Motor control architectures for humanoid robot joints

Introduction

Humanoid robot joints must deliver precise torque, smooth motion, and fast dynamic response within strict size, weight, power, and thermal constraints. With many distributed degrees of freedom, each joint becomes an intelligent control node combining real-time control, current and position feedback, power conversion, communication, protection, and local power management.

Joint requirements vary significantly across the robot. Compact auxiliary axes may operate below 50 W or below 10 Nm of torque. Medium joints usually range from tens to hundreds of watts. They can also have a torque from single digits to a few hundred Nm. Load-bearing joints can operate from several hundred watts to multikilowatt power levels. Their torque can exceed 500 Nm. Across these classes, designs are moving toward higher control-loop rates, richer feedback, deterministic communication, and stronger local functional safety mechanisms.

STMicroelectronics supports this evolution with scalable semiconductor building blocks across the actuator signal chain:

- STM32 real-time control
- Motor drivers
- GaN and MOSFET power stages
- Current sensing
- Protection
- Wired connectivity
- Safety
- Security
- Local power management

This application note helps the user to select the appropriate ST products for the architectures of humanoid robot joints. See Table 5 and Table 6.

Figure 1. Humanoid robot joint



1 General information

The devices STM32G4 series, STM32H5 series, and STM32H7 series have an Arm® Cortex® core.

arm

Note: Arm and Cortex are registered trademarks of Arm Limited (or its subsidiaries or affiliates) in the US and/or elsewhere.

The Arm word and logo are trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. All rights reserved.

Table 1. Reference documents

Document number	Title
[1]	*Motor control architectures for dexterous humanoid robot hands* (AN6570)

2 Glossary

Table 2. **Glossary**

Term	Meaning
ADC	Analogic digital converter
BLDC	Brushless DC motor
CAN FD	Controller area network flexible data-rate
DC	Direct current
EMI	Electromagnetic interference
ESD	Electrostatic discharge
FOC	Field-oriented control
FSM	Finite state machine
GaN	Gallium nitride
IMU	Inertial measurement unit
ISPU	Intelligent sensor processing unit
KPI	Key point indicator
MAC	Media access control
PCB	Printed circuit board
PMSM	Permanent magnet synchronous motor
PWM	Pulse width modulation
RDS(on)	Drain-to-source on-resistance
RMS	Root mean square
SBB	System building block
SGMII	Serial gigabit media-independent interface
STO	Safe torque off
TSN	Time-sensitive networking

3 Architecture of humanoid joint-control nodes

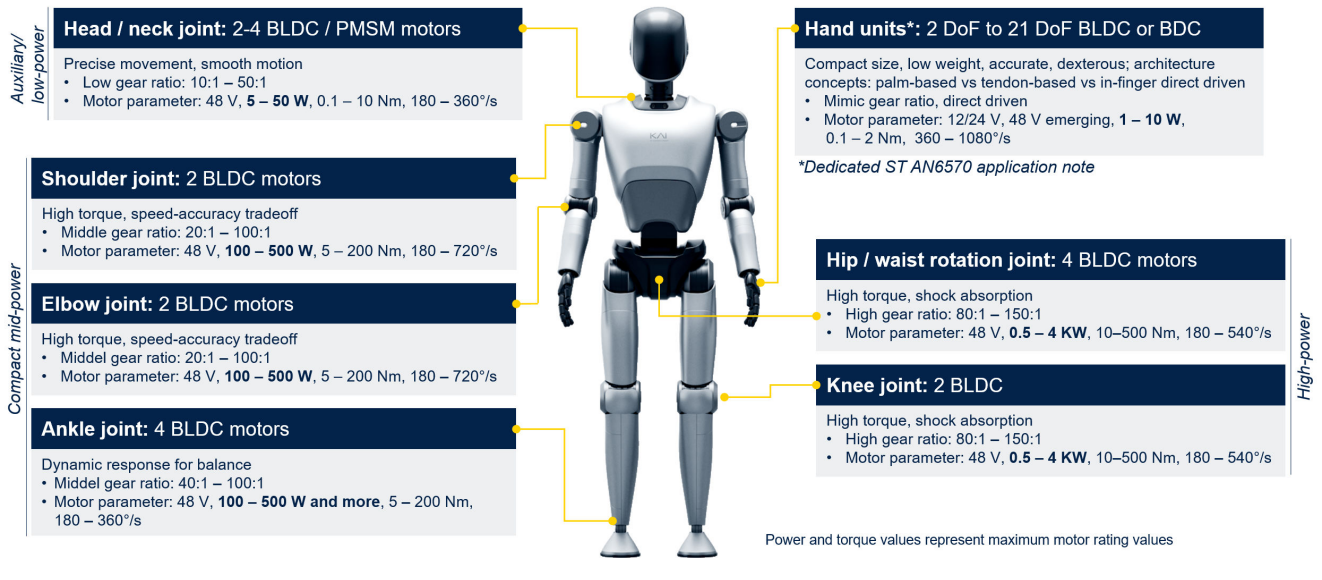
Joint performance depends on the complete control and power signal chain. The microcontroller (MCU) and other parts work together as one system. These parts include pulse width modulation (PWM) generation, current acquisition, position feedback, the power stage, communication, and local protection.

