White Paper
Manufacturing
December 2023

IT@Intel: Intel Cuts Downtime and Costs with Fault Detection Systems for Factory Equipment

Intel IT implements closed-loop control of subfab equipment by developing an innovative infrastructure based on MQTT, open-source software and Intel® IoT Gateways that communicate with open standards and proprietary protocols

Authors

Robert Colby
Principal Engineer,
IT Infrastructure

Paul Donohue
Senior Systems Architect,
IT Infrastructure

John Mao
Senior Software Architect,
Manufacturing

Table of Contents

Executive Summary

For many years, Intel IT has utilized a standard infrastructure with Industrial Internet of Things (IIoT) sensors and Intel® IoT Gateways for various applications on the factory floor. Simultaneously, we have employed a separate and expensive vendor-based system for fault detection and classification (FDC) on factory equipment. This system has contributed to reduced factory downtime and enhanced predictive maintenance on certain subfab equipment, such as pumps, gas abatement systems, chillers, and more.

To lower costs, increase quality and accelerate necessary changes in the factory, we explored displacing the vendor's system with our standardized IoT infrastructure. We also added new, innovative command-and-control capabilities. As we connected tools to our IIoT infrastructure, our system proved to be 10x less expensive than the vendor's solution. It is also more flexible and can communicate with open standards and proprietary protocols.

Having demonstrated lower costs and improved product quality, we are now scaling our IIoT-based FDC solution to over a thousand subfab tools. We estimate the solution's value as high as USD 16 million for just a single factory that contains hundreds of subfab entities that support the wafer-etching process.

Contributors

Haim Lichaa, Senior Systems Architect, Intel IT
James O'Neill, Network Engineer, Intel IT
Joe Sartini, Industry Engagement Manager Manufacturing 4.0

Acronyms

FFU	fan filter units
FDC	fault detection and classification
IIoT	Industrial Internet of Things
MQTT	message queue telemetry transport
SECS	Semiconductor Equipment Communication Standard

Background

Intel IT constantly pursues factory automation and efficiency. To that end, we have been developing Industrial Internet of Things (IIoT)-based fault detection and classification (FDC) solutions in Intel's factories for almost a decade. We started with small but impactful projects in Intel's fabrication facilities (fabs). For example, we installed sensors and Intel® IoT Gateways to monitor the condition of fan filter units (FFUs) and send data to an analytics application; if excessive vibration data indicated an FFU was about to fail, the application issued an alert. Our IIoT FFU project reduced unplanned downtime due to FFU failure by 66% compared to manual inspection. IIoT solutions also enable us to reduce planned downtime by predicting when equipment will fail and running the equipment longer between scheduled maintenance.

We now have several IIoT projects in production, such as the following:

- Pump vibration monitoring for predictive analytics
- Capital tool asset tracking
- Lithography chemical temperature monitoring
- Mobile equipment tracking that is regulated by the U.S. Occupational Safety and Health Administration (OSHA)
- Electrical tool tracking to allow construction trade partners to track the tools that they own

However, monitoring equipment—collecting data from the equipment—is only part of the bigger picture. To benefit further from directly interacting with tools and equipment, we saw value in also directly controlling the behavior of monitored equipment. Gathering data from a machine, analyzing it, and then sending commands back to the machine to modify its behavior is called "closed-loop control." Controlling the behavior of a machine automatically can increase factory efficiency by reducing downtime and scrap. These have always been important goals, but are now even more crucial as Intel builds its Foundry Services (IDM 2.0).

The subfab equipment in Intel's assembly and test factories seemed a good candidate for establishing closed-loop control. The subfab environment consists of equipment located under the factory floor that provides critical support for overall factory health and uptime. Subfab equipment includes pumps, fans, chillers, heat tracers, chemical delivery and gas abatement.[1] We estimate that 10% of fab unscheduled downtime is due to subfab equipment failures.

Unlike the main factory floor tools, the subfab equipment data was not well integrated into fab automation systems due to the lack of a standard Semiconductor Equipment Communication Standard/Generic Equipment Model (SECS/GEM) data communication protocol from subfab equipment vendors. For on-equipment monitoring capabilities, tool owners have few options due to this lack of communication standards, which creates inefficiencies and requires additional effort for tool owners. For some critical subfab tools, we use a vendor's product to connect equipment data to fab station controllers. However, the vendor's product was expensive and had limited scalability. Additionally, using this product to establish closed-loop control required additional costs. As we explored alternatives to the vendor's product, we established the following solution requirements:

- Reusable infrastructure that scales to many use cases.
- Affordable and scalable components, using open-source software to avoid incremental licensing costs.
- Easily supportable with internal expertise.
- Common security design.

