Industrial grade enhancement for human machine interface with controller-based semiconductor module

Mentor

Akash Kumar

Staff engineer at Aark Global Inc.

Team members

Srihari VV

Tarun MR

Veeranandha Kumar G

Abstract

In the fast-paced and precision-driven environment of semiconductor manufacturing, wafer alignment and tracking systems play a pivotal role in ensuring accurate layer deposition, reduced defect rates, and optimized production throughput. This project presents the design, enhancement, and system-level transformation of a wafer alignment and counting module, developed specifically for integration into semiconductor fabrication workflows.

The initial prototype—originally built on a Raspberry Pi 4 platform with LDR sensors, a basic DC motor, and Node-RED for visualization—has been significantly upgraded to meet industrial and Industry 4.0 standards. The upgraded system is centered around the **STM32F407VG microcontroller**, which brings advantages of real-time embedded control, lower power consumption, high peripheral integration, and long-term reliability. By integrating **FreeRTOS**, the system now operates with deterministic task scheduling, enabling precise control over wafer edge detection, alignment correction, and data communication without the unpredictability of Linux-based multitasking.

Sensor performance has been dramatically improved by replacing low-speed LDRs with **high-speed photodiodes** and optional **laser triangulation sensors** for non-contact distance measurement. These upgrades allow for microsecond-level edge detection, ensuring real-time feedback and correction. As an advanced extension, the system architecture also supports **vision-based wafer alignment**, utilizing basic image sensors and onboard image processing for fiducial detection, further enhancing positioning accuracy.

The motor subsystem has been transitioned from an open-loop brushed DC motor to a **closed-loop stepper or BLDC motor**, driven by modern controllers like **DRV8825** or **STSPIN**. Combined with encoder feedback and PID control algorithms, this setup delivers high-precision wafer positioning, smooth motion profiles, and reduced mechanical wear. The STM32F407's hardware PWM timers and encoder interface facilitate accurate motion synchronization in real time.

Human-Machine Interface (HMI) capabilities have been upgraded from Node-RED to a more professional solution such as a **PyQt5-based GUI or Ignition SCADA platform**, enabling features such as real-time process visualization, interactive status indicators, event logging, trend graphs, and alarm handling. Additionally, **web and mobile accessibility** are enabled to support remote monitoring, a key feature in Industry 4.0 environments.

To ensure full industrial interoperability, the system now supports multiple **communication protocols** including **Modbus RTU/TCP**, **CANbus**, **MQTT**, and **OPC UA**, enabling integration with programmable logic controllers , supervisory control and data acquisition (SCADA) systems,

and cloud platforms. These communication enhancements allow the system to function as an intelligent IoT node in a larger automation network.

Reliability and safety are addressed through multiple layers of protection: hardware **watchdog timers**, **CRC memory checks**, **opto-isolation**, and **transient voltage suppression (TVS)** are incorporated to ensure fault-tolerance under industrial conditions. Furthermore, power stability is ensured through brownout detection and voltage regulation safeguards.

The final enhancement is **edge-to-cloud integration**, where wafer count and alignment data can be streamed securely to **cloud platforms** such as **AWS IoT Core**, **Azure IoT Hub**, or **ThingsBoard** via secure MQTT or HTTPS protocols. This feature supports centralized monitoring, historical data storage, and **predictive maintenance** using cloud-based analytics or AI models.

Overall, this upgraded wafer alignment system transitions from a basic simulation into a **fully scalable, real-time, industrial-grade embedded platform**. With its STM32-based core, advanced sensing and actuation, professional HMI, industrial communication support, and IoT/cloud readiness, the system exemplifies a complete migration toward modern **smart factory** and **semiconductor automation** standards aligned with **Industry 4.0**.

Objective

The objective of this project is to design and implement a real-time, intelligent wafer alignment and counting system for semiconductor automation using an STM32F407VG microcontroller. The project aims to improve accuracy, system responsiveness, and industrial scalability by integrating embedded control, advanced sensing, and professional HMI capabilities.

Key objectives include:

- **Real-Time Control**: Replace Raspberry Pi with STM32F407VG to enable deterministic, low-latency motion control using FreeRTOS. This ensures precise timing for wafer alignment and counting tasks.
- **Precision and Throughput**: Enhance sensing and actuation using high-speed photodiodes and closed-loop stepper or BLDC motors with PID control to improve alignment accuracy and increase operational speed.
- **Professional HMI**: Develop an advanced human-machine interface using platforms such as PyQt5 or Ignition SCADA to display wafer status, trends, alarms, and system diagnostics in real time.
- **Industrial Communication**: Integrate protocols like Modbus RTU/TCP, CANbus, OPC UA, and MQTT to support connectivity with STM32, SCADA systems, and cloud platforms for seamless data exchange.
- **Reliability and Safety**: Incorporate watchdog timers, CRC checks, and hardware protections (opto-isolators, TVS diodes, brown-out detection) to ensure safe and fault-tolerant system operation in industrial environments.
- **IoT and Cloud Integration**: Enable cloud connectivity and remote monitoring using protocols like MQTT or HTTPS, allowing data to be uploaded to platforms such as AWS IoT, Azure IoT Hub, or ThingsBoard. Support predictive maintenance and performance tracking through edge-to-cloud analytics.

These objectives collectively align the project with Industry 4.0 standards, making the system scalable, intelligent, and suitable for deployment in modern semiconductor manufacturing environments.

Introduction

In the field of semiconductor manufacturing, **wafer alignment** is one of the most critical stages, ensuring that each processing step—such as lithography, deposition, etching, and inspection—is applied with **micron-level accuracy**. Even the slightest misalignment in wafer positioning can result in **circuit defects, yield loss, or critical device failure**, especially as semiconductor geometries continue to shrink. Therefore, high-precision alignment systems play an essential role in maintaining product quality and process efficiency across fabrication lines.

