VDC|Strategy

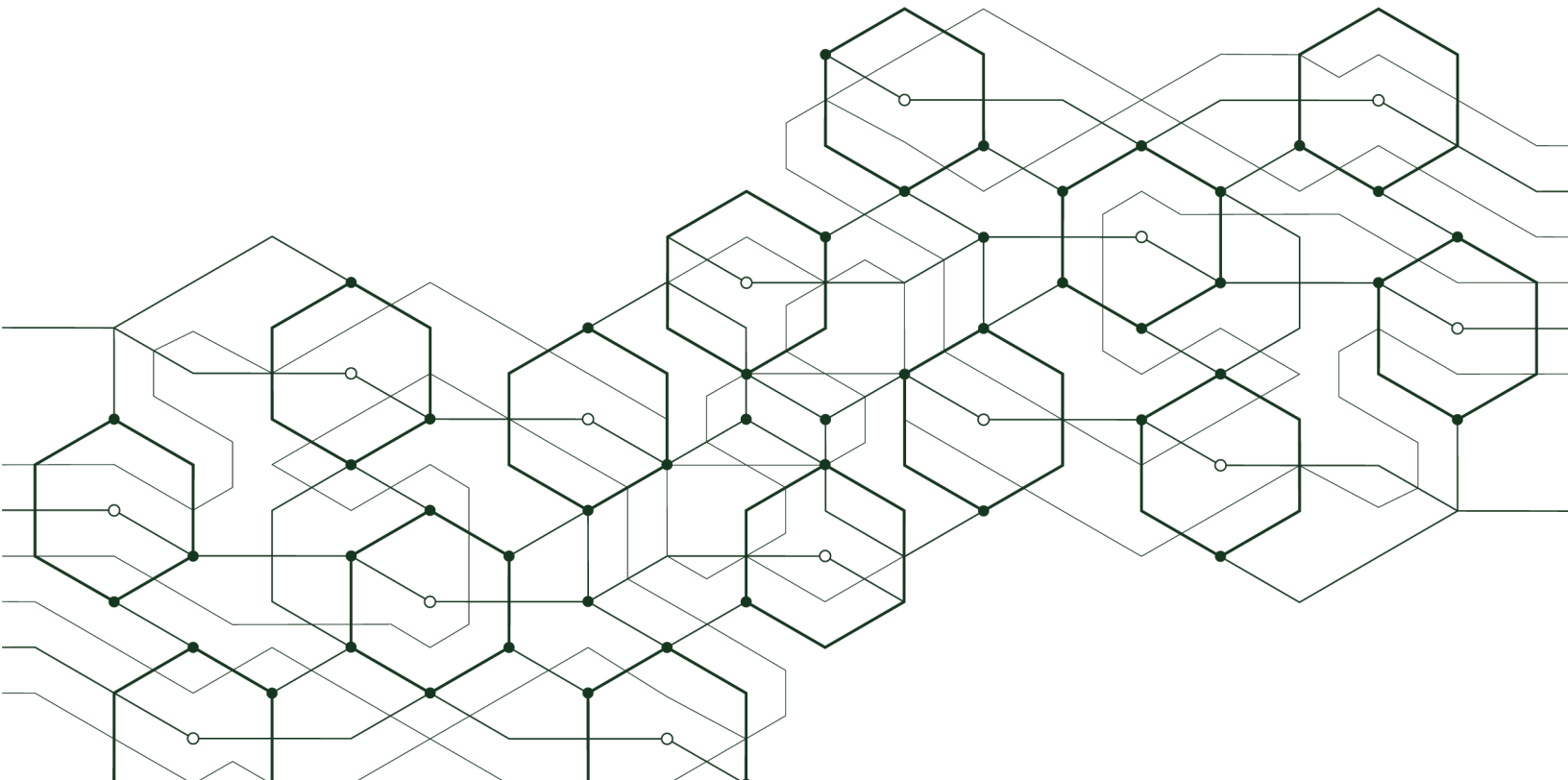
June 2026

VDC|VIEW

Agentic AI at the Edge

Running OpenClaw on a New Qualcomm Arduino UNO Q

by **Chris Rommel**, *EVP*



Introduction

As AI moves from centralized experimentation toward real-word deployment, the next phase of innovation will depend on how easily developers can bring intelligence to the edge. Qualcomm has been increasingly clear about its ambition to expand its role in the developer ecosystem. From Foundries.io to Edge Impulse to Arduino, the company has been adding pieces that connect silicon, AI, connectivity, edge computing, and developer enablement. The UNO Q launch and Qualcomm's acquisition of Arduino represent the firmest signals yet of that direction. The launch of the UNO Q put Dragonwing-based compute into a familiar Arduino form factor. Beyond that, App Lab creates a more unified workflow across Linux, Python, and real-time control as well as AI development. These moves help codify the broader shift from platform capability alone to a movement to promote developer accessibility and activation.

AI and compute increasingly span cloud, PC, mobile, automotive, industrial, embedded, and edge environments. This dynamic demands workflows that can move across those environments, not remain tied to one class of device. The UNO Q offers a unique combination of Linux-class compute, real-time control, Arduino accessibility, and edge AI relevance. In this framing, the UNO Q serves as a practical edge and compute-continuum test case and useful proxy for testing whether agentic tools can operate closer to the target device.