Then, we realized that the solution was under our own roof: Our existing IIoT infrastructure based on Intel IoT Gateways could be extended to closed-loop control use cases and was 10x less expensive than the vendor's product.

Solution Overview

We invented our own closed-loop control solution—currently, there is nothing similar available in the general market. The solution is open-source and based on Intel® architecture (see Figure 1).

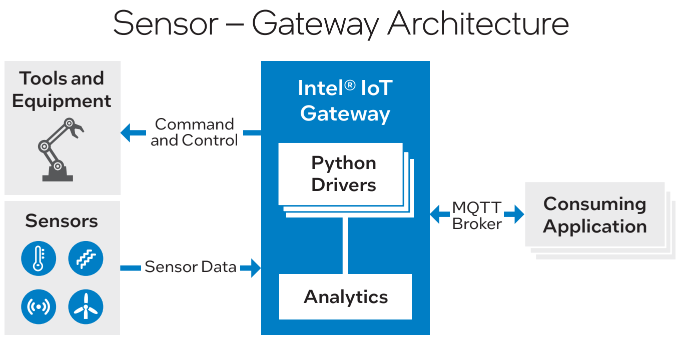
Sensor – Gateway Architecture

Figure 1. We expanded our affordable, open-source-based IIoT architecture to include closed-loop control capabilities in addition to equipment monitoring.

[1] Abatement equipment removes toxic chemicals, such as gases, from the factory environment.

The "brains" of the system are a collection of Python drivers—written by software developers in any Intel group who wants to contribute to the code—that control both the data collection and the command-and-control capabilities. This open-source solution has zero license costs and democratizes the solution; if another machine needs closed-loop control, an engineer can simply add a new driver to the library. The solution consists of the data collection plane and the control plane:

- **Data collection plane.** Python drivers set up the communication with the equipment, collect data, analyze and format the data, and publish it to consuming applications using the message queue telemetry transport (MQTT) protocol.
- **Control plane.** The same Python drivers apply logic to create the control command, send the command to the relevant equipment and verify that it succeeded. The result of the command is published to the appropriate backend manufacturing system, again using MQTT.

Solution Architecture

The software and hardware stack is fully standardized and reusable for many use cases. Let's look at each aspect of the closed-loop control infrastructure in more detail (see Figure 2).

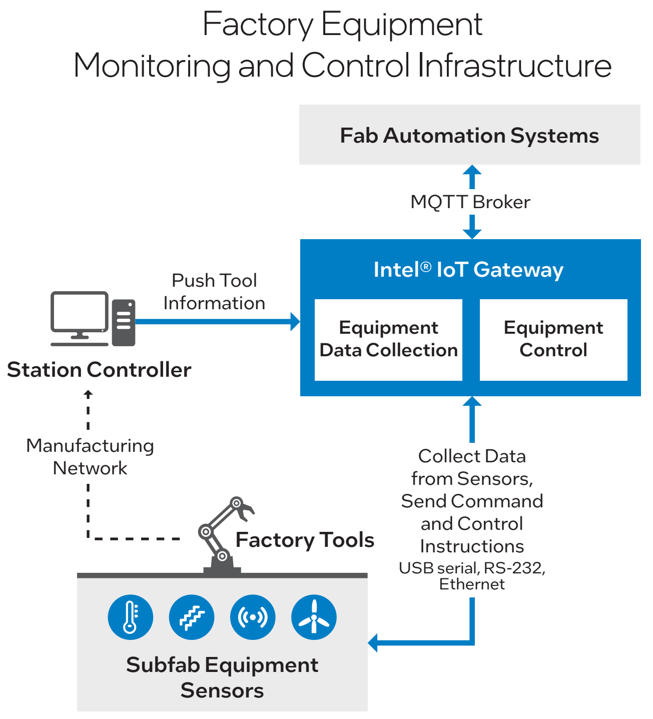

Figure 2. Our common IIoT infrastructure enables tool health monitoring and closed-loop control.

Data Collection

Some subfab equipment had integrated sensors and communication ports and could send data directly to the gateway; other equipment did not have a sensor and needed one installed. We have provided tool owners with a standardized collection of sensors, which makes implementing a new use case quick and easy. If engineers had to shop and compare sensors for a new IIoT use case, it would slow deployment and reduce return on investment. To accelerate onboarding a new use case, we developed an online catalog of pre-validated gateways and sensors for engineers. Our IIoT infrastructure governance does not allow one-off deployments of non-validated sensors, but we will work with our partners to standardize a new sensor where gaps in our catalog exist. A firm commitment to standardization helps scale the benefits of IIoT.

We have validated one or more sensors in each of the following categories. Sometimes, we validated several versions of the same sensor because some have multiple connectivity options, like Wi-Fi or USB.