The original prototype system was developed using a **Raspberry Pi 4**, LDR-based sensors for edge detection, a simple DC motor with L293D for actuation, and **Node-RED** for HMI visualization. While effective as a proof-of-concept, this architecture suffers from limitations that prevent real-world industrial deployment—such as high OS-related latency, lack of deterministic control, slow sensor response, and limited integration with industrial communication protocols or IoT infrastructure.

To address these limitations, this project proposes a comprehensive system upgrade using an **STM32F407VG microcontroller**, which offers a powerful ARM Cortex-M4 core, real-time interrupt handling, and rich peripheral support. The STM32 platform allows the integration of **FreeRTOS**, enabling **multitasking with deterministic timing**, essential for real-time wafer detection, correction, and feedback control.

The **sensor subsystem** is upgraded from LDRs to **high-speed photodiodes**, which provide microsecond-level light response, and optionally, **laser displacement or vision-based alignment systems** can be integrated for advanced feedback and higher resolution. The **motor subsystem** transitions from an open-loop brushed DC setup to a **closed-loop stepper or BLDC motor** with encoder feedback, allowing precise PID control of wafer positioning.

The Human-Machine Interface (HMI) is also enhanced from a basic Node-RED dashboard to a more advanced, professional solution such as **PyQt5 GUI** or **Ignition SCADA**, enabling features like real-time status indicators, trend graphs, error logs, alarms, and remote access through mobile or web platforms.

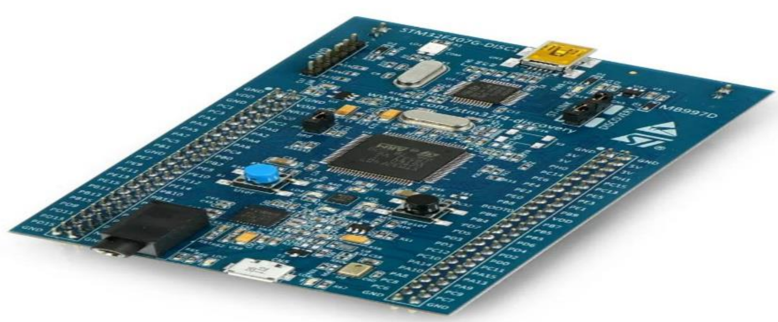
To ensure interoperability with industrial environments, the system now supports **Modbus RTU/TCP**, **CANbus**, **MQTT**, and **OPC UA**, allowing seamless data exchange with programmable logic controllers , SCADA servers, and IoT platforms. Combined with **hardware-level reliability features** such as watchdog timers, CRC checks, opto-isolation, and transient protection, the system becomes robust enough for long-term industrial operation.

Finally, with the inclusion of **cloud connectivity options** (e.g., AWS IoT Core, Azure IoT Hub, ThingsBoard), the system supports real-time data logging, remote monitoring, and predictive analytics. These enhancements transform the original simulation into a scalable, intelligent, and fully integrated **Industry 4.0-compliant wafer alignment system**, ready for deployment in modern semiconductor fabrication facilities.

Rationale for switching RPI5 TO STM32F4O7VG

As the semiconductor industry moves toward highly reliable, real-time, and energy-efficient embedded systems, selecting the appropriate hardware platform becomes critical. The original wafer alignment and counting prototype was built using a **Raspberry Pi 5**, which provided flexibility and ease of prototyping. However, with the release of **Raspberry Pi 5**, while performance has improved, it still remains a general-purpose **Linux-based single-board computer** and is not optimized for deterministic real-time control or industrial automation use.

For a system that demands **millisecond-level responsiveness**, **low power consumption**, and **stable, fault-tolerant operation**, a shift to a **microcontroller-based architecture**—specifically the **STM32F407VG**—is both logical and beneficial. Unlike the Raspberry Pi, which relies on a full operating system (Raspberry Pi OS or Linux), the STM32 MCU can run **bare-metal code or FreeRTOS**, ensuring cycle-accurate timing and strict control over task scheduling—critical for motor actuation, sensor reading, and alignment precision.