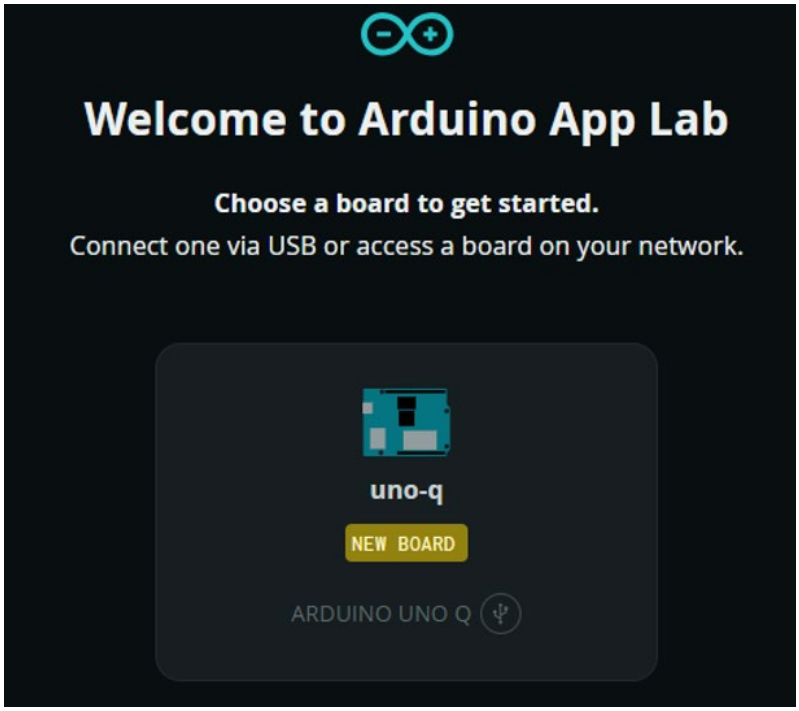
Engineers need new ways to experiment and bring products to market. Playing the role of novice Arduino user, we experimented with a new Qualcomm Arduino UNO Q updated through Arduino App Lab Software, installed OpenClaw, and developed an initial edge workload. While our experience documented a practical first-run experience, it was not a polished benchmark. Our research note documents what worked, what required adjustment, and what the experience suggests for broader edge-agent adoption.

Bringing Up the Agentic Workflow on UNO Q

The real test of any platform is not simply whether the hardware is capable, but how quickly a developer can move from setup to a functional workflow. Time always matters when beginning a new development project. Our experiment here was no different. Delays in setup or configuration can quickly undermine developer confidence and reduce their willingness to continue a prototype. Here, setup quickly moved from baseline device to a functional agentic workflow. Beginning from a clean device baseline after the App Lab software update, there are install **instructions** that are simple to follow. Furthermore, our use of ChatGPT allowed us to make the setup flow more repeatable and simply helped us navigate a few stumbling blocks. Ultimately, OpenClaw successfully installed and configured locally on the UNO Q without much friction, even declaring:

"Butter your workflow like a lobster roll: messy, delicious, effective."

