**Please note that Cypress is an Infineon Technologies Company.**
The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

**Continuity of document content**
The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

**Continuity of ordering part numbers**
Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.

# THIS SPEC IS OBSOLETE

Spec No: 002-15438

Spec Title: AN215438 - ADJACENT CHANNEL INTERFERENCE DAEMON

Replaced by: NONE

AN215438

# Adjacent Channel Interference Daemon

**Associated Part Family: CYW43XX**

This document contains a description of the Adjacent Channel Interference (ACI) daemon of the Wireless LAN (WLAN) Media Ducati driver distribution (releases 5.24.20 and later).

It is currently intended for Cypress engineers, particularly field application engineers, who need an understanding of ACI daemon operation.



## Contents



# 1 About This Document

## 1.1 Cypress Part Numbering Scheme

Cypress is converting the acquired IoT part numbers from Broadcom to the Cypress part numbering scheme. Due to this conversion, there is no change in form, fit, or function as a result of offering the device with Cypress part number marking. The table provides Cypress ordering part number that matches an existing IoT part number.

Table 1. Mapping Table for Part Number between Broadcom and Cypress

| Broadcom Part Number | Cypress Part Number |
|---|---|
| BCM43XX | CYW43XX |

## 1.2 Acronyms and Abbreviations

In most cases, acronyms and abbreviations are defined upon first use. For a more complete list of acronyms and other terms used in Cypress documents, go to: http://www.cypress.com/glossary.

## 1.3 References

The references in this section may be used in conjunction with this document.

**Note:** Cypress provides customer access to technical documentation and software through its customer support portal (CSP). To access the CSP, go to http://www.cypress.com/support. For help accessing the CSP, contact your Sales or Engineering support representative.

| Document (or Item) Name | Item Number | Source |
|---|---|---|
| [1] Agreement Reached Regarding U.S. Position on 5 GHz Wireless Access Devices, United States Department of Commerce News | – | http://www.ntia.doc.gov/ntiahome/press/2003/5ghzagreement.htm |
| [2] IEEE 802.11h-2003 | – | http://standards.ieee.org/ |

# 2 IoT Resources

Cypress provides a wealth of data at http://www.cypress.com/internet-things-iot to help you to select the right IoT device for your design, and quickly and effectively integrate the device into your design. Cypress provides customer access to a wide range of information, including technical documentation, schematic diagrams, product bill of materials, PCB layout information, and software updates. Customers can acquire technical documentation and software from the Cypress Support Community website (https://community.cypress.com/)

# 3 ACI Daemon File Location

The Ducati driver distribution is contained within the linux_router_bom, Ducati release branch, in the ACI module under WLAN source control. The ACI daemon is in src/router/aci/wl_aci_linux.c.

# 4 Adjacent Channel Interference Daemon Conditional Invocation

The Adjacent Channel Interference (ACI) daemon is run at AP start-up depending on the value of the `aci_daemon` NVRAM variable (see aci_daemon on page 7). It may be run manually from the AP and station consoles by typing the `wl_aci` command. The advantage of using a console is that informational output on channel switches gets displayed.

On start-up, the ACI daemon is run automatically on the Access Point (AP) and stations if `aci_daemon` is set to “up”, which is the default setting. To determine whether the ACI daemon is enabled, issue the following command using the Linux® console:

```
nvram get aci_daemon
```

To have the ACI daemon run automatically every time the AP and stations are booted, issue the following commands using the linux console:

```
nvram set aci_daemon=”up”
nvram commit
reboot
```

Issue the following commands to disable the ACI daemon:

```
nvram set aci_daemon=”down”
nvram commit
reboot
```

**Note:** Although the ACI daemon can run on STAs, the STA ACI daemon doesn’t participate in ACI detection.

# 5 Randomized Initial Channel Selection

When there are multiple APs using the ACI algorithm within detection distance of each other, each AP may choose the same optimal channel. In this scenario, the ACI algorithm within each AP will detect the other APs as interference and try to switch channels. To minimize this at start-up, the ACI daemon algorithm randomizes its initial channel selection. The ACI daemon algorithm:

1. Initializes the AP.
2. Sleeps for a randomly selected duration between 0 and 10 seconds.

**Note:** The randomization ensures that multiple APs will come up at different times.

3. Does a complete scan of all included channels upon waking up.
4. Evaluates the scan results to determine if there is an optimal channel. From this evaluation:

- If the current channel is optimal, then the ACI daemon starts normal operation using the current channel (a channel change is not required).
- If a new channel is optimal, then the daemon switches to the new channel and starts normal operation.
- If neither the current channel nor a new channel is optimal and three scans have taken place since ACI daemon initialization, then the ACI daemon starts normal operation on the current channel.
- If neither the current channel nor a new channel is optimal and three scans have not taken place since ACI daemon initialization, then the ACI daemon sleeps for a randomly selected duration between 2 and 5 seconds before its next scan (go to Step 3).

