hello (el cual se usará para que el nodo) y mensaje fragmentado (para el desafío). Cada mensaje a modo comando debe ser identificado con el campo “Protocolo” de IPv4 Simplificado mostrado en la [tabla 1](#).

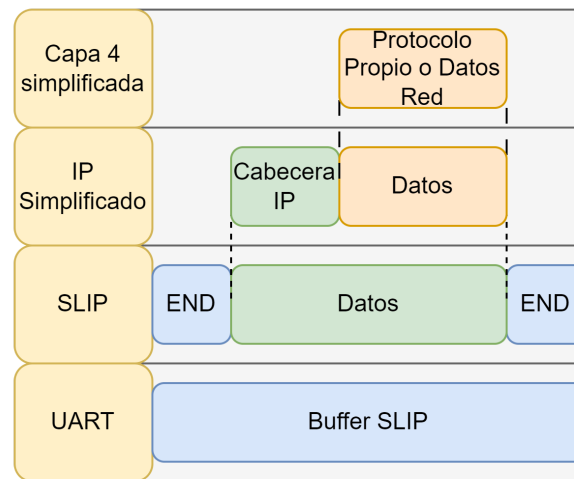


Figura 2: Capas utilizadas para intercambiar mensajes entre los dispositivos Raspberry-ESP32 por medio de UART.

De esta forma cuando el **Módem** recibe un dato desde **nodo** debe desencapsular el protocolo SLIP y trabajar leer los mensajes de **capa 3** verificar que el origen del mensaje es correcto (debe conocer su dirección) y realizar las siguientes acciones:

- Cuando el destino del mensaje sea el mismo (IP de destino == origen) y el campo **Protocolo** sea igual a 0, se considerará que los datos son de tipo protocolo propio y realizará las acciones correspondientes a la tarea 1 y 2, posteriormente el mensaje debe ser descartado.
- Cuando el destino sea diferente al nodo propio (IP de destino != origen) o sea un mensaje tipo broadcast (dirección 0xFFFF) y tenga el campo **Protocolo** diferente de 0, el nodo debe enviar directamente el mensaje por LoRa para ser dirigido a su destino manteniendo el



Protocolos de Comunicación y Sistemas Embebidos

mismo encapsulado en capa 3 referida a IP simplificado. En caso contrario el mensaje debe ser descartado.

- Cuando se reciba un mensaje por LoRa el dispositivo solo lo enviará por **UART** manteniendo el formato de la capa 3 cuando la dirección de destino del mensaje sea la del nodo o sea una dirección Broadcast.

1.2 Mensajes intercambiados por **LoRa**:

Para los casos en donde **Modem** recibe un dato desde la Radio LoRa estos vendrán desde la capa PHY (física) como se muestra en la figura 3 en donde los datos provienen de un encapsulamiento LoRa (El código para el funcionamiento de la radio LoRa será facilitado por el docente), para el caso de envío el dispositivo debe cumplir las condiciones establecidas en el punto 1.1 manteniendo el mismo encapsulado en capa 3 pero ahora utilizando las capas encerradas en el cuadro de línea discontinua de la figura 3.

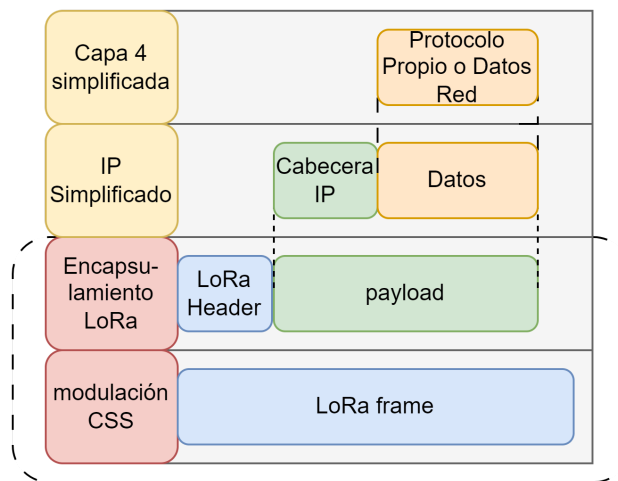


Figura 3: Capas utilizadas para intercambiar mensajes entre los dispositivos LoRa por medio Inalambrico.

