

ESP-NOW MQTT Cliente con ESP8266

Este código está diseñado para un sistema de sensor basado en ESP8266 que utiliza ESP-NOW para la comunicación con una pasarela ESP-NOW -> MQTT. A continuación, se detallan sus componentes principales y funcionalidades:

Propósito

- **Transmisión de Datos:** El sensor envía datos en formato JSON a una pasarela ESP-NOW -> MQTT.
- **Identificación de la Pasarela:** La pasarela objetivo se identifica por su dirección MAC (2E:3A:E8:12:7D:0E).

Modo de Sueño Profundo (Deep Sleep)

- **Activación del Modo de Sueño:** Después de enviar datos o en caso de error o tiempo de espera, el dispositivo entra en modo de sueño profundo durante aproximadamente 30 segundos.

Configuración de Despertar

- **Conexión GPIO para Despertar:** Para despertar del sueño profundo, el GPIO16 debe estar conectado al pin RST (reset).
- **Precauciones Durante la Programación:** Esta conexión debe ser retirada al programar la placa.

Estructura del Código

El código para el sensor ESP8266 con ESP-NOW está estructurado en varias funciones clave, cada una con una responsabilidad específica en el sistema.

Función `byte2HEX(byte data)`

Convierte un único byte en una cadena de caracteres hexadecimal. Esta función es útil para visualizar datos binarios (como direcciones MAC) en un formato más legible. Utiliza operaciones bit a bit para convertir el valor numérico de un byte en su representación hexadecimal correspondiente.

Callback `fin_envio_espnw(uint8_t *mac, uint8_t sendStatus)`

- **Propósito:** Maneja la finalización de la transmisión de datos ESP-NOW.
- **Funcionalidades Clave:**
 - **Indicación de Estado de Envío:** Muestra si la transmisión fue exitosa o no.

- **Registro de Dirección MAC:** Imprime la dirección MAC del destinatario del mensaje, lo que ayuda en la depuración y el seguimiento de la transmisión.
- **Activación del Sueño Profundo:** En caso de éxito o fracaso en la transmisión, dirige al dispositivo a entrar en modo de sueño profundo para conservar energía.

Función setup()

Propósito

La función setup() prepara el dispositivo para su funcionamiento. Esta función se ejecuta una vez al iniciar el dispositivo y configura los componentes necesarios.

Pasos Clave

- **Inicialización del Puerto Serie:**
 - Configura la comunicación serial a 115200 baudios para la depuración y el monitoreo.
 - Imprime mensajes iniciales para confirmar el inicio del proceso de configuración.
- **Configuración del Sensor DHT:**
 - Inicializa el sensor DHT11 conectado al GPIO 5.
 - Lee los valores de temperatura y humedad del sensor.
 - Gestiona los valores no numéricos (NaN) asignando un valor predeterminado para evitar errores en el mensaje JSON.
- **Configuración de WiFi:**
 - Establece el modo WiFi del ESP8266 en modo estación (STA).
 - Desconecta de cualquier red WiFi previamente almacenada.
 - Imprime la dirección MAC del dispositivo para facilitar la identificación en la red.
- **Inicialización de ESP-NOW:**
 - Intenta inicializar el protocolo ESP-NOW.
 - Configura el rol del dispositivo en la red ESP-NOW como controlador.
 - Agrega un peer (dispositivo ESP-NOW) con la dirección MAC especificada.
 - Registra un callback para la finalización del envío de datos ESP-NOW.
 - En caso de fallo en la inicialización de ESP-NOW, manda el dispositivo a dormir para conservar energía.
- **Lectura de Datos de los Sensores:**
 - Espera el tiempo mínimo de muestreo del sensor DHT11.
 - Lee la temperatura, humedad y luminosidad (a través de un sensor analógico).
 - Mapea el valor del sensor analógico a un rango de 0 a 1000 para estandarizar los datos.
- **Construcción y Envío del Mensaje JSON:**
 - Construye un mensaje JSON con los datos recopilados de los sensores, incluyendo un campo de estado.

- Convierte el mensaje JSON en una cadena de texto y lo imprime para depuración.
- Envía el mensaje JSON a través de ESP-NOW a la dirección MAC especificada.

Función `loop()`

- **Propósito:** Contiene la lógica principal del programa que se ejecuta continuamente.
- **Funcionamiento:**
 - **Verificación de Tiempo de Espera:** Comprueba si se ha superado el tiempo límite para el envío de datos.
 - **Activación del Modo de Sueño:** Si se supera el tiempo de espera, invoca la función `gotoSleep` para minimizar el consumo de energía.

Función `gotoSleep()`

- **Propósito:** Gestiona la entrada del dispositivo en el modo de sueño profundo.
- **Características:**
 - **Registro de Tiempos:** Antes de dormir, registra y muestra el tiempo transcurrido en varias etapas del proceso, como el arranque, la configuración, y el envío de datos.
 - **Configuración del Sueño Profundo:** Pone el dispositivo en un estado de bajo consumo energético durante un tiempo definido, optimizando así la vida de la batería.

Este diseño permite una gestión eficiente de la energía y una comunicación robusta mediante ESP-NOW.

Transmisión de Datos

- **Lectura de Sensores:** Los datos de temperatura, humedad y luminosidad se leen desde un sensor DHT y un sensor analógico.
- **Empaquetamiento de Datos:** Los datos se empaquetan en un mensaje JSON junto con el estado del sensor.
- **Envío de Datos:** El mensaje JSON se envía a la pasarela ESP-NOW especificada.

Manejo del Sueño Profundo

- **Registro de Tiempos:** Antes de entrar en sueño profundo, el dispositivo registra varios sellos de tiempo para el arranque, la configuración, el envío de datos y el tiempo actual.
- **Optimización de Energía:** El uso del modo de sueño profundo y la desactivación de la calibración RF al despertar ayuda a conservar la energía.

Este código está optimizado para un nodo sensor en una aplicación IoT, centrándose en la gestión eficiente de la energía a través del sueño profundo y la transmisión de datos efectiva utilizando ESP-NOW.