# 6 Adjacent Channel Interference Detection

## 6.1 Active/Passive Scanning

After STA and AP startup, the ACI daemon:

1. Verifies that the driver is loaded.
2. Determines the current band and channel.

In addition to the above steps, the AP ACI daemon:

1. Determines if a station is associated.
2. Performs an active and passive scan of all the channels.
3. Scores every channel in the band and stores the score in a list.

If no station is associated with the AP, the ACI daemon repeats the above steps at three-second intervals until a station associates. If, during Step 1, it's determined that no station is associated, then the AP is free to scan all channels during Step 2.

When a station associates, the ACI daemon checks the score of the current channel and begins scanning individual channels at the ACI scan sleep interval. This interval is controlled by an NVRAM variable and can be changed by the user. The default interval is four seconds.

The AP will dwell on each channel being scanned for a few milliseconds. If an active scan is allowed, the AP can send out probe requests and get responses. If only a passive scan is allowed, then the AP needs to dwell for longer because listening to beacons is its only source of information. Channel dwell time is always in milliseconds.

The AP uses scan information to update channel scores.

The ACI algorithm will make the decision to change channels only if:

- The packet error rate (PER) in the current channel exceeds the PER threshold.
- Non-Wi-Fi Electromagnetic Interference (EMI) is detected on the current channel, that is, the current channel glitch counter exceeds the glitch count threshold.
- Another optimal channel is available. An optimal channel can be one that is free of detected interference and has no occupants in the target or adjacent channels. It can also be a channel with fewer current occupants, less adjacent channel interference, or a higher transmission power.

The AP will send a Channel Switch Announcement (CSA) to the stations associated with the AP before a channel change takes place.

## 6.2 Channel Scoring

The ACI daemon performs active scanning on the AP to detect Wi-Fi devices. The active scanning mechanism is a probe request issued on each channel in succession. This approach has the advantage that it can be done while streaming data without dropping packets. The limitations of this approach are:

- It can only detect other active Wi-Fi devices.
- The interference may not be from another Wi-Fi device transmitting or producing interference.
- In Dynamic Frequency Selection (DFS) channels, all scanning has to be passive, so the ACI daemon can only listen for beacons and not actively probe for Wi-Fi devices.
- A response to interference is not immediate.

The AP can also detect non-Wi-Fi EMI on the current channel; therefore, if non-Wi-Fi devices produce sufficient interference to adversely affect throughput, the AP will perform a channel switch.

The channel scoring algorithm uses scores of –2 for adjacent occupants and –1 for co-occupants in the 5G band. Additionally:

- Current channel occupants are counted.
- The occupants of channels within 4 and 8 channels from the current channel are counted.
- The current-channel score is computed using the scores of the current and adjacent channels.

After each scan, the current-channel score is compared with that of every other channel in the list.

# 7 Enhancements to ACI Scanning

Enhancements to ACI scanning include:

- Combing the results of multiple scans (referred to as adding damping) before declaring a channel optimal. A channel can be declared optimal after three active scans or 20 passive scans.
- Ensuring that excluded channels that are not adjacent to usable channels are not scanned to reduce the total scan time.
- Updating scan results from valid usable channels only to ensure that the daemon minimizes scan time.

# 8 Enhancements to the ACI Channel Selection Algorithm

Enhancements to the ACI channel selection algorithm include:

- Calculating channel occupancy using a rolling average to give a better estimate of how many other APs are using a particular channel.
- Using PER to help decide whether a channel change is required. This is in addition to the information gathered by ACI scanning. Channel changes are done only if the PER exceeds a PER threshold. This ensures that channel changes are done only if the channel quality is interfering with video traffic.
- Using a Receive Signal Strength Indication (RSSI) threshold to declare an interferer: –80 dBm for a cochannel interferer and –60 dBm for an adjacent channel interferer.
- Figuring out which station associated to the AP has the weakest RSSI to determine if a lower-power channel can be used.

# 9 ACI Robustness

To overcome severe interference or low receive signal strength that can cause client interference messages to be lost, and to minimize the time it takes to detect interference, the following two ACI daemon features have been implemented:

- Before each channel switch, a CSA is sent as an IEEE 802.11™ frame to all connected clients. In addition, a special data packet containing the channel switch announcement is sent to each connected client using the socket that the AP uses to receive interference information from each client. Doing this ensures that even if the client misses the CSA, it can still act on the notification received by the ACI daemon on its socket.
- Scan intervals are subsecond to reduce the time to recognize interference, and, thus, minimize video glitches when interference is detected.

