

# Anleitung zum benutzerdefinierten Kabelaufbau für Cisco IP-Telefonkonsolen

## Inhalt

[Einführung](#)

[Hintergrundinformationen](#)

[Arten von Konsolenports](#)

[Herstellung des Konsolenkabels](#)

[USB-Konsolen-Port-Kabel](#)

[Sechs-polige aux-Konsolen-Port-Kabel](#)

[Schritte zur Verwendung des Konsolenport-Kabels](#)

[Ähnliche Diskussionen in der Cisco Support Community](#)

## Einführung

In diesen Dokumenten wird beschrieben, wie Sie ein Cisco IP-Telefon-Konsolenkabel erstellen, das zum Erfassen von Telefonprotokollen direkt vom Konsolenport verwendet wird.

## Hintergrundinformationen

Die Telefon-Debug-Protokolle werden mit drei verschiedenen Methoden gesammelt:

- Konsolenprotokolle für den Webzugriff
- Debug-Modus durch Secure Shell (SSH)-Zugriff
- Konsolen-Port des Telefons

Verwenden Sie nur die Methoden 1 und 2, wenn das Telefon über eine IP-Verbindung erreichbar ist. Gelegentlich ist aufgrund der Netzwerksicherheit, z. B. Firewall-Richtlinien, kein Webzugriff oder SSH-Zugriff verfügbar.

Methode 3 ist bei Problembehebungsszenarien nützlich, in denen Protokolle während des Telefonstarts oder vor dem Erreichbaren der Telefon-IP erfasst werden müssen, z. B.:

- Fehlerbehebung bei DHCP-Problemen des Telefons
- VPN-Problem
- Telefon-Erreichbarkeit oder Sicherheitsprobleme

In diesen Situationen wird ein Konsolenkabel verwendet.

## Arten von Konsolenports

Auf Cisco Telefonen stehen je nach Modell zwei Arten von Konsolenports zur Verfügung.

- USB-Konsolen-Port an Telefonmodellen 99XX und 8961

- Ein 6-poliger AUX-Port auf Telefonen der Modelle 79XX und 78XX

**Hinweis:** Das Telefonmodell 88XX verfügt über USB- und AUX-Ports.

## **Herstellung des Konsolenkabels**

Es gibt zwei Arten von Konsolenkabeln:

1. USB-Konsolen-Port-Kabel
2. Sechs-polige aux-Port-Kabel

### **USB-Konsolen-Port-Kabel**

Um ein USB-Konsolen-Port-Kabel zu bilden, benötigen Sie Folgendes:

- Zwei serielle USB-Kabel (DB9), wie in diesem Bild gezeigt:



- Ein serielles Kabel mit Buchse-Port DB9 auf jeder Seite, wie in diesem Bild gezeigt:



Die seriellen Ports der drei Kabel müssen miteinander verbunden werden, um das Kabel abzuschließen, wie in dieser Abbildung gezeigt:



Wenn das USB-Konsolen-Port-Kabel an das Telefon angeschlossen ist, wird auf dem Telefon eine Flash-Meldung angezeigt: Serielles USB-Konsolenkabel erkannt.