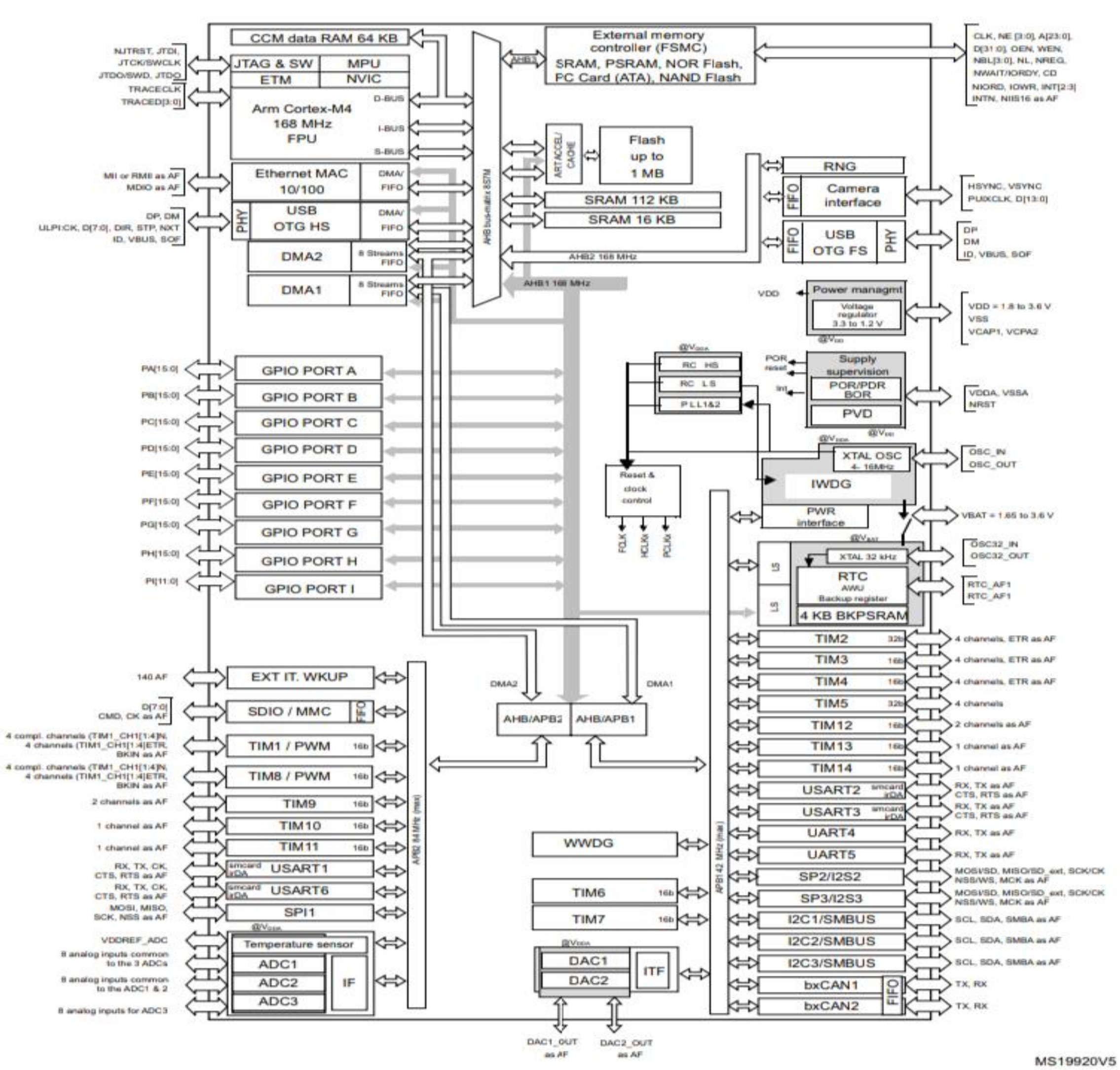


Moreover, STM32 microcontrollers are designed with **built-in peripherals** optimized for real-time applications, including **timers**, **PWM generators**, **ADCs**, **DMAs**, **UART/SPI/CAN interfaces**, and **quadrature encoder inputs**. These are essential for closed-loop motor control and sensor interfacing and are available directly at the silicon level without needing external hardware.

Additionally, STM32F407VG offers a **more compact PCB footprint**, dramatically lower power consumption, and better thermal efficiency compared to Raspberry Pi 5. This makes it suitable for **space-constrained and embedded deployments** in cleanroom environments where minimal heat dissipation and high reliability are critical.

In terms of cost, STM32F407VG units are **substantially more economical**, typically priced between \$5–10, whereas Raspberry Pi 5 boards range from \$60–\$80 (or more in bulk-constrained supply chains). For scalable production or OEM integration, this cost difference becomes a decisive factor.

Finally, the **industrial support ecosystem** around STM32—ranging from ST's long-term product lifecycle, robust documentation, example code (HAL/LL), and professional-grade development tools like STM32CubeIDE—makes it the preferred choice for commercial and industrial applications.



Raspberry pi 5 vs STM32F4O7VG (MATRIX)

Category	Raspberry Pi 5	STM32F407VG
Processor Architecture	Quad-core ARM Cortex-A76 @ 2.4 GHz (Linux-based)	ARM Cortex-M4 @ 168 MHz (bare-metal or FreeRTOS)
Operating System	Full OS (Linux, Raspbian)	No OS / Real-Time OS (e.g., FreeRTOS)
Determinism	Non-deterministic, subject to OS latency	Hard real-time (interrupt and timer-based)
Power Consumption	5V @ 2.5–3 A (12.5W+)	~100–150 mW active, <10 mW sleep
Boot Time	~20–30 seconds (OS-dependent)	<100 ms (instant startup)
Size & Form Factor	85.6 × 56.5 mm (credit card size)	LQFP100 (~16×16 mm IC, custom PCB)
Cost	$60–$80 (board only)	$5–$10 (IC)
I/O Interfaces	GPIO, UART, SPI, I2C via OS-layer	Native GPIO, ADC, PWM, CAN, UART, SPI, I2C, Timers
ADC Support	Not available onboard	12-bit ADC (3.6 MSPS), up to 16 channels
PWM & Encoder Support	Limited via software	Multiple hardware PWM channels and encoder timers
Real-Time Performance	Low (interrupts managed by OS kernel)	High (cycle-accurate interrupt response)
Thermal Efficiency	Generates significant heat; needs cooling	Low power; no heatsink required
Industrial Reliability	Not certified or hardened for industrial use	Widely used in industrial automation and control
Development Tools	Python, Linux SDKs	STM32CubeIDE, Keil, IAR, STM32 HAL/LL libraries
Lifecycle Support	Community-driven, may face availability issues	Long-term support from STMicroelectronics

Embedded Control and Processing Upgrades

To meet the real-time demands of semiconductor wafer alignment, the embedded control system has been significantly upgraded by transitioning from a general-purpose platform to a **microcontroller-based real-time environment**. This section outlines the key enhancements in control architecture, task scheduling, and sensor data acquisition—each designed to improve precision, responsiveness, and stability.

1. Real-Time Multitasking with FreeRTOS

In the original setup, all functions—such as sensor reading, motor driving, and communication—were handled in a sequential loop, which made the system prone to delays, especially when tasks competed for processing time. This limitation has been addressed by integrating **FreeRTOS**, a lightweight real-time operating system tailored for embedded microcontrollers.

By running **FreeRTOS on the STM32F407VG**, the system now supports **multitasking**, where different functions are executed as independent threads with assigned priorities. For example:

- A **sensor task** can continuously monitor wafer presence and alignment feedback.
- A **motor control task** can update PWM signals and PID corrections.
- A **communication task** can handle data transfer to the HMI.
- A **logging or status update task** can manage system messages.