2. **Nodo:** En cuanto al nodo este permitirá al usuario realizar las acciones mostradas en la figura 4 en donde se pueden observar la estructura recomendada del menú de usuario las cuales permitirán interactuar con la red. Además el nodo debe contar con los siguientes requerimientos:



Protocolos de Comunicación y Sistemas Embebidos

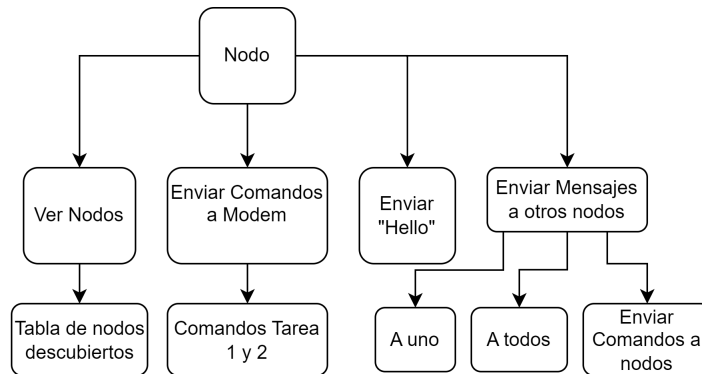


Figura 4: Posibles acciones y comandos del nodo.

- El nodo debe contar con un menú en donde el usuario pueda acceder a las siguientes funciones:
 - o Ver Nodos Disponibles: para este caso el nodo debe mostrar una Tabla de IP de origen de todos los nodos que hayan enviado un mensaje Hello y el tiempo transcurrido en segundos desde dicho evento.
 - o Contar con un menú de acceso a comandos internos del modem el cual debe permitir encapsular el protocolo propio en capa 3 y transmitirlo al modem.
 - o La función de envíos de mensajes hello permitirá al nodo ser descubierto por los demás nodos en la red.
 - o La posibilidad de enviar los mensajes desde el 2 hasta el 7 de la tabla 2.
- Adicionalmente cada vez que se envíe un mensaje que no sea Broadcast o Protocolo Propio o Hello (lista de mensajes disponibles en tabla 2), el nodo debe esperar por un mensaje ACK hasta 2 intentos antes de descartar el mensaje e informar el resultado al usuario.
- Los mensajes Hello recibidos no deben ser mostrados, pero si debe guardarse un registro del momento en que se recibió el mensaje junto con la IP para ser mostrados en cualquier momento en la tabla.
- Cada vez que el nodo reciba un mensaje desde el 2 hasta el 7 debe mostrar por consola el origen del mensaje y el mensaje en el caso que no sea un comando vacío, caso contrario debe indicar qué comando se recibió. Excepción del mensaje 4.
- Cada vez que el nodo reciba un mensaje desde el 5 hasta el 7 debe preparar un mensaje en el protocolo propio para ejecutar la operación correspondiente en el Modem.
- Todos los mensajes enviados deben ser encapsulados en el protocolo IPv4 modificado (Capa 3) y luego empaquetados en SLIP (Capa 2) para su posterior envío.
- Todos los menús por consola deben ser auto explicativos.

Puntajes Extra:

- **0.2 pts:** Por la utilización de desarrollo modular.
- **0.4 pts:** Por la utilización de archivo log que registra todos los mensajes recibidos tipo unicast.
- **0.3 pts:** Por menú de al menos 3 niveles e intuitivo.



Protocolos de Comunicación y Sistemas Embebidos

- **0.7 pts:** Si el grupo logra realizar el envío de un arreglo de 1024 bytes el cual representará una imagen en la pantalla OLED del receptor (el dispositivo receptor será el kit del docente), para lo cual se deben usar los campos de fragmentación del protocolo IP simplificado, en donde el Offset de Fragmento será la posición de los bytes a recibir en el arreglo, y cuando Flag Fragmento sea igual a 2 se mostrará la imagen.

