



Scaling OCR and ADB Devices: Why Most PCs Fail - and How PaytestCore Solves It

As automated payment testing environments grow more complex, teams often encounter surprising stability issues. Camera streams, used for OCR, begin to flicker or break the stream completely, ADB devices disconnect without warning, and entire environments behave inconsistently despite being configured correctly. Although these symptoms can appear to be software problems, the root cause is almost always much simpler and far more fundamental: the PC is running out of USB bandwidth. In multi-environment setups where cameras, Android devices – essential for terminal test automation and running regressing tests for payments - run in parallel, most standard computers simply cannot sustain the required data throughput.

In this blog, we explore why scaling fails on typical hardware, why adding more hubs won't solve the issue, and how PaytestCore eliminates these bottlenecks entirely.

Understanding Where Stability Breaks Down

The common assumption is that more USB ports equal more capacity. In reality, USB controllers determine how many high-bandwidth devices a PC can reliably support. Every USB port maps to a controller, and in many cases several ports share that same controller, which has a fixed bandwidth budget. When multiple webcams or ADB devices share the same controller, they also share that bandwidth. Once saturated, devices begin dropping frames, transmitting inconsistent data, or disconnecting entirely. This is why even a PC with ten ports may support only two or three 1080p cameras before problems appear.

OCR cameras are especially demanding; while a compressed 1080p stream can consume 150–200 Mbps, our Full HD streams can require up to 240 Mbps. USB 2.0 offers only about 350–400 Mbps of usable throughput, and even USB 3.x bandwidth is quickly exhausted once several streams run simultaneously. The moment multiple cameras compete for the same controller, instability becomes unavoidable, impacting critical functions like contactless card testing and payment terminal probe accuracy.

ADB devices add an additional layer of complexity. Screen sharing via ADB requires continuous frame capture and compression, which adds both bandwidth and CPU demand. When OCR cameras and Android devices, often used for POS-ECR integration testing and ECR-EMV automation, operate on the same controller, they contend for resources, resulting in lag, delayed robotic actions, and inconsistent test outcomes. Even if bandwidth is available, decoding several video streams simultaneously can saturate CPU cores and memory throughput, creating another point of instability. The combination of USB limitations and computer load is why scaling fails so quickly on standard machines running high-volume payment software QA.

Why Hubs Don't Help – and Often Make Things Worse

When devices begin disconnecting or camera feeds become unstable, many teams assume the problem can be fixed by adding a powered USB hub. In practice, hubs never increase bandwidth. They only split the existing bandwidth of their upstream port. Whether a hub is powered or not, every device connected to it still competes for the same controller. Although a powered hub can fix power shortages, it cannot fix USB congestion, and in most cases adding more devices to a saturated controller increases instability. In high-density testing environments, hubs often worsen performance by masking bandwidth issues until the system becomes overloaded beyond recovery, making terminal QA automation unreliable.

Workarounds and Their Limits

There are only three real ways to increase USB capacity: distribute devices across unused onboard controllers, add PCIe USB expansion cards, or upgrade the machine entirely. Most modern motherboards include more than one controller, but ports are rarely labeled in a way that makes this visible. Tools such as Windows Device Manager or Linux's `lsusb -t` can help map ports to controllers, and simply moving a few cameras to unused controllers can dramatically improve stability. When no controllers remain available, PCIe USB cards offer a cost-effective expansion method, because each card adds a new controller. However, even these solutions eventually hit scaling limits, especially once a test setup requires more than five cameras and several ADB devices for comprehensive EMV testing and electronic cash register testing. At this point, even high-end desktop PCs struggle to maintain stable parallel streams for extended periods.

USB bandwidth also isn't the only bottleneck. Multi-camera OCR processing, ADB decoding, and test orchestration software all require significant CPU and memory resources. Once bandwidth issues are resolved, performance may still degrade if the system lacks the processing capacity to handle multiple decoding pipelines simultaneously. This is why traditional PCs, no matter how many hubs or expansion cards are added, cannot sustainably support large multi-terminal test tool setups.

