

#include "nf_wifi.h"

Copia los archivos "nf_wifi.c" y "nf_wifi.h" en la carpeta "source" de tu proyecto. Puedes incluir la librería poniendo `#include "nf_wifi.h"` en tu código.

```
bool NF_WiFiConnectDefaultAp(void);
```

Intenta conectarte al punto de acceso por defecto, definido en la WFC de la consola. Devuelve verdadero o falso, dependiendo de si la conexión se ha realizado con éxito.

```
void NF_WiFiDisconnectAp(void);
```

Desconectate del punto de acceso y cierra el módulo WIFI de la consola.

```
bool NF_WIFI_CreateUdpSender( const char* adress,    // Direccion IP
                             u16 port             // Puerto
                             );
```

Crea un socket en modo "cliente" usando el protocolo UDP. Devuelve verdadero si se ha creado con éxito.

Ejemplo:

```
NF_WIFI_CreateUdpSender("192.168.1.201", 12345);
```

Crea un socket UDP para comunicarse con la dirección IP 192.168.1.201 usando el puerto nº12345.

```
bool NF_WIFI_CreateUdpListener(    u16 port        // Puerto
                                   );
```

Crea un socket en modo "servidor" en el puerto especificado, aceptando datos desde cualquier dirección IP. Devuelve verdadero si se ha creado con éxito.

Ejemplo:

```
NF_WIFI_CreateUdpListener(12345);
```

Crea un socket UDP en el puerto 12345 y déjalo a la escucha de cualquier dirección IP.

```
bool NF_WIFI_UdpSend(const char* data);
```

Envía una cadena de texto a través del socket UDP abierto. Los datos se almacenan también la variable global `NF_SEND_BUFFER` para poder realizar funciones de checksum.

Ejemplo:

```
NF_WIFI_UdpSend("Esto es una prueba");
```

Envía la cadena de texto "Esto es una prueba" a través del socket abierto.

```
s32 NF_WIFI_UdpListen(u32 timeout);
```

Espera datos en el socket abierto y en caso de recibirlos, almacenalos en la variable global `NF_RECV_BUFFER`. Tambien almacena el numero de bytes recibidos en la variable global `NF_BYTES_RECIEVED`. La funcion devuelve un valor `>0` en caso de recibir datos, si se produce un `TIMEOUT` (especificado en milisegundos X1000), devuelve un valor `<0`.

Ejemplo:

```
NF_WIFI_UdpListen(300000);
```

Deja el socket abierto a la escucha de datos con un timeout de 300 milisegundos.

Nota: Estas funciones son MUY BASICAS Y PUEDEN VARIAR. Dato que es mi primer intento de comunicaci3n en WIFI, quizas cambien mas adelante al ser optimizadas.