- Vibration
- Wireless Gauge Reader
- Temperature
- Acoustic
- Vision
- Humidity
- Sonar
- Leak Detection
- Differential Pressure
- Accelerometer

Intel IoT Gateways

Intel IoT Gateways are crucial for collecting sensor and embedded controller data at the network edge, and then filtering it to analyze and normalize the data for sharing. Intel IoT Gateways are low-cost compute devices; our model has an Intel Atom® x6425RE processor (1.90 GHz). We aim to share how we implemented our solutions so you can understand how they can help solve your company's challenges.

Software and Analysis

Our IIoT infrastructure runs on open-source Linux. The gateway's software application separates core functionality from use-case-specific functionality.

- **Core functionality —** Supports all use cases and is installed across the entire fleet of gateways. We update core functionality on an annual basis.
- **Use case-specific code —** The Python drivers can be frequently developed, tested and released to gateways that will be used for a particular use case without requiring any change to gateways used for other use cases. We tag each gateway so that its use can be tracked to an approved use case and its associated business value. Tagging the gateways enables us to accurately count instances, realize the full business value of that use case, and disconnect the gateway if the use case is not generating business value.

The Python drivers enable us to directly connect to various factory equipment, whether it is SECS-compliant or not.

Once the data is collected, the drivers format it appropriately and then use unsupervised machine-learning algorithms (developed by Intel IT) to analyze it. The algorithms use historic data and preset thresholds to determine if the data represents a potential equipment issue.

The analysis results are sent through the gateway to central systems through an MQTT broker using a "sub/pub" model. Most of these systems reside in Intel's private factory cloud, called the Intel Manufacturing Cloud.

Fab Automation Integration

With the data communication in place, online automation systems inside the fab subscribe to the data from the MQTT broker and use it to make process control decisions. Several types of fab responses are possible when a parameter (such as pump vibration) reaches or exceeds its upper control limit:

- **Notify the tool owner** of the issue and keep the current fab operation running until it is done. Tool owners can plan preventive maintenance within a few hours of anomaly detection. For example, when the vibration intensity of a dry etch pump reaches its upper control limit, the pump can continue working for another four to five hours. Tool owners can keep the operation running for one more hour. After the operation is finished, the tool owner can stop the tool and schedule the maintenance.
- **Stop the operation** within a few seconds of anomaly detection. For example, if a pump pressure exceeds its upper control limit, which indicates that something is preventing a gas or liquid from flowing, it would be best to stop the operation immediately.

Setting the proper control limits for parameters (such as the allowable amount of vibration or temperature) is critical for effective tool health monitoring. We set different limits, each of which is associated with a particular response. Figure 3 is a graph from the dashboard showing that a parameter is exceeding the upper control limit, indicating that a tool fix is needed. After the fix, the parameter reading drops to the lower control limit, indicating normal tool behavior.

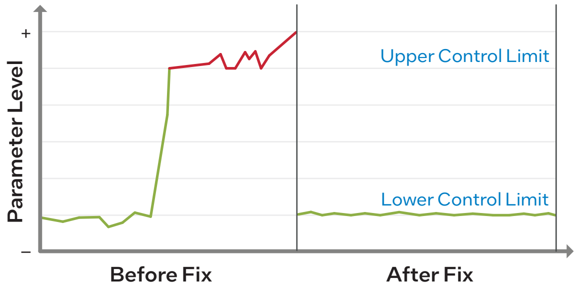

Figure 3. A tool parameter exceeded its upper control limit. The fab tool owner was notified to perform preventive maintenance and fix the tool.

Designing for Agility

The factory and subfab environments can sometimes be dynamic. When factory tool owners ask us to add new tools, there is little technical overhead to accommodate their requests. Our centralized repository of drivers makes it easy to build on existing knowledge and add new drivers. It typically takes about a week between connecting to a tool for the first time and publishing proof of concept output to the MQTT broker and displaying user graphs. The new driver can be put into production after just a few weeks of validation and testing.

This streamlined approach to extending capabilities helps us maintain our primary focus—generating value at scale to ensure we properly prioritize requests.

Provisioning for Scalability

We have proactively worked with the OEM that builds our Intel IoT Gateways so that the gateways include the OS and our IT build when they ship. This enables factory tool owners—who are not IT experts—to automatically provision the system without assistance from IT staff. Tool owners can simply un-box the gateway and power it up without a monitor, keyboard or any user interaction. The gateway completely provisions itself, and then the tool owner receives an email confirmation that the provisioning is complete. The result is a system that scales easily and does not incur support costs.

