

# VIC-2DID 구성 및 문제 해결

## 목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[표기 규칙](#)

[배경 정보](#)

[통화 시작 신호](#)

[즉시 시작](#)

[윙크 시작](#)

[다이얼 지연](#)

[통화 감독](#)

[응답 감독](#)

[감독 연결 끊기](#)

[실패한 통화 처리](#)

[직접 안쪽으로 다이얼 구성](#)

[문제 해결](#)

[관련 정보](#)

## [소개](#)

이 문서에서는 FXS(Foreign Exchange Station)-2DID 카드의 기본 컨피그레이션을 구현하는 방법 및 시그널링 사용 방법에 대해 자세히 설명합니다. 하드웨어 및 Cisco IOS® 소프트웨어 지원에 대한 자세한 내용은 [2 Port Direct Inward Dial \(2 DID\) Voice Interface Card 이해](#) 를 참조하십시오.

## [사전 요구 사항](#)

### [요구 사항](#)

이 문서에 대한 특정 요건이 없습니다.

### [사용되는 구성 요소](#)

이 문서의 정보는 다음 소프트웨어 및 하드웨어 버전을 기반으로 합니다.

- Cisco VG200 게이트웨이
- Cisco IOS Software 릴리스 12.2(8)T

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우, 모든

명령어의 잠재적인 영향을 미리 숙지하시기 바랍니다.

## 표기 규칙

문서 규칙에 대한 자세한 내용은 [Cisco 기술 팀 표기 규칙](#)을 참조하십시오.

## 배경 정보

DID(Direct Inward Dialing)는 전화 회사가 제공하는 서비스로서 전화 교환원이나 자동 통화 교환 없이 발신자가 PBX(Private Branch Exchange) 또는 패킷 음성 시스템(예: Cisco CallManager 및 IOS 라우터/게이트웨이)의 내선 번호로 직접 전화를 걸 수 있습니다. 이 서비스는 전화 번호의 마지막 3~5자리만 PBX 또는 라우터/게이트웨이로 전달하는 DID 트렁크를 사용합니다. 예를 들어, 한 회사가 전화 내선 번호 555-1000~555-1999를 가지고 있고 발신자가 555-1234로 전화를 걸면, 로컬 CO(Central Office)는 234를 PBX 또는 패킷 음성 시스템으로 전달합니다. PBX 또는 패킷 음성 시스템은 234로 확장됩니다. 또한 DID 트렁크는 최대 50개의 내선 번호를 제공할 수 있으므로 비교적 작은 트렁크를 사용하여 많은 수의 내선 번호를 제공할 수 있습니다.

## 통화 시작 신호

DID는 숫자를 전송하기 전에 DID 트렁크와 PBX 간에 핸드셰이킹 프로토콜을 요구합니다. 신호 처리는 윙크 시작, 다이얼 지연 또는 즉시 시작입니다. 이러한 신호 유형은 Ear and Moth(E&M) 시그널링에 사용되는 것과 유사합니다.

## 즉시 시작

즉각적인 시작은 가장 간단한 프로토콜입니다. 시작 끝은 오프후크를 이동하여 회선을 고정하고 응답을 기다리지 않고 숫자를 풀기 시작합니다. 표준에 따르면, 즉시 시작에 사용되는 주소 신호 처리는 다이얼 펄싱입니다.

## 윙크 시작

윙크 시작 시 발신 측에서 오프후크를 시작하고 시작하기 전에 아웃포싱이 다른 쪽에서 확인 응답을 기다리는 방식으로 회선을 고정합니다. 확인은 140ms에서 290ms의 극성(오프 후크)의 반대로, 윙크라고도 합니다. 수신 발작 신호를 수신한 후 윙크는 100ms 이내에 발생해야 합니다. 시그널링 기능 외에도 윙크 시작은 제대로 작동하지 않는 트렁크를 식별하고 네트워크에서 발신자에게 재주문음을 보낼 수 있도록 하는 무결성 검사 역할을 합니다.

## 다이얼 지연

지연 다이얼 모드에서 원래 측은 회선을 분리(후크 해제)하고 약 200ms를 기다린 다음 원거리의 후크(배터리 정상)가 있는지 확인합니다. 이 경우 다이얼 숫자가 출력됩니다. 원끝이 오프후크(배터리 반대)인 경우, 원후크(배터리 정상)가 될 때까지 기다립니다. 그러면 숫자가 다이얼됩니다.

## 통화 감독

### 응답 감독

대답 감리는 배터리 교체 신호입니다. 응답 감리는 통화가 스테이션에서 응답하거나 녹음된 알림 또

는 IVR(Interactive Voice Response)로 라우팅될 때 반환됩니다. 응답 감리를 반환하지 않는 조건은 통화 내선 번호가 사용 중이거나 번호를 사용할 수 없는 경우입니다.