How PaytestCore Solves the Scaling Problem

PaytestCore was engineered specifically to overcome these limits. Instead of relying on two or three shared controllers, it provides eight USB controllers, each with its own fully isolated bandwidth pool. Seven of these controllers power 14 USB ports, two ports per controller, ensuring every camera and Android device receives a dedicated, high-performance path with no contention, while the eighth controller is reserved for Wi-Fi and other internal peripherals to keep external device traffic completely separate.

Combined with optimized CPU and memory architecture, PaytestCore supports continuous OCR camera streaming, stable ADB communication, and robotic payment testing



control even under heavy parallel load. This capability for centralized test execution and concurrent terminal control allows it to power sophisticated L3 test execution engine workflows. In other words, it delivers the kind of throughput that normally requires multiple PCs chained together.

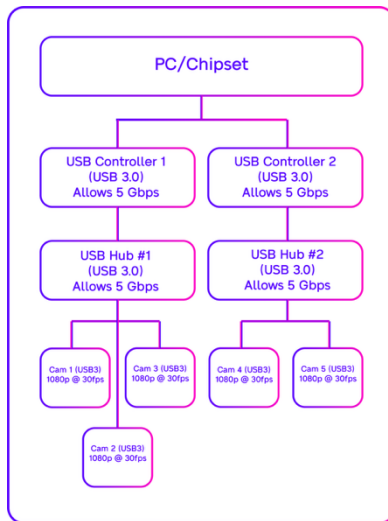
The difference between PaytestCore and a standard PC:

| Parameter | Standard PC (shared controllers) | PaytestCore (8 controllers) |
|----------------------------------|----------------------------------|-----------------------------|
| Stable camera streams | 2–3 cameras | 8–10 cameras |
| ADB devices sustained | 3–4 | 8+ |
| Average CPU usage under OCR load | 80–95% (frequent spikes) | 50–80% (stable) |
| USB disconnection rate (24h) | 5–10 events | 0 events |

Managing large OCR and ADB setups is as much about understanding USB architecture as it is about software configuration. Once you visualize your controllers and distribute devices intelligently, stability improves - up to the point that your hardware allows. But for true scalability, high-throughput testing demands more than ports, hubs, or incremental upgrades. It requires a system specifically designed for parallel, bandwidth-intensive, payment device automation workloads.

PaytestCore was built from the ground up for this purpose. It eliminates guesswork, removes USB bottlenecks, and provides a stable foundation for running multiple robots, cameras, and Android devices in parallel. This is a foundation for a robust payment core testing engine and a reliable payment QA orchestration system. For modern payment testing labs, it represents not just more powerful hardware, but the architecture required to achieve predictable, high-volume, automation-driven testing at scale.

Mapping USB controllers and Hubs:



Consider a desktop machine equipped with multiple front and rear USB ports and configured with four OCR webcams. Although these ports appear independent at the physical layer, they often terminate on the same USB host controller, which forces all connected devices to share a single 5 Gbps bandwidth domain. Three compressed 1080p OCR streams can each consume approximately 150 to 200 Mbps, bringing the aggregate load close to the controller's practical throughput ceiling, especially when legacy USB 2.0 devices introduce additional overhead and reduce available bandwidth. Adding external USB hubs does not expand capacity; it simply redistributes the existing bandwidth across more downstream devices.

A secondary controller, such as one provided through a dedicated PCIe USB 3.0 expansion card, offers an isolated bandwidth pool that allows additional cameras such as Camera 4 and Camera 5 to operate without contention. Understanding the difference between physical port availability and actual controller level allocation is essential for diagnosing and preventing instability in high density OCR and payment automation environments.

As the graph illustrates, bandwidth contention is not a theoretical concern but an inherent limitation of standard PC architecture. Stable multi-device automation requires a system built to eliminate these bottlenecks entirely, and PaytestCore provides the dedicated controller structure needed to achieve that.

If your team is ready to eliminate instability and scale your automated EMV certification environments with confidence, contact us today to learn how PaytestCore can transform your setup.

Matija Mazalin

Head of Technology



sales@paytestlab.com
PaytestLab
Förrlibuckstrasse 66
8005 Zürich, Switzerland

www.paytestlab.com