Automated, autonomous provisioning is highly valuable in any manufacturing environment. Other manufacturing industries—beyond silicon wafers—can use the same approach. For example, staff using paint tools in an automotive factory might not have experience in IT and programming. However, they may want to know how much paint is delivered so they can help the company save money. By enabling the gateway to build itself, it is possible for the painter to simply take a gateway out of its box and connect it. With a few clicks, the painter can quickly see results on a graph and receive alerts if the tool sprays too much paint.

Security

Similar to having a standardized infrastructure, a uniform approach to security means we don't have to reinvent security policies for every use case. There's already a common security policy and a single set of security controls. We have nearly completed the definition of an IoT Security Standard that satisfies Intel's Minimum-Security Specification. The IoT Security Standard covers IoT and IIoT sensors, data, applications, networks, and relevant upstream and downstream infrastructure. The new standard also specifies security practices for threat and vulnerability management, monitoring and logging, business continuity, incident response and documentation. Implementing such a standard for our IIoT solutions helps ensure systems remain compliant with Intel's overall security posture and keeps Intel's valuable intellectual property as secure as possible.

Best Practices for Applying IIoT-Based Fault Detection Systems to Other Manufacturing Environments

Many other manufacturers have factory tools similar to those in Intel factories, which may not support the industry-standard communication protocol for that industry.

Within our own factories, we have proven this solution to be flexible enough to communicate with and control various subfab machines, from pumps to gas abatement equipment to chillers. We believe our standardized, low-cost solution can be deployed in other types of factories, and we are interested in sharing our discoveries and best practices. Standardization—and the resulting scalability and economy of scale—is at the heart of our success.

Standardize Everything

- Single Intel® IoT Gateway model for all use cases globally, shipped from the OEM with the standard IT build.
- Shared infrastructure with zero one-off deviations.
- Common Python driver development method for all new use cases.
- Standard security model that meets InfoSec policies.
- Single support model for all use cases.

Establish Centralized Governance

- Form an IoT technical working group or council that reviews proposed IIoT projects.
- Ensure a project aligns with the security standards, has a standard support model and meets business value metrics.
- If the project doesn't meet all these criteria, it doesn't get a network connection.

Results

Our IIoT infrastructure is a "build-once-use-everywhere" solution that is highly scalable. We have created a catalog of pre-validated sensors, IIoT gateways and drivers so that as new tools land in the subfab environment, establishing closed-loop control of that new tool is fast, efficient and repeatable. We avoid costly and complex vendor-based one-off solutions and have a single support model and set of processes. The solution is also highly affordable (even for just one use case)—10x less expensive than the vendor's solution we were using, which did not support closed-loop control without additional cost.

Multiple use cases can reside on a single gateway. Our standardized IIoT infrastructure results in more efficient and reliable factories with demonstrable cost savings through reduced scrap and less factory downtime. Data provided by factory tool owners shows that we are detecting problems and taking action through fab automation systems to mitigate those problems. The solution is applied to wafer process fabs and used in assembly and testing manufacturing (ATM). With the IoT solution integrated with fab and ATM automation systems, timely alerts from our FDC IIoT solution have enabled us to save Intel products from becoming scrap.

Conclusion

We are actively scaling our solution for IIoT-based equipment monitoring and control in Intel's high-volume factories and we continue to expand the use cases on our IIoT infrastructure. Replacing the vendor's monitoring systems with a more capable, less-expensive solution (up to 10x cheaper) provides significant savings. Additional benefits from reduced scrap and less factory downtime can provide additional savings—up to USD 16 million per factory. We hope our IIoT success inspires other manufacturers to build comparable solutions and achieve similar results.

Related Content

If you liked this paper, you may also be interested in these related stories:

- Expanding Low-Cost Industrial Internet of Things (IIoT) Manufacturing Use Cases white paper
- Smart Manufacturing Using Computer Vision and AI for Inline Inspection white paper
- Minimizing Manufacturing Data Management Costs white paper
- Reliability Engineering Helps Intel Cut Manufacturing Downtime in Half white paper
- Transforming Industrial Manufacturing with Software-Defined Networking white paper
- Accelerated Analytics Drives Breakthroughs in Factory Equipment Availability white paper
- Transforming Manufacturing Yield Analysis with AI white paper
- Streamline Deep-Learning Integration into Auto Defect Classification solution brief

For more information on Intel IT best practices, visit intel.com/IT.

IT@Intel

We connect IT professionals with their IT peers inside Intel. Our IT department solves some of today's most demanding and complex technology issues, and we want to share these lessons directly with our fellow IT professionals in an open peer-to-peer forum.

Our goal is simple: improve efficiency throughout the organization and enhance the business value of IT investments.

Follow us and join the conversation on **X (formerly known as Twitter)** or **LinkedIn**. Visit us today at **intel.com/IT** if you would like to learn more.