This structure ensures that **time-critical operations** like motor control or edge detection are never delayed by less important tasks, leading to more predictable and reliable behavior—an essential requirement for semiconductor handling equipment.

2. Closed-Loop Motor Control Using PID Algorithm

In semiconductor automation, even a slight positional error can lead to misalignment and production faults. To improve motion accuracy, the system moves away from open-loop DC motor control and now employs a **closed-loop control mechanism** using a **PID (Proportional-Integral-Derivative) algorithm**.

The STM32's internal hardware timers are configured to work with a rotary encoder connected to the motor shaft. This setup provides **real-time feedback on motor position or speed**, which is continuously compared against the target value. The PID controller then adjusts the motor's PWM signal to minimize the error.

- **Proportional** control ensures fast response.

- **Integral** correction eliminates steady-state offset.
- **Derivative** action helps prevent overshooting.

This feedback loop allows the motor to stop exactly where it should—essential for ensuring each wafer is properly aligned before processing continues.

3. High-Speed Sensor Acquisition with ADC + DMA

The sensor array—especially for edge detection—is a critical component of the system. In this upgraded version, the **analog photodiode sensors** are connected directly to the **STM32's 12-bit ADC (Analog-to-Digital Converter)** channels. To ensure that data is captured quickly and without CPU delay, the ADC is configured in **continuous mode** with **DMA (Direct Memory Access)**.

DMA allows sensor data to be transferred from the ADC to memory without interrupting the main program or slowing down other tasks. This method provides:

- **Higher sampling rates** — ideal for detecting fast wafer movement
- **Consistent sensor updates** — ensures no loss of input data
- **Reduced CPU load** — frees up the processor for control and communication tasks

Using this architecture, the system can respond immediately to changes in sensor input, such as wafer arrival or misalignment, which improves both accuracy and overall throughput.

Sensor Upgrades

In precision applications like wafer handling and alignment, the choice of sensors directly impacts the system's responsiveness, accuracy, and reliability. The original system relied on **LDRs (Light Dependent Resistors)**, which, although easy to interface, are **too slow and imprecise** for modern semiconductor automation. As part of the system upgrade, the sensing mechanism has been redesigned with faster and more accurate alternatives suitable for industrial use.

1. Transition from LDR to Photoelectric Sensors

LDRs typically have **response times ranging from 50 ms to several hundred milliseconds**, depending on ambient light and component quality. This delay introduces significant timing errors, especially when wafers move at high speed across the sensor path.

To overcome this limitation, the upgraded system now uses **high-speed photoelectric sensors**. Photodiodes can respond in **microseconds**, making them ideal for real-time edge detection, alignment verification, and slot counting.

The photodiodes are used in **reverse-biased mode**, connected to transimpedance amplifiers to convert the photocurrent into a measurable voltage. When a wafer interrupts the light beam (from an IR LED or laser diode), the voltage output changes sharply—allowing immediate detection by the STM32's ADC.

Benefits:

- Instant detection of wafer edges or gaps

- High immunity to ambient light variations
- Greater repeatability and measurement stability

This change alone significantly improves the speed and accuracy of wafer counting and position detection compared to the original (LED+LDR) setup.

2. Laser Displacement and Distance Sensors (Optional)

In more advanced versions of wafer alignment systems, precise height or position measurements are necessary—especially in inspection or robotic handling scenarios. To address this, the system design supports the integration of **1D laser displacement sensors** or **time-of-flight sensors**.

One example is **laser triangulation sensors** (such as those from Keyence or Panasonic), which can provide **sub-millimeter or even micron-level resolution**. These sensors work by emitting a focused laser beam toward the wafer and measuring the reflected angle to determine exact position or surface height.

This allows:

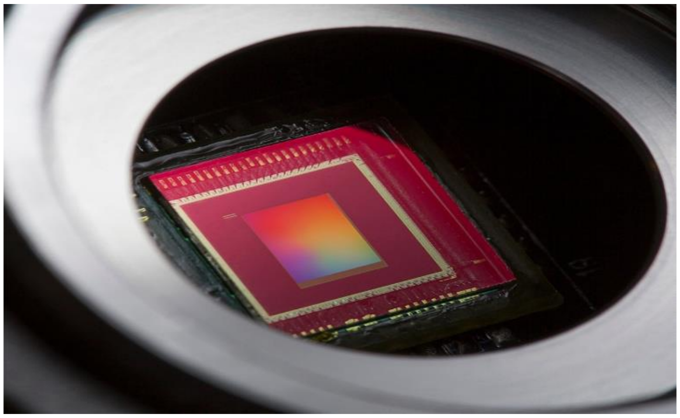
- Measurement of **wafer flatness**
- Detection of **slot depth or edge taper**
- Correction for **tilt or misalignment**

Even though these sensors are optional, designing the control system with flexibility to integrate such hardware adds professional scalability and positions the system for high-end applications.

3. Machine Vision for Automated Alignment (Advanced Option)

For the highest level of accuracy and automation, a **vision-based alignment system** can be incorporated. These systems use small industrial cameras (CMOS sensors) to visually capture wafer position, orientation, or fiducial markers (such as notches or reference dots).