# 10 Feature Settings

The ACI daemon incorporates configurable features, all of which can be set through the Linux console `nvram set` command and saved using the `nvram commit` command. In addition, the most common configurable features can be set from the Basic page of the browser-based utility that Cypress supplies with its AP software.

The NVRAM variable names and settings are described below.

## 10.1 aci_daemon

This variable may be set to “up” or “down”. If set to “up”, the ACI daemon will start upon bootup. The AP and station must use the same setting because the ACI daemon on the AP listens for interference information from associated stations.

The default setting is “down”.

## 10.2 aci_glitch_threshold

This variable can be set to decimal integer values from 200 to 5000. For example:

```
nvram set aci_glitch_threshold=900
nvram commit
```

The `aci_glitch_threshold` determines AP sensitivity to non-Wi-Fi EMI. When the current-channel glitch count exceeds this threshold, the AP will search for and switch to a clear channel if one is available.

The default is 2000.

## 10.3 aci_excluded_channels

This variable can be set to a comma-separated list of up to four integer channel numbers. If the variable is set to one or more valid 5G channel numbers, the ACI daemon will remove the channel or channels from the valid channel list. It will not consider the channel as a candidate when searching for a new channel, and, if the current channel is set to an excluded channel and a station is associated, it will force a channel switch.

The default is “140, 108, 38, 42, 116, and 124”.

## 10.4 aci_preferred_channels

This variable can be set to a comma-separated list of up to four integer channel numbers. If the variable is set to one or more valid 5G channel numbers, the ACI daemon will consider the channel or channels for a channel switch before searching the remainder of the channels in the band.

The default is “ ” (null).

## 10.5 aci_auto_channel

This variable can be set to “on” or “off”. If set to “on”, the AP will automatically switch channels when interference is detected. If set to “off”, the AP will continue to detect interference but will not switch channels.

The default value is “on”.

## 10.6 aci_info_prints

This variable can be set to “on” or “off”. If set to “on”, and the ACI daemon is invoked from the console, it will print information about channel switches.

The default value is “on”.

## 10.7 aci_debug_prints

This variable can be set to “on” or “off”. If set to “on”, and the ACI daemon is invoked from the console, it will print debug information about the channel map and detected interference.

The default value is “off”.

## 10.8 aci_scan_sleep_secs

This variable can be set to decimal integer values from 1 to 10. It determines the scan frequency in seconds when a station is associated with the AP. The default value is 1.

## 10.9 aci_def_ap_ipaddr

This variable can be set to an IPV4 IP address string in decimal dot notation. It must be set to the LAN address used by the AP, otherwise, the station will be unable to send interference information to the AP.

The default, which corresponds to the default AP LAN address, is “192.168.1.1”.

**Note:** Because the STA does not scan, it sends only current-channel information.

## 10.10 aci_pref_dfs

This variable can be set to “true” or “false”. If set to “true”, the AP will search the 5 GHz DFS channels for a clear channel before searching other channels in the band.

The default is “false”.

## 10.11 aci_exclude_dfs

This variable can be set to “true” or “false”. If set to “true”, then the AP will avoid using 5 GHz DFS channels.

The default value is “false”.

## 10.12 aci_dfs_scan_type

This variable may be set to “passive” or “active”. If set to “passive”, then the AP and clients will perform passive (listening) scans on DFS channels. If set to “active”, then the AP and clients will perform active scans by sending broadcast probes on DFS channels.

## 10.13 aci_reuse_dfs

This variable may be set to “true” or “false”, the default being “false”. If set to “false”, then the ACI algorithm will not choose a DFS channel except at initial channel selection. If set to “true”, the ACI algorithm will consider using a DFS channel, even after initial channel selection.

# 11 Dynamic Frequency Selection Policies

This section contains a description of DFS in IEEE 802.11 and the current approach to streaming data wirelessly in the 5 GHz band while meeting DFS channel usage requirements.

## 11.1 DFS Channel Definition and Rules

DFS channels, codified in IEEE 802.11h-2003, are channels mandated by the FCC in the United States in the 5250−5725 MHz U-NII mid-band for radar avoidance. The standard requires that special rules be observed by IEEE 802.11 devices in order to receive FCC certification. The rules as they apply to ACI are:

- Upon entering a DFS channel, the AP must listen for radar signals on the channel for one minute before transmitting.
- If the device is transmitting on a DFS channel, it must detect radar transmissions within 250 milliseconds of their initiation and immediately leave the channel.

## 11.2 Streaming Applications over DFS Channels

Streaming applications require continuous, uninterrupted access to the medium. Effective ACI mitigation requires unfettered access to all of the available channels to assure that the best possible channel is used. The DFS rules are obviously incompatible with the requirements of streaming applications with ACI mitigation.