**참고:** Telco 장비에서 응답 감시가 올바르게 구성되지 않은 경우 단방향 음성이 발생할 수 있습니다

## 감독 연결 끊기

발신된 스테이션이 원래 종료 전에 연결이 끊기면 배터리가 정상으로 전환되고 수신 끝에서 오프후크를 기다립니다. 트렁크 연결 해제 후 호출된 스테이션의 연결이 끊기지 않으면 DID 포트에서 정상 배터리 전압이 복원되고 트렁크와 스테이션 간의 연결이 끊깁니다.

## 실패한 통화 처리

할당되지 않은 번호 또는 영구적으로 제한된 스테이션에 대한 통화가 완료되면 발신자가 다시 걸기 신호음을 수신합니다. 호출된 사용자가 오프후크 상태이면 발신자가 통화 중 신호음을 수신합니다.

## 직접 안쪽으로 다이얼 구성

이러한 DID 관련 CLI(Command-line Interface) 명령은 음성 포트 하드웨어가 DID 기능을 지원하는 경우에만 유효합니다. 또한 DID와 E&M 포트의 기능이 유사하므로 E&M 음성 포트에 사용되는 모든 CLI 명령을 DID 포트에서도 사용할 수 있습니다. 기본 운영 모드는 DID입니다.

FXS-DID 음성 포트를 구성하려면

```
voice-port x/y
signal did wink-start
delay-start
immediate-start
did-digit-length <Digit length>
```

FXS-DID를 비활성화하여 일반 FXS가 되려면

```
voice-port x/y
no signal did
```

**참고:** DID(구성된) 음성 포트에서는 발신 전화를 걸 수 없습니다. 하드웨어가 이러한 포트에서 벨소리 발신 및 발신 통화를 지원할 수 있지만 소프트웨어를 사용하여 이 포트를 사용할 수 없습니다.

## 문제 해결

DID 라인의 특성을 Telco로 확인하는 것이 중요합니다. DID 카드의 컨피그레이션은 이 확인된 정보를 기반으로 합니다. 가장 일반적인 문제는 다이얼 플랜의 잘못된 구성(CO에서 전달되는 숫자 수), 잘못된 통화 신호 및 회선 극성(Telco 장비는 극성에 민감하므로 팁과 링 연결을 전환해야 할 수 있음)입니다.

이는 숫자 4609를 호출하는 윙크 DID 트렁크의 debug vpm all 명령의 출력입니다.

```
*Mar 5 00:54:22.783: http_dsp_message: SEND/RESP_SIG_STATUS: state=0xC
timestamp=13671 systime=34886280
```

\*Mar 5 00:54:22.783: htsp\_process\_event: [1/0/0, DID\_ONHOOK, E\_DSP\_SIG\_1100]  
did\_onhook\_offhook htsp\_setup\_ind

\*Mar 5 00:54:22.787: [1/0/0] get\_local\_station\_id calling num= calling name=  
calling time=00/00 00:00

\*Mar 5 00:54:22.791: dsp\_digit\_collect\_on: [1/0/0] packet\_len=20 channel\_id=128  
packet\_id=35 min\_inter\_delay=240 max\_inter\_delay=9760  
min\_make\_time=10 max\_make\_time=100 min\_brake\_time=10  
max\_brake\_time=100

\*Mar 5 00:54:22.791: dsp\_soutput: [1/0/0]

\*Mar 5 00:54:22.795: dsp\_digit\_collect\_on: [1/0/0] packet\_len=20 channel\_id=128  
packet\_id=35 min\_inter\_delay=240 max\_inter\_delay=9760  
min\_make\_time=10 max\_make\_time=100 min\_brake\_time=10  
max\_brake\_time=100

\*Mar 5 00:54:22.795: dsp\_soutput: [1/0/0]

\*Mar 5 00:54:22.795: dsp\_digit\_collect\_on: [1/0/0] packet\_len=20 channel\_id=128  
packet\_id=35 min\_inter\_delay=240 max\_inter\_delay=9760  
min\_make\_time=10 max\_make\_time=100 min\_brake\_time=10  
max\_brake\_time=100

\*Mar 5 00:54:22.795: dsp\_soutput: [1/0/0]

\*Mar 5 00:54:22.795: htsp\_process\_event: [1/0/0, DID\_WAIT\_SETUP\_ACK,  
E\_HTSP\_SETUP\_ACK]did\_wait\_setup\_ack\_get\_ack

\*Mar 5 00:54:22.795: dsp\_digit\_collect\_off: [1/0/0] packet\_len=8 channel\_id=128  
packet\_id=36

\*Mar 5 00:54:22.795: dsp\_soutput: [1/0/0]

\*Mar 5 00:54:22.799: htsp\_timer2 - 88 msec

\*Mar 5 00:54:22.799: htsp\_dsp\_message: SEND/RESP\_SIG\_STATUS: state=0xC  
timestamp=13685 systime=34886282