Using onboard image processing (OpenCV or custom edge-detection algorithms), the STM32 (or an external processor, if needed) can:



- Detect wafer center and edge alignment
- Identify rotational misalignment or tilt
- Adjust motor position dynamically based on image data

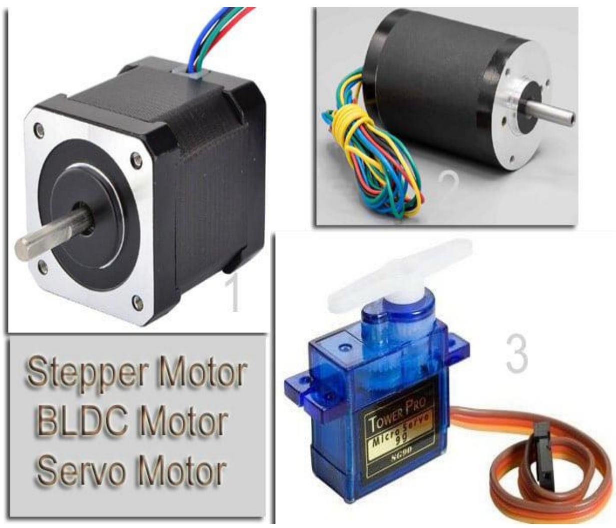
While this approach requires more resources in terms of computation and lighting control, it reflects real-world practices in semiconductor fabs, where vision systems are often used for **non-contact alignment and inspection** tasks.

In our project, the vision module is listed as a future upgrade pathway, ensuring that the base system can evolve into a **more autonomous and intelligent platform**.

Motor and Actuator Enhancements

Motor control is one of the most critical parts of a wafer alignment system. In the original design, we used a **brushed DC motor** controlled through an **L293D driver**, which is basic and easy to set up—but it lacks the precision, feedback, and torque control required for high-accuracy semiconductor processes.

To overcome these limitations and bring the system closer to industry standards, the motor and driver subsystem has been upgraded to support **closed-loop stepper or servo motors**, paired with **modern driver ICs**. These improvements ensure better performance in terms of both accuracy and reliability.



1. Closed-Loop Motion Control

In open-loop systems like a basic DC motor, the controller sends a signal to move, but there is no way to confirm whether the movement actually occurred as expected. This leads to missed steps, overshooting, and uncorrected errors—especially when the motor is under load.

To address this, the upgraded design now uses either:

- A **NEMA 17 or NEMA 23 stepper motor** with an **encoder**, or
- A **brushless DC (BLDC) motor** with **Hall effect sensors** or an **optical encoder**

These motors provide **closed-loop feedback**, meaning the actual motor position is continuously measured and compared to the target position. This feedback is processed using a **PID control algorithm**, which dynamically adjusts the motor’s speed and direction in real time to ensure precise alignment.

This setup allows the system to:

- Hold and maintain exact positions during wafer placement
- Adjust for overshoot or mechanical disturbances
- Perform fine corrections during edge detection or slot alignment

The use of closed-loop motors makes the entire positioning process **more repeatable, responsive, and error-tolerant**.

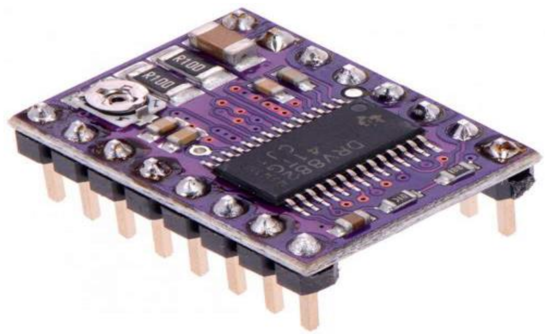
2. Replacing L293D with Modern Motor Drivers

The L293D is a classic H-bridge driver used in many educational projects, but it has several drawbacks for industrial use: limited current handling (~600 mA per channel), inefficient bipolar transistors (instead of MOSFETs), and minimal protection features.

In our project, the L293D has been replaced with more advanced and efficient drivers based on the motor type.

For stepper motor Drivers like **DRV8825** or **TMC2208** are used. These support:

- Microstepping (up to 1/32 or higher) for smooth motion
- Adjustable current limiting
- Built-in thermal shutdown and short-circuit protection
- Quieter operation and lower vibration



For bldc motor The system uses **STSPIN series drivers** from STMicroelectronics. These provide:

- MOSFET-based drive stages for high efficiency
- Integrated current sensing and overtemperature protection
- PWM input compatibility with STM32 timers
- Compact form factor with industrial-grade features

These drivers ensure reliable motor control, better power handling, and smoother motion profiles. They also make the system more robust during long-duration operation, which is important in production environments.

HMI and Dashboard Enhancements

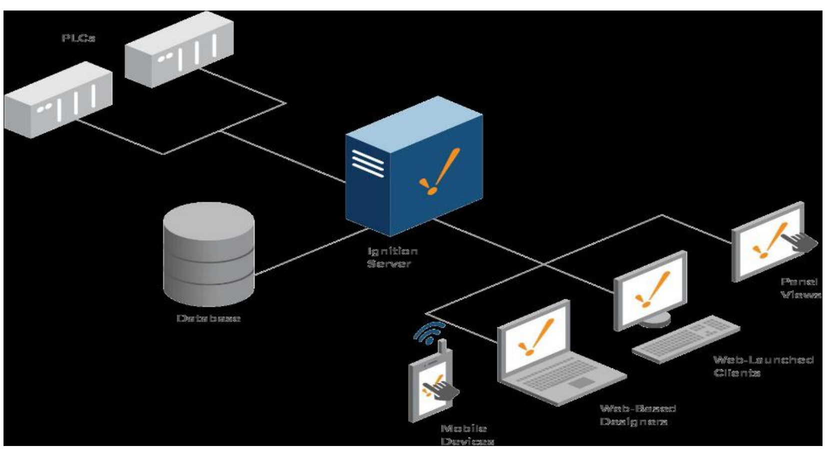
The Human-Machine Interface (HMI) is the central point where operators monitor, control, and respond to real-time system conditions. In the original prototype, a basic dashboard was created using Node-RED. While functional for simulation, it lacks the advanced features, industrial polish, and remote accessibility expected in real-world semiconductor environments.