## 11.3 Current ACI Implementation in DFS Channels

The following implementation details apply to and in DFS channels:

- All scanning is passive.
- The ACI algorithm will select a DFS channel only at start-up so that the 60 second quiet-time requirement will not disrupt operation.
- After start-up, the default behavior is to eliminate DFS channels from the usable channel list. This ensures that the ACI daemon will not select a DFS channel after start-up. DFS channels that are not adjacent channels of usable channels will not be scanned after start-up.
- The `aci_reuse_dfs` NVRAM variable provides the option of using DFS channels after initial channel selection. The default is set to “false” so that no DFS channel is selected after initial channel selection.

## 11.4 Other Approaches for Handling DFS Channels

This section contains some possible approaches to wireless data streaming with ACI mitigation. The approaches are presented in order of increasing complexity and/or cost.

### 11.4.1 Avoid DFS Channels

The simplest and lowest-cost approach is to simply exclude DFS channels from consideration when attempting to mitigate ACI by switching channels. Unfortunately, excluding the DFS channels effectively excludes most usable channels because of the low transmission power limits of channels 36 through 48. This approach is not workable for a practical product.

### 11.4.2 Choose a Start-up DFS Channel but Exclude DFS Channels for ACI Mitigation

The simplest workable approach is to choose a DFS channel at start-up, but exclude DFS channels from ACI mitigation if interference is detected. At AP start-up, the one-minute start-up delay is tolerable because, without any associated STAs, there won’t be stream interruptions. However, if interference is detected after streaming begins, then the ACI daemon, in order to avoid the one-minute delay incurred when switching to a DFS channel, must switch to a better non-DFS channel if one is available. This approach could work provided the *start-up* condition and a low-impact policy for returning to a DFS channel are defined.

### 11.4.3 Choose a Start-up DFS Channel but Exclude Quiet DFS Channels for ACI Mitigation

A slightly more complex approach is to choose a DFS channel at startup, but exclude *quiet* DFS channels from ACI mitigation if interference is detected. In a fully IEEE 802.11h-compliant environment, if there is a Wi-Fi device transmitting on a DFS channel, then the device paid the one-minute penalty and detected no radar signals in the channel. In this case, it is acceptable to switch to the DFS channel, assuming that it has the capacity to support the current streaming requirements and that the channel is monitored for future radar signals.

This approach is risky in that it may not pass regulatory muster. It requires that DFS channels receive different treatment than non-DFS channels. It also requires the ACI daemon to calculate channel capacity and stream throughput, which is fairly challenging with two or more clients in the presence of noise and other Wi-Fi devices on the channel.

### 11.4.4 Continuously Scan Using an Idle Client

A relatively complex approach would be to use any idle client to scan one or more DFS channels continuously and report interference and radar detection information to the AP. In the event that no client is available, DFS channels would be excluded for ACI mitigation. However, if recent DFS radar detection information is available, the AP could switch to any of the available DFS channels in the event that interference is detected. This approach is low cost but highly complex due to the ACI daemon management requirements.

### 11.4.5 Continuously Scan Using a Dedicated Client or Scanner

A relatively high cost, but simpler approach would be to use a dedicated client, or add-on radar detector to scan one or more DFS channels continuously and report interference and radar detection information to the AP. In this case, recent DFS radar detection information would always be available, so the AP could switch to any available DFS channel if interference is detected. This approach carries the higher cost of a dedicated receiver, but relieves the ACI daemon of much of the management complexity of using an idle client.

# Document History Page

| Document Title: AN215438 - Adjacent Channel Interference Daemon<br>Document Number: 002-15438 | | | | |
|---|---|---|---|---|
| **Rev.** | **ECN No.** | **Orig. of Change** | **Submission Date** | **Description of Change** |
| ** | - | - | 11/16/2010 | Initial release |
| *A | 5445106 | UTSV | 09/28/2016 | Updated to Cypress format. |
| *B | 5883842 | AESATMP8 | 09/14/2017 | Updated logo and Copyright. |
| *C | 6382981 | KEMA | 11/13/2018 | Obsoleting the spec. |

# Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

### Products

| | |
|---|---|
| Arm® Cortex® Microcontrollers | cypress.com/arm |
| Automotive | cypress.com/automotive |
| Clocks & Buffers | cypress.com/clocks |
| Interface | cypress.com/interface |
| Internet of Things | cypress.com/iot |
| Memory | cypress.com/memory |
| Microcontrollers | cypress.com/mcu |
| PSoC | cypress.com/psoc |
| Power Management ICs | cypress.com/pmic |
| Touch Sensing | cypress.com/touch |
| USB Controllers | cypress.com/usb |
| Wireless Connectivity | cypress.com/wireless |

### PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6 MCU

### Cypress Developer Community

Community | Projects | Video | Blogs | Training | Components

### Technical Support

cypress.com/support





Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709