\*Mar 5 00:54:22.799: htsp\_process\_event: [1/0/0, DID\_WAIT\_SETUP\_ACK,  
E\_DSP\_SIG\_1100]did\_wait\_setup\_ack\_offhook

\*Mar 5 00:54:22.799: did\_stop\_timer

\*Mar 5 00:54:22.799: htsp\_timer\_stop

\*Mar 5 00:54:22.887: htsp\_process\_event: [1/0/0, DID\_WAIT\_SETUP\_ACK,  
E\_HTSP\_EVENT\_TIMER2] did\_wait\_prewink\_timer

\*Mar 5 00:54:22.887: dsp\_digit\_collect\_off: [1/0/0] packet\_len=8 channel\_id=128  
packet\_id=36

\*Mar 5 00:54:22.887: dsp\_soutput: [1/0/0] did\_offhook

\*Mar 5 00:54:22.887: [1/0/0] set signal state = 0x6 timestamp = 0

\*Mar 5 00:54:22.887: dsp\_set\_sig\_state: [1/0/0] packet\_len=12 channel\_id=128  
packet\_id=39 state=0x6 timestamp=0x0

\*Mar 5 00:54:22.887: dsp\_soutput: [1/0/0] did\_onhook

\*Mar 5 00:54:22.887: [1/0/0] set signal state = 0x4 timestamp = 200

\*Mar 5 00:54:22.887: dsp\_set\_sig\_state: [1/0/0] packet\_len=12 channel\_id=128  
packet\_id=39 state=0x4 timestamp=0xC8

\*Mar 5 00:54:22.887: dsp\_soutput: [1/0/0]

\*Mar 5 00:54:22.891: dsp\_digit\_collect\_on: [1/0/0] packet\_len=20 channel\_id=128  
packet\_id=35 min\_inter\_delay=240 max\_inter\_delay=9760  
min\_make\_time=10 max\_make\_time=100 min\_brake\_time=10  
max\_brake\_time=100

\*Mar 5 00:54:22.891: dsp\_soutput: [1/0/0]

\*Mar 5 00:54:23.879: htsp\_digit\_ready(1/0/0): digit = 4

\*Mar 5 00:54:24.983: htsp\_digit\_ready(1/0/0): digit = 6

\*Mar 5 00:54:26.483: htsp\_digit\_ready(1/0/0): digit = 0

\*Mar 5 00:54:27.891: htsp\_digit\_ready(1/0/0): digit = 9

\*Mar 5 00:54:27.891: dsp\_digit\_collect\_off: [1/0/0] packet\_len=8 channel\_id=128  
packet\_id=36

\*Mar 5 00:54:27.891: dsp\_soutput: [1/0/0]

\*Mar 5 00:54:27.895: htsp\_dsp\_message: SEND/RESP\_SIG\_STATUS: state=0xC  
timestamp=18781 systime=34886792

\*Mar 5 00:54:27.895: htsp\_process\_event: [1/0/0, DID\_OFFHOOK, E\_HTSP\_PROCEEDING]

\*Mar 5 00:54:27.895: htsp\_process\_event: [1/0/0, DID\_OFFHOOK, E\_DSP\_SIG\_1100]  
did\_offhook\_offhook

\*Mar 5 00:54:27.895: did\_stop\_timer

\*Mar 5 00:54:27.895: htsp\_timer\_stop wrong offhook eventhtsp\_alert\_notify

\*Mar 5 00:54:32.415: htsp\_process\_event: [1/0/0, DID\_OFFHOOK,

```
E_HTSP_VOICE_CUT_THROUGH] htsp_connect: no_offhook 0
*Mar 5 00:54:32.419: htsp_process_event: [1/0/0, DID_OFFHOOK, E_HTSP_CONNECT]
    did_offhook_connect
*Mar 5 00:54:32.419: htsp_timer2 - 40 msec did_offhook
*Mar 5 00:54:32.419: [1/0/0] set signal state = 0x6 timestamp = 250
*Mar 5 00:54:32.419: dsp_set_sig_state: [1/0/0] packet_len=12 channel_id=128
    packet_id=39 state=0x6 timestamp=0xFA
*Mar 5 00:54:32.419: dsp_soutput: [1/0/0]
*Mar 5 00:54:32.459: htsp_process_event: [1/0/0, DID_CONNECT_MIN,
    E_HTSP_EVENT_TIMER2]
*Mar 5 00:55:01.659: htsp_dsp_message: SEND/RESP_SIG_STATUS: state=0x4
    timestamp=52547 systime=34890168
*Mar 5 00:55:01.659: htsp_process_event: [1/0/0, DID_CONNECT,
    E_DSP_SIG_0100]did_offhook_onhook
```

## 관련 정보

- [음성 기술 지원](#)
- [음성 및 통합 커뮤니케이션 제품 지원](#)
- [Cisco IP 텔레포니 문제 해결](#)
- [Technical Support - Cisco Systems](#)