To bridge that gap, the system has been upgraded to support a more **robust and professional HMI framework**, using tools such as **PyQt5 (for a custom desktop GUI)** or **Ignition SCADA** for browser-based monitoring. These platforms offer a wide range of features such as trend visualization, alarm handling, secure access, and scalable user interface design.

1. Migration to Professional HMI Platforms

Instead of relying on browser nodes and flow-based UI like Node-RED, the upgraded system uses either:

- A **PyQt5-based custom dashboard** developed in Python, which allows tight control over UI layout, logic, and styling.
- Or a **SCADA system like Ignition**, which supports web-based runtime environments, industrial widgets, and data-binding.



Both platforms allow integration with UART, Modbus, or MQTT data sources and can display live values from the STM32 microcontroller in a clean, operator-friendly interface.

This change provides a professional look and feel, makes the system easier to operate, and introduces features such as:

- Real-time process overview
- Graphical indicators (gauges, bar charts, meters)
- Touchscreen support for industrial HMI panels

2. Live Trending and Alarm Management

One of the most valuable features of an advanced HMI is **live data trending**. In this upgraded system, the dashboard can display:

- **Wafer count** over time
- **Motor current or temperature**
- **Alignment error rates**
- **Sensor voltage levels**

The data is logged in the background, allowing the operator or supervisor to **review performance trends** and detect anomalies or degradation early.

Additionally, **alarm conditions** are defined based on process limits. For example:

- If a wafer is missing from a slot
- If a misalignment occurs repeatedly
- If motor current exceeds its set threshold

These alarms are automatically logged and displayed as notifications, color-coded by severity. This improves safety, operator awareness, and makes troubleshooting much faster.

3. Web and Mobile Accessibility

Modern SCADA platforms like Ignition provide built-in **mobile and browser access**, allowing the HMI to be viewed from any authorized device on the network—without needing to install special software.

For instance, with **Ignition Perspective or Vision modules**, the same dashboard can be accessed from:

- A control room PC
- A supervisor’s tablet
- A technician’s smartphone

This flexibility is especially valuable in semiconductor facilities where cleanroom restrictions limit movement. Operators can check system status or acknowledge alarms remotely, improving overall responsiveness and efficiency.

The system is designed with **secure access control**, ensuring that only authorized personnel can interact with critical functions or view sensitive data.

Communication and Networking Upgrades

In industrial semiconductor environments, data must flow seamlessly between embedded controllers, operator interfaces, Controllers, and higher-level management systems. Communication isn’t just about transmitting values—it’s about ensuring that the system can **interact reliably, in real time, and in standard formats** with the rest of the factory or enterprise infrastructure.

To meet this requirement, the upgraded wafer alignment system has been enhanced with **multi-protocol communication support**, using the STM32F407VG’s built-in hardware peripherals. This enables the system to interface with **field devices, SCADA systems, and cloud-based IIoT platforms**, ensuring both horizontal and vertical integration across automation layers.

1. Support for Industrial Communication Protocols

The STM32 platform includes multiple serial interfaces (UART, SPI, I2C) and an on-chip Ethernet MAC, which makes it well-suited for supporting a wide range of industrial protocols. These include:

- **Modbus RTU over UART/RS-485**

  Commonly used in legacy and compact industrial systems. The UART peripheral on STM32 can be connected to an RS-485 transceiver for long-distance, noise-resistant

communication.
Useful for connecting with older Controllers or HMI panels.

- **Modbus TCP over Ethernet**
  This protocol runs over IP and can be used to exchange data with SCADA servers or modern controllers. The STM32F407 supports Ethernet communication using an external PHY and lightweight TCP/IP stack (e.g., LwIP).

- **CANbus / CANopen**
  CAN is ideal for real-time, peer-to-peer communication among multiple controllers. The STM32 has a dedicated CAN controller that can be configured to run proprietary or standardized protocols like CANopen.

- **OPCUA**
  OPC UA is a platform-independent industrial protocol widely used in smart factories. It supports encrypted communication and both client-server and publish-subscribe models. STM32 can act as an OPC UA node using embedded stacks and is capable of sharing sensor or process data directly with higher-level systems.

- **MQTT**
  MQTT is a lightweight messaging protocol optimized for IoT. Using STM32's TCP/IP stack, the system can publish real-time wafer counts, alignment errors, or system health data to cloud platforms like **AWS IoT**, **Azure IoT Hub**, or **ThingsBoard**.

2. Real-World Integration Scenarios

With these communication capabilities, the STM32-based wafer alignment system can function as an intelligent node in a variety of architectures:

- **Factory Floor Integration**
  The system can connect to a **Controller** via **Modbus RTU or TCP**, sending live wafer data, alignment feedback, and system alarms. This allows full integration with existing automation lines.
- **Edge-to-Cloud Data Streaming**
  Through **MQTT**, sensor values and event logs can be published to a cloud broker. From there, data can be visualized, analyzed, or archived for traceability and performance monitoring.

- **SCADA/Monitoring System Compatibility**
  Using **OPC UA**, the system can directly share structured data with SCADA systems like Ignition, Siemens WinCC, or open-source tools. This reduces the need for middleware or additional protocol converters.
- **Multi-Device Coordination**
  If deployed in multi-station environments, several STM32 controllers can communicate over **CANbus** to synchronize motor control, status sharing, or process handoff between stations.

Reliability and Fault Tolerance

In semiconductor automation, where continuous operation and precision are critical, system stability cannot be compromised. Even short disruptions caused by software errors, voltage fluctuations, or electrical noise can result in missed wafer alignment, inaccurate counts, or costly downtime. Therefore, this project incorporates a comprehensive set of features to enhance **reliability**, **fault detection**, and **system protection**.

1. Watchdog Timers and Firmware Integrity Checks

To guard against software-related faults such as infinite loops or task failures, the STM32F407VG's built-in **watchdog timers** are enabled. Both the **independent watchdog (IWDG)** and **window watchdog (WWDG)** are configured to monitor the system:

- If the application fails to "refresh" the watchdog within a specified time window—typically due to a crash or unexpected delay—the watchdog automatically resets the microcontroller.
- This ensures the system can recover autonomously without user intervention, preventing extended downtime.