A complete joint-control node typically includes:

- A real-time MCU for local current, torque, speed, and position control
- Advanced PWM timers and PWM-synchronized ADC sampling
- One-, two-, or three-shunt current sensing
- One or two position-feedback interfaces, depending on the mechanism
- Motion sensing (for example, IMUs)
- Gate driver and integrated or discrete inverter stage
- Independent current, voltage, temperature protection, and communication-fault supervision
- CAN, EtherCAT or Ethernet TSN connectivity
- Regulated local power rails
- Nonvolatile memory for calibration and module identification

Figure 2 highlights typical actuator locations where motor-control architecture choices must be adapted to the local mechanical and power constraints.

Figure 2. Typical humanoid joint locations and motor control classes



Note: The values of Figure 2 are indicative of design envelopes derived from aggregated application requirements. Actual ratings depend on joint mechanics, motor selection, duty cycle, peak-torque profile, and thermal design.

The three joint classes, high, compact mid, and auxiliary or low power, require different power-stage integration levels, but they can share common control, sensing, communication, and software concepts. See Table 3 for detailed humanoid joint classes.

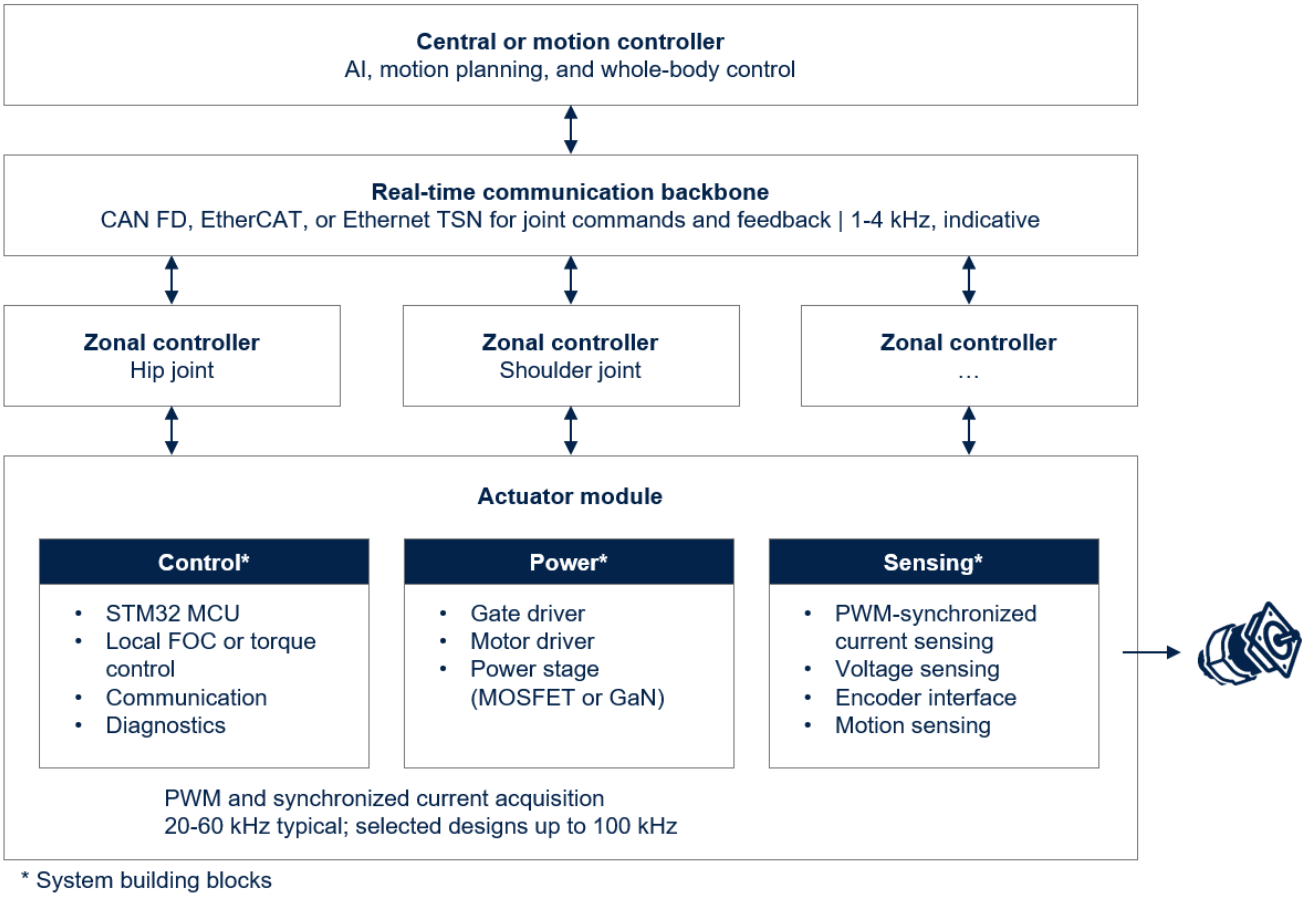
For palm-based, tendon-based and in-finger hand architectures, see the dedicated application note [1] for dexterous humanoid robot hands.

Table 3. Humanoid joint classes and architecture directions for motor control

Joint class	High-power joints	Compact midpower joints	Auxiliary/low-power joints
Typical locations	Hip, knee, ankle, torso, and high-load shoulder	Elbow, wrist, load-bearing neck rotation/tilt axis	Low-load neck axis, head adjustment, small local servo modules