The local-first configuration reinforced the edge value proposition. We configured OpenClaw to run on the UNO Q rather than through a remote gateway, which allowed the workflow to center on the device and its actual environment. The local gateway then allowed us to validate through needed system health checks and mirrored the idea of development happening closer to the target system.



The runtime bring-up showed the value of flexible developer tooling. While some manual setup and gateway management was required during first-run configuration, the process remained workable and transparent enough to keep moving. Moreover, the AI chatbot assistance helped compress the path from instructions to execution and served as another useful example of how developers can bootstrap new workflows on accessible Qualcomm-powered edge hardware.

When experimenting with agentic workflows for the edge, model flexibility is critical and helps maintain a productive workflow. After initially testing a path based upon OpenAI/Codex, we quickly reached usage limit issues during execution. The workflow continued by switching to Anthropic via an API key and selecting Claude Sonnet. The value of that pivot was not lower token usage directly, but the ability to keep the agentic workflow moving by shifting to another available model. The flexibility turned a potential blocker into a straightforward configuration change.

This experimentation informs a broader lesson for edge-agent adoption. Developer momentum is best enabled via repeatable setup, visible runtime state, and flexible model routing. From our experience, UNO Q helped lower the barrier to Qualcomm-powered edge compute experimentation. To this end, agentic tools compress the path from device access to local inspection and software creation. As we look forward, this trend and utility mark a positive signal for Qualcomm's compute-continuum vision: accessible hardware plus adaptable tooling can help activate new edge development workflows.

From Hello World to Local System Monitoring

Many development workflows begin with a simple hello-world test. This was no different. Beginning from a basic Python script created in the local workspace, we quickly made a runnable Hello World script on the UNO Q. This served as confirmation the agent could operate against the local file system before moving into more meaningful edge-related tasks.

The development of a system monitoring function is where we transition from simple system tests to a relevant edge use case. The agent was asked to create a local system monitor for the UNO Q. We queried for CPU, memory, disk, uptime, and thermal data. The result was a safe, simple, and dependency-free use case that was able to run locally with limited system resources. More importantly, the task tested whether the agent could adapt to the local environment rather than simply generate generic code. The agent recognized that external Python dependencies were not available and used standard Linux interfaces instead. After checking multiple thermal zones exposed by the device, the agent generated a standard library Python implementation based on that local system data.

In the IoT and edge world, system telemetry drives insights, uptime, and high-consequence maintenance decisions. Our example showed the use of an agent to read that system telemetry directly from the device without any added packages or complex setup. The utility was tailored to the actual UNO Q runtime. That alone represents a stronger proof point than any hello world because it is paired with local context within that edge compute device. It is a practical demonstration of an agent building against the target environment rather than any abstract device profile an engineer might otherwise evaluate.

A Proof Point for the Qualcomm Compute Continuum

While the exercise was small in scope, its implications extend beyond a single board or agentic workflow. Engineers need flexible solutions. The OpenClaw exercise provides a small but useful proof point for Qualcomm's compute-continuum narrative. Qualcomm's offerings span mobile, PC, automotive, industrial IoT, robotics, Edge, and the data center. Given the convergence of these domains, developers need tangible entry points into that continuum. The UNO Q helps connect the well-known Arduino accessibility with Qualcomm-powered edge compute.

New edge work streams enabled by agents are becoming the norm, not the exception. From local device bring-up assistance to local diagnostics and monitoring, script and tool generation, and hardware-aware prototyping, workflow automation must directly target edge systems.