In addition, the **CRC (Cyclic Redundancy Check) unit** is used during system startup to verify the integrity of the firmware stored in flash memory. This prevents the system from running corrupted code after a failed update or memory error.

2. Hardware-Level Protections for Electrical Robustness

Given that the system interfaces with external sensors, actuators, and power supplies, it is exposed to various electrical disturbances. To ensure robustness against these factors, several hardware protections are integrated:

- **Opto-isolators** are used on key input/output lines to provide electrical isolation between the microcontroller and external components. This protects the MCU from voltage spikes and ground loop issues, which are common in industrial settings.
- **TVS (Transient Voltage Suppression) diodes** are placed on power inputs and communication lines to absorb surge energy caused by ESD or switching noise.

- **Brown-out detection** is enabled within the STM32 to detect supply voltage drops. If the voltage dips below a safe operating threshold, the microcontroller is automatically reset to avoid unpredictable behavior.
- All critical components—such as connectors, resistors, and relays—are selected with industrial-grade specifications to withstand temperature variations, vibration, and long-term wear.

These protections collectively ensure that the system can continue functioning reliably in electrically noisy environments typical of semiconductor fabs.

3. Runtime Fault Detection and Safe Response

Apart from electrical faults, the system must be able to detect **functional errors** during runtime and respond appropriately. Several software-based self-check mechanisms are implemented:

- **Sensor Plausibility Monitoring**: The system continuously checks whether sensor values fall within expected operating ranges. For instance, a stuck or disconnected sensor will be flagged if its reading doesn't change over time.
- **Motor Stall Detection**: By comparing the commanded motion with encoder feedback, the system can identify if the motor is stuck or slipping. In such cases, the motor is halted and the operator is alerted.
- **Error Logging and Alerts**: Faults such as communication failures, misalignment, or power loss are logged in memory and reflected on the HMI with visual or audible alerts. This helps operators take corrective action quickly.

When a fault is detected, the system is programmed to transition into a **safe state**. This typically includes stopping all motion, disabling outputs, and notifying the user—either through the HMI or via an alarm signal.

Edge-to-Cloud Integration (Industry 4.0)

As semiconductor manufacturing continues to adopt smart factory principles, the ability of embedded systems to connect with cloud platforms and perform intelligent, decentralized processing becomes increasingly important. This project incorporates several features that enable seamless **edge-to-cloud integration**, aligning it with the core goals of **Industry 4.0**: interoperability, real-time visibility, predictive analytics, and remote accessibility.

1. Real-Time Cloud Connectivity

The upgraded system is designed to transmit real-time data—such as wafer counts, alignment status, sensor readings, and error flags—to external monitoring systems or cloud services. This is accomplished through protocols like **MQTT**, **HTTPS**, or **OPC UA**, which are supported by many IoT platforms.

To make this integration easier, **STMicroelectronics offers STM32Cube Expansion libraries** for platforms like **AWS IoT Core** and **Azure IoT Hub**. These libraries include built-in tools for secure authentication, TLS encryption, and device provisioning, which significantly reduce development time.

Once connected to the cloud, the system can:

- Send live data for visualization or alerting
- Receive configuration commands or firmware updates
- Participate in centralized monitoring systems shared across multiple production lines

This connectivity provides **remote visibility and control**, a key expectation in modern industrial environments.

2. On-Device Processing with Edge Computing

While cloud connectivity enables centralized management, it's often impractical to send all data to the cloud due to bandwidth, latency, or security concerns. To address this, the system supports **basic edge processing** directly on the STM32 or via a connected local gateway (e.g. Raspberry Pi, industrial PC).

At the edge, the system can:

- Filter raw sensor signals (e.g., detect only rising edges or abnormalities)

- Perform basic calculations such as moving averages or signal thresholds
- Flag early signs of failure (e.g., drift in alignment accuracy or unusual motor behavior)

For more advanced use cases, the STM32 can forward selected data to an **edge gateway running platforms like AWS Greengrass**, which allows local execution of cloud-based logic or even **machine learning inference models**.

This approach minimizes data transfer, improves reaction time, and enables **autonomous decision-making** at the machine level.

3. Predictive Maintenance and Data Analytics

By continuously logging key parameters—such as motor current, alignment accuracy, sensor voltage fluctuations, or fault event frequency—the system builds a historical record of its operational health. This data can be uploaded to cloud storage or analyzed locally to detect patterns over time.

With enough data, **predictive maintenance models** can be implemented. For example:

- A gradual increase in motor load may indicate mechanical wear
- Frequent misalignment corrections could point to a sensor drift or positioning issue
- Repeated undervoltage detections might suggest an unstable power supply

Using these insights, maintenance can be scheduled **before** a failure occurs, avoiding unplanned downtime and increasing overall equipment effectiveness (OEE).

Such analytics are typically run on cloud platforms like:

- **AWS IoT Analytics**
- **Azure Machine Learning**
- **ThingsBoard with Influx DB and Grafana**

The STM32 functions as a **data collection endpoint**, sending only relevant summaries or alerts to the cloud, while more sensitive or frequent processing happens locally.