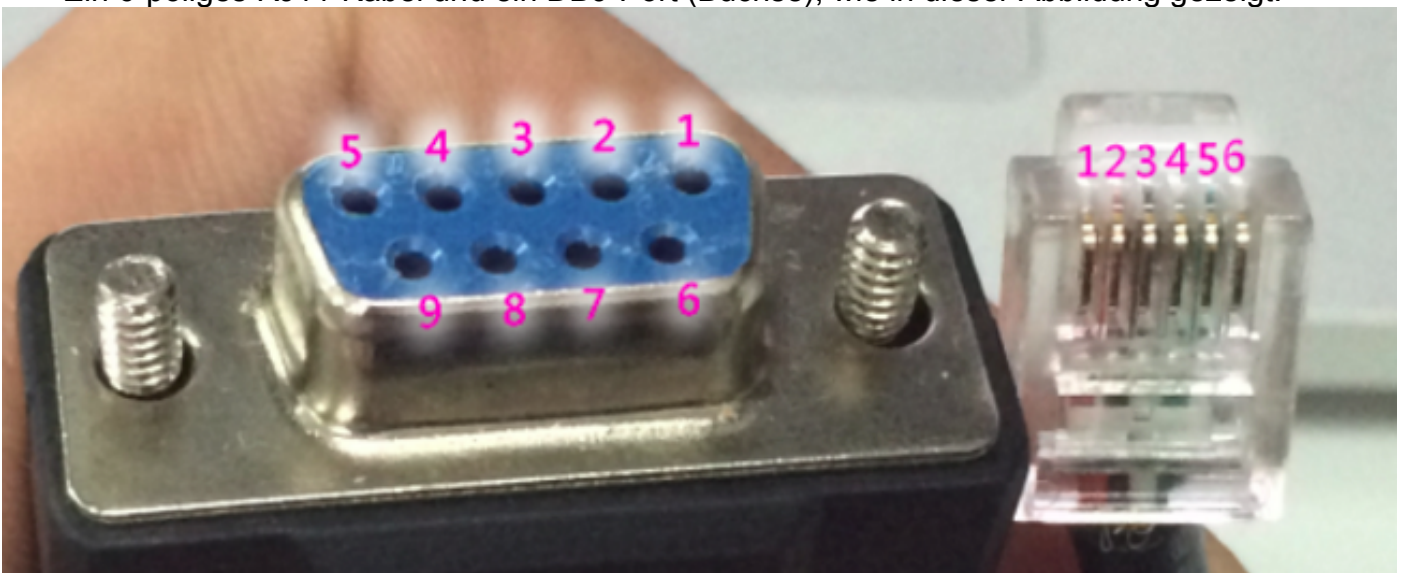
### **Sechs-polige aux-Konsolen-Port-Kabel**

Um ein 6-poliges Aux-Port-Kabel benötigen Sie:

- Ein serielles USB-Kabel (DB9), wie in diesem Bild gezeigt:



- Ein 6-poliges RJ11-Kabel und ein DB9-Port (Buchse), wie in dieser Abbildung gezeigt:



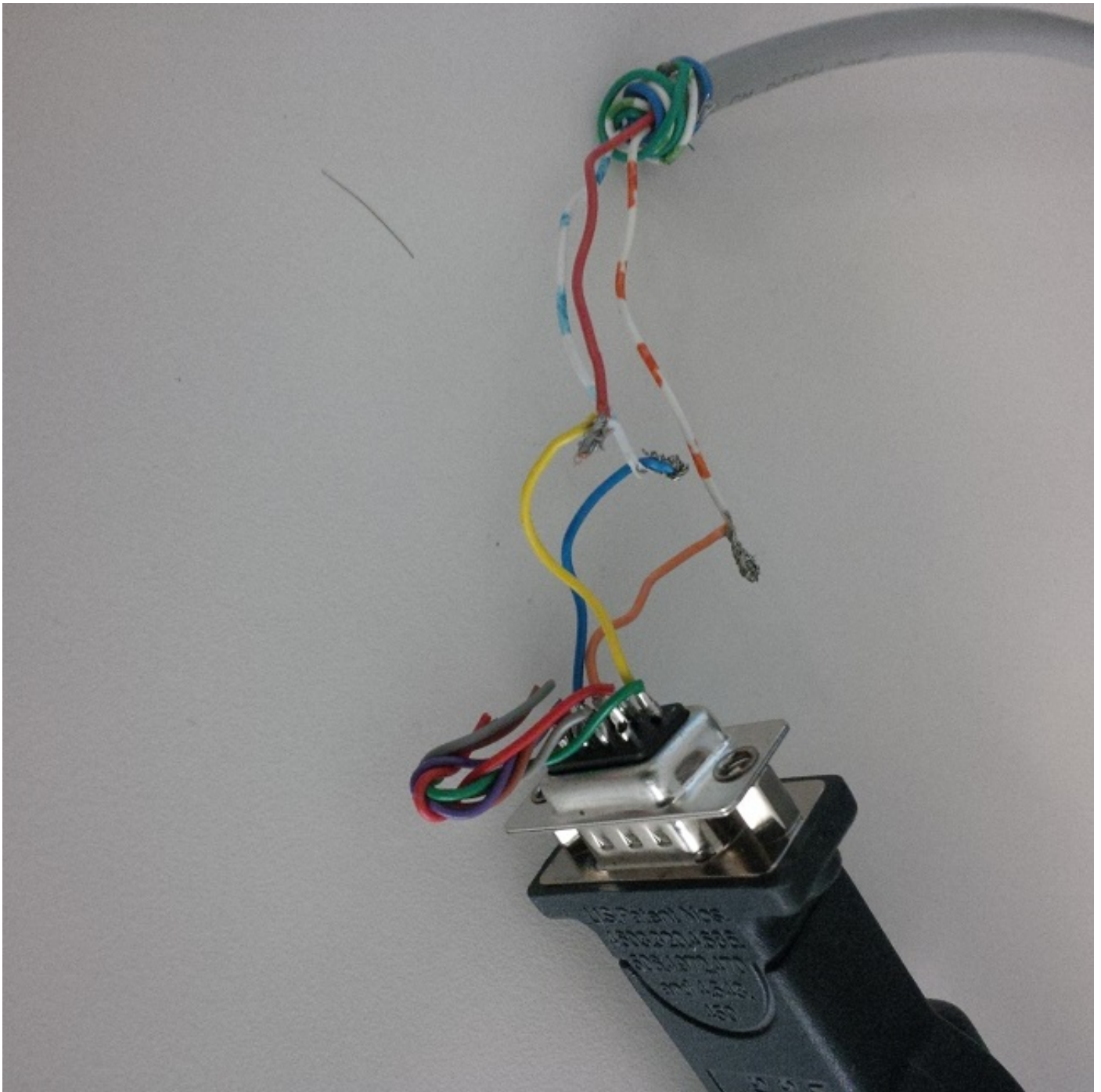
Auf dem DB9-Port werden die Zahlen für jeden Port genau angegeben. Für den RJ11-Port wird die Kabelnummer von links gezählt, während der Clip nach unten zeigt.

Drehen Sie die Kabel des RJ11 und des DB9 mit:

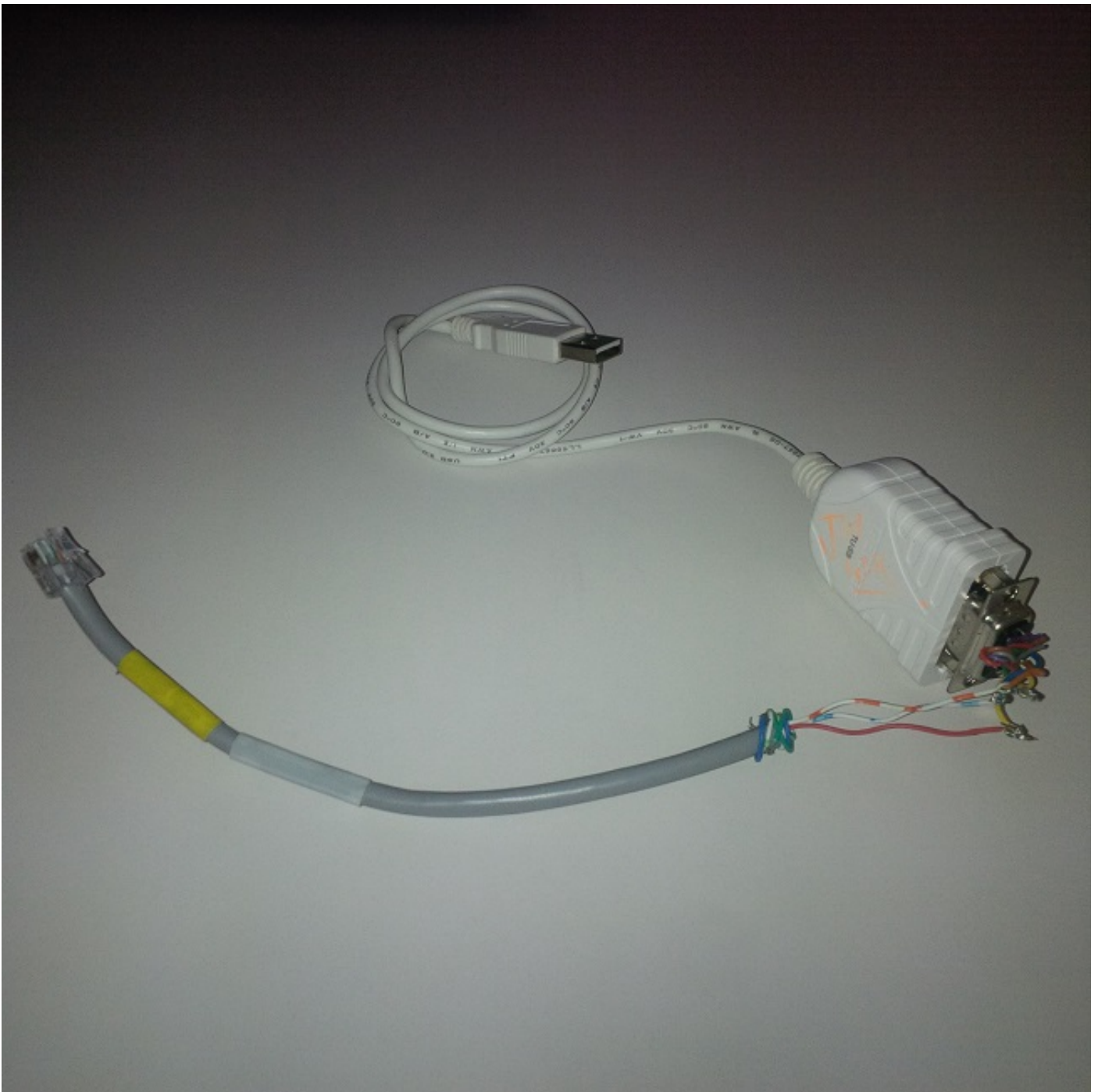
**DB9 RJ11**

2	2
1	4
5	1

Das Bild zeigt den RJ-11-Port und den DB9-Buchsenport, der zusammen gepatcht wurde.



In diesem Bild wird ein 6-poliges Aux-Port-Kontrollkabel angezeigt.

