

CC32xx WLAN Scan Policies Application

Overview

CC3200 device allows you to set scan policy and enables the scan. Using `sl_WlanPolicySet` API we can set scan policy to `SL_POLICY_CONNECTION`, `SL_POLICY_SCAN` or `SL_POLICY_PM`. Once you set the scan cycle and enables the same, we can retrieve the results of Scan using `sl_WlanGetNetworkList` API.

It gives you information related to RSSI (Received Signal Strength Indication), Security Information, BSSID and SSID information.

[Return to CC31xx & CC32xx Home Page](#)[Return to CC31xx Sample Applications](#)

Application details

The example intends to demonstrate how scan policy is set in the device. The procedure includes the following steps:

- Make sure the connection policy is not set (so no scan is run in the background)
- Enable scan, set scan cycle to 10 seconds and set scan policy. This starts the scan
- Get scan results - all 20 entries in one transaction
- Get scan results - 4 transactions of 5 entries
- Disable scan

Source Files briefly explained

- **main.c** - The main file that explains steps to set scan policy, enables scan and gets the scan result

Supporting Files

- **pinmux.c** - Generated by the PinMUX utility
- **startup_ccs.c** - CCS related functions
- **startup_ewarm.c** - IAR related functions
- **gpio_if.c** - GPIO Interface APIs

Usage

1. Run the reference application .
 - Open the Project in IAR or CCS. Build and download the application to the board
2. The application scans and gets the Network List.
 - Below screenshot shows you the number of Networks found and Network List.
 - **IRetVal** indicates the number of Networks found and **netEntries** contains the Network List.

The screenshot shows the CC32xx IDE with the `main.c` file open. The code implements a WLAN scan function that iterates through a list of networks and stores the results in the `netEntries` array. The Quick Watch window displays the contents of the `netEntries` array, showing five network entries with their respective SSIDs, security types, and other details.

```

return lRetVal;
}
MAP_UtilsDelay(8000000);
// get scan results - all 20 entries in one transaction
//
ucIndex = 0;
//
// retVal indicates the valid number of entries
// The scan results are occupied in netEntries[]
//
lRetVal = sl_WlanGetNetworkList(ucIndex, (unsigned char)WLAN_SCAN_COUNT,
                                &netEntries[ucIndex]);
if(lRetVal==0)
{
    GPIO_IF_LedOn(MCU_EXECUTE_FAIL_IND);
    UART_PRINT("Unable to retrieve the network list\n\r");
    return lRetVal;
}
/* put a break point here and check netEntries[] value for scan ssid list
//
// get scan results - 4 transactions of 5 entries
//
ucIndex = 0;
memset(netEntries, 0, sizeof(netEntries));

do
{
    lRetVal = sl_WlanGetNetworkList(ucIndex,
                                    (unsigned char)WLAN_SCAN_COUNT/4,
                                    &netEntries[ucIndex]);

    ucIndex += lRetVal;
}
while ((lRetVal == WLAN_SCAN_COUNT/4) && (ucIndex < WLAN_SCAN_COUNT));
/* put a break point here and check netEntries[] value for scan ssid list
//
// disable scan
//
ucpolicyOpt = SL_SCAN_POLICY(0);
lRetVal = sl_WlanPolicySet(SL_POLICY_SCAN, ucpolicyOpt, NULL, 0);
if(lRetVal != 0)
{
    GPIO_IF_LedOn(MCU_EXECUTE_FAIL_IND);
    UART_PRINT("Unable to Clear the Scan Policy\n\r");
    return lRetVal;
}

```

Expression	Value	Location	Type
netEntries	<array>	0x20008484	struct
[0]	<struct>	0x20008484	struct
ssid	"CC3200-Audio"	0x20008484	unsig
ssid...	'\f' (0x0C)	0x200084A4	unsig
sec...	'\0' (0x00)	0x200084A5	unsig
bssid	"E's# 1"	0x200084A6	unsig
rsi	'-' (0xAC)	0x200084AC	signed
rese...	"	0x200084AD	signed
[1]	<struct>	0x200084B0	struct
ssid	"NETGEAR-24G"	0x200084B0	unsig
ssid...	'\v' (0x0B)	0x200084D0	unsig
sec...	'\0' (0x00)	0x200084D1	unsig
bssid	"aFInk"	0x200084D2	unsig
rsi	'e' (0xBA)	0x200084D8	signed
rese...	"	0x200084D9	signed
[2]	<struct>	0x200084DC	struct
ssid	"stability_ap_1"	0x200084DC	unsig
ssid...	'.' (0x0E)	0x200084FC	unsig
sec...	'\0' (0x00)	0x200084FD	unsig
bssid	"4pi>p!"	0x200084FE	unsig
rsi	'!' (0xA6)	0x20008504	signed
rese...	"	0x20008505	signed
[3]	<struct>	0x20008508	struct
ssid	"cc3200demo"	0x20008508	unsig
ssid...	'\n' (0x0A)	0x20008528	unsig
sec...	'\0' (0x00)	0x20008529	unsig
bssid	"4piJ0D0"	0x2000852A	unsig
rsi	'0' (0xD8)	0x20008530	signed
rese...	"	0x20008531	signed
[4]	<struct>	0x20008534	struct
ssid	"TAG_FF"	0x20008534	unsig
ssid...	'.' (0x06)	0x20008554	unsig
sec...	'\0' (0x00)	0x20008555	unsig
bssid	"4pi=ö"	0x20008556	unsig
rsi	'¶' (0xB6)	0x2000855C	signed
rese...	"	0x2000855D	signed
[5]	<struct>	0x20008560	struct
ssid	"belkin_audio"	0x20008560	unsig
ssid...	'\f' (0x0C)	0x20008580	unsig
sec...	'\0' (0x00)	0x20008581	unsig

3. On the Board Led Orange will be on If some error occurs. On successful execution Led Green Will be on.

Limitations/Known Issues

None.

Article Sources and Contributors

CC32xx WLAN Scan Policies Application *Source:* <http://processors.wiki.ti.com/index.php?oldid=184780> *Contributors:* A0221015, Codycooke, Jitgupta, Malokyle

Image Sources, Licenses and Contributors

File:Cc31xx cc32xx return home.png *Source:* http://processors.wiki.ti.com/index.php?title=File:Cc31xx_cc32xx_return_home.png *License:* unknown *Contributors:* A0221015

File:Cc32xx return sample apps.png *Source:* http://processors.wiki.ti.com/index.php?title=File:Cc32xx_return_sample_apps.png *License:* unknown *Contributors:* A0221015

Image:CC32xx Scan Policy IAR expressionList 1.0.0.png *Source:* http://processors.wiki.ti.com/index.php?title=File:CC32xx_Scan_Policy_IAR_expressionList_1.0.0.png *License:* unknown *Contributors:* Jitgupta