Total estimated for this project

Category	Component	Qty	Approx. Cost (INR)	Total (INR)
Controller	STM32F407VG MCU (Dev board like STM32F4 Discovery)	1	₹1,800 – ₹2,500	₹2,200
Sensor System	High-speed photodiodes (or phototransistors)	1	₹100 each	₹100
	Transimpedance amp circuit components	3	₹80 per channel	₹240
	Optional: Laser displacement sensor (Keyence-style)	1	₹12,000 – ₹18,000	₹15,000 (opt.)
	Optional: Camera module (basic vision)	1	₹800 – ₹1,200	₹1,000 (opt.)
Motor & Driver	Closed-loop stepper motor (NEMA17 with encoder)	1	₹2,500 – ₹3,500	₹3,000
	Stepper Driver (DRV8825 / TMC2209)	1	₹250 – ₹500	₹400
HMI & Display	3.5” or 5” TFT/IPS LCD (touch optional)	1	₹1,200 – ₹2,000	₹1,600
	(Optional) Industrial HMI/SCADA software license	1	₹5,000 – ₹15,000	₹7,500 (opt.)
Communication Modules	RS485 transceiver (Modbus RTU)	1	₹120 – ₹180	₹150
	CAN transceiver (e.g., MCP2551)	1	₹150 – ₹200	₹180
	Ethernet PHY module (LAN8720A)	1	₹250 – ₹400	₹300
Protection & Reliability	TVS diodes, opto-isolators, fuses, filtering caps	–	₹500 – ₹800 (total)	₹600

	5V/12V SMPS or Regulated Power Supply	1	₹500 – ₹700	₹600
PCB and Assembly	Custom PCB or Breadboard setup	–	₹300 – ₹700	₹500

Total Estimated Cost (Without Optional Vision/SCADA)

```
= ₹2,200 + ₹300 + ₹240 + ₹3,000 + ₹400 + ₹1,600 + ₹150 + ₹180 + ₹300 + ₹600 + ₹600 + ₹500
```

= **₹10,070 INR**

With Optional Laser Sensor and SCADA/Camera Upgrade

```
= Estimate + ₹15,000 (laser) + ₹1,000 (camera) + ₹7,500 (SCADA License)
```

Total with optional upgrades: ₹33,570 INR

Conclusion

The development of this upgraded wafer alignment and counting system marks a significant evolution from a basic prototype to a robust, scalable solution that is in line with modern industrial standards. By shifting from a Raspberry Pi–based setup to a dedicated **STM32F407VG microcontroller**, the system gains the ability to perform **deterministic, real-time control**—a key requirement in high-speed semiconductor environments.

The integration of **FreeRTOS** allows precise scheduling of time-critical tasks such as motor control, sensor acquisition, and communication, all without OS-level delays or interruptions. Combined with **closed-loop motor control using PID algorithms** and high-speed sensing (via photodiodes or advanced optical systems), the platform ensures **accurate wafer positioning**, even under dynamic conditions or process variability.

From a usability standpoint, the transition to a **professional-grade HMI**—developed using PyQt5 or SCADA tools like Ignition—adds significant value. Features like live trending, real-time alarms, and remote access make it easier for operators and engineers to monitor and interact with the system. These visual tools turn raw process data into actionable insights.

In terms of connectivity, the system now supports standard industrial protocols such as **Modbus**, **CANbus**, **OPC UA**, and **MQTT**, enabling seamless integration with Controllers, SCADA systems, and cloud platforms. This not only improves interoperability but also positions the system as a future-proof component within larger automation networks.

On the reliability front, built-in protections such as **watchdog timers**, **CRC checks**, **opto-isolators**, **TVS diodes**, and **brown-out detection** ensure safe operation in harsh industrial environments. The system is designed to handle faults gracefully, with proper logging and error notifications that improve maintainability and reduce downtime.

Perhaps most importantly, the platform has been designed with **Industry 4.0 and IoT** in mind. It supports **edge processing**, **secure cloud communication**, and **predictive maintenance frameworks**—allowing for remote diagnostics, performance optimization, and data-driven decision-making. Whether connected to AWS, Azure, or local IoT dashboards, the system contributes real-time insights that enhance operational efficiency.

In conclusion, this project successfully demonstrates how a well-structured embedded system—centered around STM32 and supported by modern design principles—can evolve from a lab prototype into a **smart, industrial-ready wafer alignment solution**. It combines real-time embedded control, intelligent sensing, professional user interaction, and advanced networking in a compact and efficient package—ready to meet the demands of today's semiconductor industry and tomorrow's smart factories.

REFERENCE

RPI5 vs STM32 - https://fulljoin.tech/blog/rpi-or-stm

STM32F4O7VGMicrocontroller-https://www.st.com/en/microcontrollers-microprocessors/stm32f407vg.html

LDR vs Photoelectric sensor -

https://www.tutorialspoint.com/difference-between-photodiode-and-light-dependent-resistor-ldr

Photoelectric sensor -https://www.ia.omron.com/products/family/3130/download/catalog.html

Laser displacement sensor - https://www.keyence.co.in/products/measure/laser-1d/

CMOS image sensor -

https://www.st.com/en/imaging-and-photonics-solutions/cmos-image-sensors.html

BLDC, Servo, Stepper motor -

https://ecampusontario.pressbooks.pub/btec315biomedicalinstrumenttroubleshooting/chapter/troubleshooting-electronic-motors/and https://anaheimautomation.com/blog/post/top-3-motor-technologies-for-compact-machines-stepper-vs-servo-vs-bldc?srsltid=AfmBOorhfHz6vJ_PLEk4eO5YGAFyEV4MDIOj4fOUwhmnUWAyujMIe-ta

DRV8824 Driver - https://www.pololu.com/product/2133

STSPIN-https://www.st.com/content/st_com/en/campaigns/stspin-family-a-complete-ecosystem-asp-mcmotdri.html

SCADA IGNITION - https://inductiveautomation.com/scada-software/

Ignitionmobile module -

https://www.docs.inductiveautomation.com/docs/7.9/getting-started/ignition-modules/core-modules/mobile-module

STM Independent watchdog timer -

https://www.st.com/resource/en/product_training/STM32WB-WDG_TIMERS-Independent-Watchdog-IWDG.pdf

Opto Couplers - https://www.vishay.com/docs/83725/4n25.pdf

X-CUBE AZURE - https://www.st.com/resource/en/data_brief/x-cube-azure.pdf