The combination of Arduino with Qualcomm provides the platforms and synergies many developers need. Arduino provides a familiarity, accessibility, and community that the ecosystem yearns for, especially in a rapid rate of change around edge technologies. Meanwhile Qualcomm brings compute, AI acceleration, connectivity, and platform breadth to the equation. The combined opportunity is to move more developers from simple prototyping into AI-enabled edge system development, addressing a challenge the industry has been working to solve for over a decade. Now the strategic value extends beyond just boards. It expands the base of developers building for Qualcomm-powered environments in a way that can be leveraged across devices and ecosystems.

The economics of agentic workflows also matter. Cloud-only approaches can create token, latency, and cost pressures for developers, especially as experimentation at the edge scales. The industry has long needed ways to make advanced edge experimentation more accessible and economically practical. Local or hybrid orchestration can shift more work closer to the edge and, in this case, Qualcomm-powered devices. The UNO Q shows a lower cost entry point for experimenting with agentic workflows and reinforces Qualcomm's role in enabling agentic AI across price points and device classes, all while preserving flexibility across AI model providers.

Edge Agents Are About Local Context, Not Just Local Execution

What the Test Demonstrated

Deploying agents on edge-based devices introduces new considerations, as will any first-run experience with emerging technology. In this case, installing OpenClaw on the UNO Q on the whole proved relatively pain-free. The gateway became reachable after some troubleshooting, and switching to Anthropic enabled us to work around the Codex usage-limit issues. Most importantly, the agent created, tested, and explained local code on the device, enabling us to have the target-specific insights needed for any edge deployment.

What the Experience Revealed

The available setup and instructions worked well, but first-run workflows still matter. Gateway and runtime visibility can either build or undermine developer confidence. Likewise, the model-routing and token use experience demonstrated how AI provider and model-route flexibility can be an equally important consideration. The ability to switch from the OpenAI/Codex route to Anthropic helped keep the workflow moving.

VDC's View

As with any prototyping or development, installation is only the first step. The real value came from the agent's ability to use local device context and the system monitor script made the edge-agent value more concrete. The challenge with embedded software development and, more specifically, the ability to leverage enterprise software technology – has always been the ability to adapt to small-footprint target devices. Without the memory or system resources available on a server or PC, it is critical to find solutions that can enable portability from high-end systems, to the smallest, workload-optimized ones. This dynamic is no different. Agentic workflows only become meaningful when the agent can adapt to the target platform.

OpenClaw on UNO Q served as a credible edge-agent proof point. The experience surfaced practical first-run lessons around routing and runtime management. The result was directionally compelling: an agent running locally on a small device created software based on the system it was actually running on. Looking ahead, this edge-agent proof point makes the concept more concrete. Local agents become most valuable when they can understand and adapt to the device they are running on. That aligns well with market needs around edge development and with Qualcomm's broader compute-continuum vision.

About the Author

Chris Rommel
Executive Vice President,
Engineering & Industrial Technology

For nearly two decades, Chris has helped clients drive growth in operational and industrial technology businesses, from start-ups to middle-market companies to global enterprises. He has helped a wide variety of clients respond to and capitalize on diverse opportunities in AI, security, IoT, Industry 4.0, the intelligent edge, value-added hardware, semiconductors, engineering solutions, industrial and operational cloud computing and more. Chris has extensive growth strategy consulting expertise including new market assessment, product strategy, M&A diligence, partner and ecosystem development, thought leadership content creation, and more. A frequent speaker at major industry events, Chris has written and published extensive research and thought leadership content on many dynamic technology markets. Chris holds a B.A. in Business Economics and a B.A. in Public and Private Sector Organization from Brown University.

Email Chris at **crommel@vdcstrategy.com**

About VDC Strategy

Founded in 1971, VDC Strategy provides in-depth insights to technology vendors, end users, and investors across the globe. As a market research and consulting firm, VDC's coverage of AutoID, enterprise mobility, industrial automation, and IoT and embedded technologies helps our clients make critical decisions with confidence. Offering syndicated reports and custom consultation, our methodologies consistently provide accurate forecasts and unmatched thought leadership for deeply technical markets. Located in Southborough, Massachusetts, VDC prides itself on its close personal relationships with clients, delivering an attention to detail and a unique perspective that is second to none.

For more information, contact us at info@vdcstrategy.com