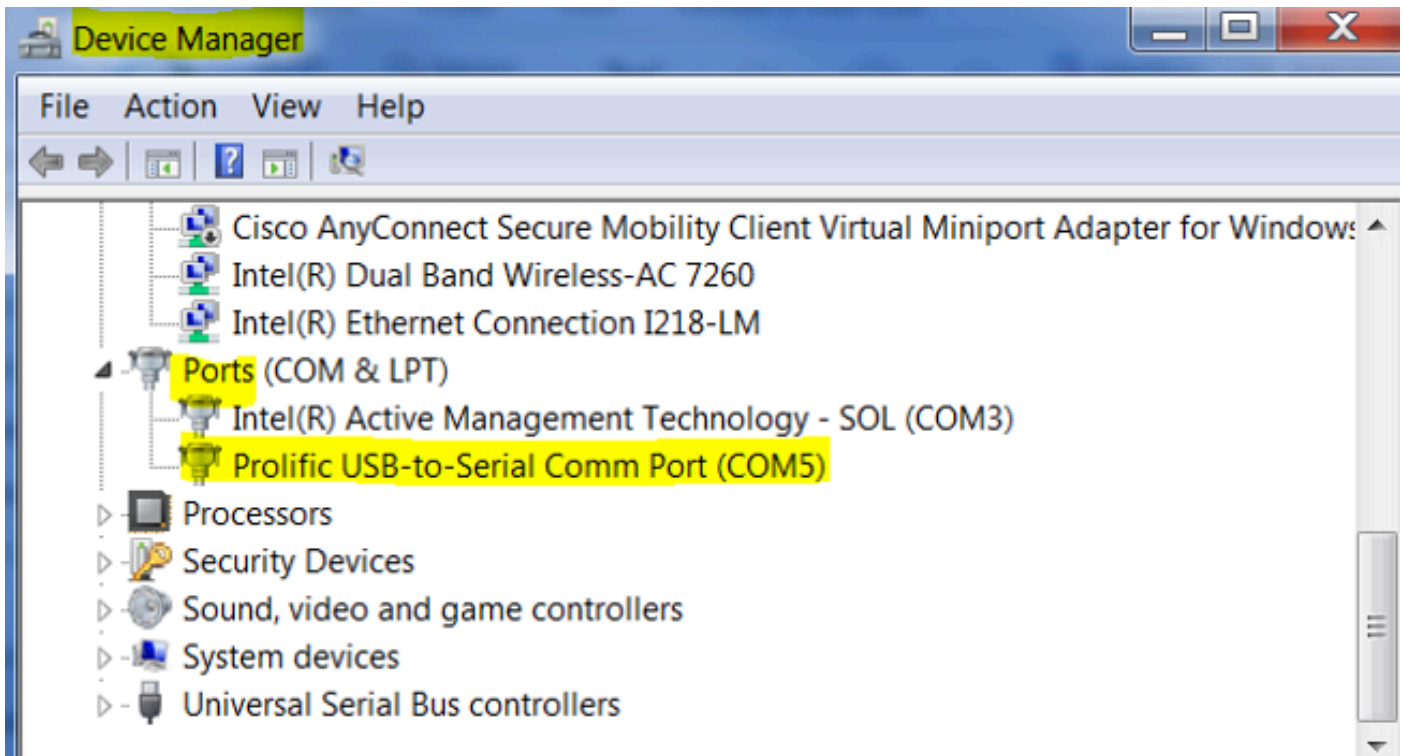


## Schritte zur Verwendung des Konsolenport-Kabels

1. Schließen Sie das entsprechende Konsolenkabel an den Konsolenport AUX oder USB an, je nach Telefonmodell.
2. Stellen Sie bei einem USB-Konsolen-Port sicher, dass der USB-Port auf dem Telefon von der Geräteseite des Telefons im Call Manager aktiviert ist.
3. Nachdem Sie das Kabel an den Computer angeschlossen haben, ermitteln Sie die COM-Anschlussnummer für diese Verbindung, indem Sie zum Geräte-Manager navigieren.

