

COMUNICACIÓN MKR WAN 1310 POR LORA

CONTROL DE CONFIGURACION DE LORA

Aitana Carreño Cabeza Y Luis Sánchez García

Mediante dos dispositivos MKR WAN 1310, se implementan dos roles, uno como Master y otro como un Slave. Ambos se comunicarán mediante LoRa. El dispositivo Master se encargará de cambiar las configuraciones LoRa según la información que le proporcione el Slave. El Slave se encargará de recibir la configuración, aplicársela y enviar su nueva configuración al Master como un ACK.

Librerías usadas:

RTCZero (librería para controlar el módulo de tiempo real integrado en el Arduino)

LoRa (librería para el control y configuración de la antena LoRa)

Arduino_BQ24195 (librería para el control y configuración de la batería BQ24195)

SPI (biblioteca estándar para interactuar con dispositivos como sensores, pantallas que se comunican a través del bus SPI)

Ficheros extras:

BQ24195L_MIC.ino (Para comprobar el funcionamiento de la batería para la antena LoRa)

Dispositivo Master:

-Se incluye la librería RTCZero para el control de un timer que sirve para volver a una configuración conocida si el dispositivo Master no recibe ningún mensaje en un tiempo en concreto

-Se incluye la librería LoRa para el control y configuración de la antena LoRa para la debida comunicación entre los dos dispositivos.

-Se incluye la librería Arduino_BQ24195 para el control y configuración de la batería BQ24195 usada en varias placas de Arduino.

En el Set-up primero comprobamos que la batería funciona correctamente, luego configuramos la antena Lora con la configuración inicial. Activamos un timer con una cuenta atrás con el tiempo que hayamos decidido.

En el loop si no estamos trasmitiendo y ha pasado un cierto tiempo desde la última vez que se envió un mensaje enviamos un mensaje con un Payload que incluye el ancho de banda, el spreading factor, el coding rate y el txPower que tengamos en nuestro nodo de configuración, el RSSI y el SNR.

Si la flag de enviar ACK está en verdadera entonces enviamos un Payload de tamaño 0 para indicar que hemos recibido el mensaje con la configuración.

Cuando se active la flag de txDoneFlag comprobamos que se respeta el duty cycle de 1% (si no se cumple ajustamos la variable del intervalo de mensajes) ponemos la flag de Transmitir a falso y empezamos a escuchar por si nos llega mensajes.

Si se activa el timer, este activará una flag para que en el loop si la flag de sinRespuesta está a verdadero entonces llamamos a una función para reestablecer la configuración del dispositivo a una ya conocida.

Cuando el Master recibe el Payload con la configuración del Slave este calcula con los datos que ha enviado el Slave la nueva configuración del LoRa usando sobre todo el SNR y el RSSI.

Cuando el Master ya ha calculado la nueva configuración este lo envía por el mensaje a el Slave y luego al terminar la transmisión se aplica la configuración.

Funciones importantes:

-sendMessage(uint8_t* Payload, uint8_t PayloadLength, uint16_t msgCount) Es la principal función para enviar un mensaje en LoRa

- onReceive(int packetSize) Es la función que llama LoRa cuando recibe un mensaje, desde esta función verificamos que el mensaje es el correcto y llega para nosotros. Luego mostramos el mensaje y la configuración que trae el Payload para luego actualizar la configuración LoRa según los parámetros que haya en él.

- TxFinished() es una función para activar la flag de TxDoneFlag para indicar la finalización de retransmisión y no es el primer mensaje que envía entonces aplica la configuración LoRa que ha calculado en su antena LoRa.

- printBinaryPayload(uint8_t* Payload, uint8_t PayloadLength) es una función que nos permite imprimir el valor de un Payload que enviamos por LoRa.

- restaurarConfiguracion() es una función que restablece la configuración a una configuración ya conocida.

- NoRespuesta() función que se llama cuando se dispara el timer para activar una flag.

-reiniciarTiempoRTC() es una función que nos permite reiniciar el temporizador para que no salte cuando recibimos un mensaje.

- actualizarConfiguraciones () es una función que calcula una nueva configuración de LoRa según la información que nos proporcione el Slave. Es importante destacar que solo se actualiza en este caso el nodo y no la antena LoRa.

- ajustarPotencia() Es la función encargada de ajustar la potencia de la antena LoRa según los datos del Slave.
- gestionarAnchoBanda() Es la función encargada de ajustar el ancho de banda de la antena LoRa según los datos del Slave.
- ajustarFactorDispersion() Es la función encargada de ajustar el factor de dispersión de la antena LoRa según los datos del Slave.
- aplicarParametrosLora() Es la función encargada de aplicar la configuración del Nodo a la antena LoRa.

Dispositivo Slave:

- Se incluye la librería RTCZero para el control de un timer que sirve para volver a una configuración conocida si el dispositivo Slave no recibe ningún mensaje en un tiempo en concreto
- Se incluye la librería LoRa para el control y configuración de la antena LoRa para la debida comunicación entre los dos dispositivos.
- Se incluye la librería Arduino_BQ24195 para el control y configuración de la batería BQ24195 usada en varias placas de Arduino.

En el Set-up primero comprobamos que la batería funciona correctamente, luego configuramos la antena Lora con la configuración inicial. Activamos un timer con una cuenta atrás con el tiempo que hayamos decidido.

En el loop si no estamos transmitiendo y ha pasado un cierto tiempo desde la última vez que se envió un mensaje enviamos un mensaje con un Payload que incluye el ancho de banda, el spreading factor, el coding rate y el txPower que tengamos en nuestro nodo de configuración, el RSSI y el SNR.

Cuando se active la flag de txDoneFlag comprobamos que se respeta el duty cycle de 1% (si no se cumple ajustamos la variable del intervalo de mensajes) ponemos la flag de Transmitir a falso y empezamos a escuchar por si nos llega mensajes.

Si se activa el timer, este activará una flag para que en el loop si la flag de sinRespuesta está a verdadero entonces llamamos a una función para reestablecer la configuración del dispositivo a una ya conocida.

Cuando el Slave recibe el mensaje, este aplica la configuración recibida y activa el flag del ACK para luego enviar al Master un Payload con su configuración actual.

Funciones importantes:

- sendMessage(uint8_t* Payload, uint8_t PayloadLength, uint16_t msgCount) Es la principal función para enviar un mensaje en LoRa
- onReceive(int packetSize) Es la función que llama LoRa cuando recibe un mensaje, desde esta función verificamos que el mensaje es el correcto y llega para nosotros. Luego mostramos el mensaje y la configuración que trae el Payload para luego actualizar nuestro nodo con esa configuración y activar el ACK
- TxFinished() es una función para activar la flag de TxDoneFlag para indicar la finalización de retransmisión
- printBinaryPayload(uint8_t* Payload, uint8_t PayloadLength) es una función que nos permite imprimir el valor de un Payload que enviamos por LoRa.
- actualizarConfiguraciones() es una función que nos permite aplicar los parámetros LoRa que nos ha pasado el Master en el Payload.
- reiniciarTiempoRTC() es una función que nos permite reiniciar el temporizador para que no salte cuando recibimos un mensaje.
- aplicarParametrosLora() es una función que aplica los parámetros a la antena LoRa
- restaurarConfiguracion() es una función que restablece la configuración a una configuración ya conocida.
- NoRespuesta() función que se llama cuando se dispara el timer para activar una flag.