Información Adicional:

En la tabla 1 se muestra el Protocolo IPv4 simplificado en donde se observan diferentes campos, con sus respectivos pesos en bits, para este caso el **Flag de fragmento**, el **offset de Fragmento** se consideran para uso en la sección de **puntos extras**, en donde el primer campo menciona indica con un 0 que el paquete no es fragmentado, con un 1 que es un fragmento y con un 2 que es último fragmento (el campo "Identificación" es el mismo para los fragmentos del dato fragmentado), los demás campos deben ser utilizados para la tarea de la siguiente manera, la **Longitud Total** de datos permite una capacidad de $2^8 - 1 = 255$ bytes el cual es tamaño máximo permitido por LoRa, la **identificación** es única por dato y es ingresada como un contador por el emisor (para paquetes fragmentados este identificador es el mismo), el campo **Protocolo** indica el tipo de paquete a recibir y/o enviar, la **Suma de verificación** es solo de la cabecera sin considerar las direcciones de origen y destino, la **IP de origen y destino** indican el direccionamiento de los nodos y los datos son el mensaje a enviar.

Tabla 1: Protocolo IPv4 Simplificado.

0	4	8	12	16	20	24	28	31
Flag Fragmento	Offset de Fragmento (12 bits)			Longitud Total (8 bits)		Relleno		
Identificador (16 bits)				Protocolo (8 bits)		Suma de verificación de Cabecera (8 bits)		
IP de origen (16 bits)				IP de destino (16 bits)				
Datos								
...								

El campo protocolo de IPv4 simplificado contará con la siguiente interpretación mostrada en la tabla 2 en donde se observa el valor numérico del campo, el tipo y la descripción.

Tabla 2: Valores posibles del campo **Protocolo** con su respectiva interpretación.

Código Protocolo	Tipo	Descripción
0	Protocolo Propio	Utilizado para interactuar internamente Nodo-Modem y desarrollado en tareas 1-2.



Protocolos de Comunicación y Sistemas Embebidos

1	ACK	Mensaje tipo unicast, utilizado para confirmar la correcta recepción de un mensaje, los datos deben interpretarse como la ID del mensaje confirmado.
2	Mensaje Unicast	Mensaje dirigido hacia un solo nodo, los datos deben interpretarse como texto plano.
3	Mensaje Broadcast	Mensaje dirigido hacia todos los nodos, los datos deben interpretarse como texto plano.
4	"Hello"	Mensaje broadcast utilizado para descubrir al nodo en la red por otros nodos. El mensaje debe ser un string "hola".
5	Prueba	Comando Vacío utilizado para que el dispositivo receptor muestre un mensaje o imagen de prueba en la pantalla OLED.
6	Estado Led	Comando Vacío utilizado para que el nodo receptor cambie el estado actual del LED integrado de su módem.
7	Mensaje en OLED	Comando utilizado para que el receptor muestre en su pantalla OLED el mensaje de texto.

Importante: Para esta tarea no considerar los 10 mensajes de prueba y sus estadísticas.

Informe:

Por definir.

Entrega

Debe ser subida a Campus Virtual la fecha indicada con el texto. Recuerde que en el archivo comprimido deben estar todos los archivos necesarios para el correcto funcionamiento de la tarea, este debe tener los apellidos de los integrantes en formato "Apellido1_Apellido2.zip". Además, se debe incluir una carpeta "doc" con el informe solicitado en latex y pdf.

Fechas

Entrega: 16 de Julio de 2024.



Protocolos de Comunicación y Sistemas Embebidos

Observación

Se descontará **1 punto** por día de atraso. **La copia será sancionada con un 1.0.** Recuerde documentar adecuadamente el código fuente, de lo contrario tendrá descuentos. **Se descontará puntaje si la tarea no cumple con los formatos solicitados.**

Material de apoyo:

Tutorial Pantalla OLED disponible en : [ESP32 OLED Display with Arduino IDE | Random Nerd Tutorials](#)