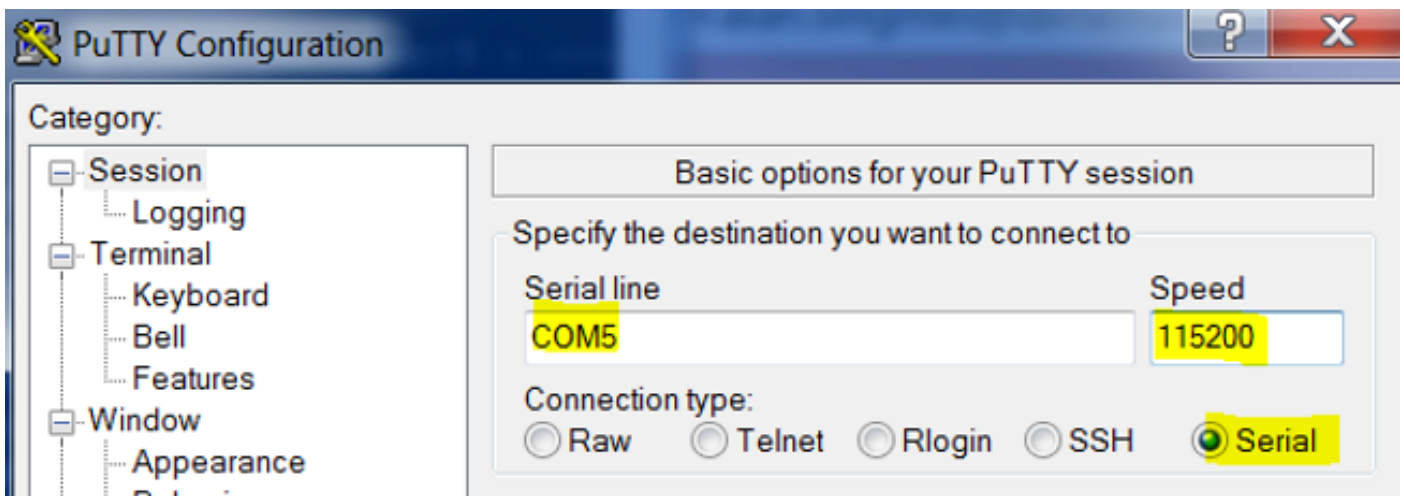
**Hinweis:** Wenn das Kabel im Geräte-Manager nicht erkannt wird, stellen Sie sicher, dass die Treiber (falls vorhanden) des seriellen USB-Kabels installiert sind.





4. Greifen Sie auf dem Windows-Rechner über die Puy-Sitzung mit Baud Rate 115200 auf den seriellen Port zu.

5. Um ein bestimmtes Debuggen zu aktivieren, konfigurieren Sie es zunächst über den Telefon-Debugmodus, indem Sie einen SSH-Zugriff verwenden.



Dieses Bild zeigt die Debug-Anmeldung auf dem Konsolenbildschirm.

```
2994: WRN 18:05:04.624182 JVM: Startup Module Loader|cip.mmgr.dt:? - [MediaMgrSM]:
  Unhandled Event, State = StateHandsetOffHook Event = EventSetSpeakerMode
2995: NOT 18:05:04.664184 JVM: Startup Module Loader|cip.cfg.ConfigManager:? - ---
>ConfigManager PropertyChanged: device.callagent.callcount
2996: NOT 18:05:04.665787 JVM: Startup Module Loader|cip.cfg.ConfigManager:? - <--
-ConfigManager PropertyChanged: device.callagent.callcount
2997: NOT 18:05:04.667458 JVM: Startup Module Loader|cip.cfg.ConfigManager:? - ---
>ConfigManager PropertyChanged: device.callagent.callcount
2998: NOT 18:05:04.669038 JVM: Startup Module Loader|cip.cfg.ConfigManager:? - <--
-ConfigManager PropertyChanged: device.callagent.callcount
2999: WRN 18:05:06.358267 JVM: Startup Module Loader|cip.mmgr.dt:? - [MediaMgrSM]:
  Unhandled Event, State = StateOnHook Event = EventSetSpeakerModeOff
3000: WRN 18:05:06.359889 JVM: Startup Module Loader|cip.mmgr.dt:? - [MediaMgrSM]:
  Unhandled Event, State = StateOnHook Event = EventEndcall
3001: NOT 18:05:06.399467 JVM: Startup Module Loader|cip.cfg.ConfigManager:? - ---
>ConfigManager PropertyChanged: device.callagent.callcount
3002: NOT 18:05:06.401203 JVM: Startup Module Loader|cip.cfg.ConfigManager:? - <--
-ConfigManager PropertyChanged: device.callagent.callcount
```