Motor types	Frameless BLDC motors		
Indicative electrical envelope	• 48-60 V nominal voltage • 500 W to several kilowatts • May exceed 100 A peak	• 48-60 V nominal voltage • 50 - 500 W • Often 5-20 A, with higher-current variants	• 48-60 V • Below 50 W • Current depends on motor type and location
Main design priority	• Current capability • Thermal margin • Fast fault reaction	• Power density • Efficiency • Control response	• Compact integration • Quiet motion • Reduced wiring
Suitable ST architecture direction	STM32 motor-control MCU + STDRIVE gate driver + discrete MOSFET or GaN power stage	STM32 motor-control MCU + compact gate driver and MOSFET or integrated GaN motor driver such as GaNSPIN	Compact STM32 motor control MCU + integrated MOSFET drivers such as STSPIN32 + discrete MOSFET power stage and integrated sensing and communication

Time-critical current and torque loops should remain local to the joint. The joint-control node receives position, velocity, or torque commands, closes the local control loops, supervises critical faults, and reports status and diagnostic data to the arm, leg, or zonal controller. This reduces control latency and avoids transmitting raw, high-rate control data across the robot network.

Figure 3. Distributed control architecture with zonal nodes and motion coordination



DT80462V1

Humanoid joints perform best when real-time motor control, sensing, power management, signal conditioning, communication, and local protection are designed to work together. The design choices must satisfy the needs for current torque control, thermal management, and communication. Furthermore, there must be space for quicker control loops. The design must enable enhanced feedback, reliable networking, and improved local safety measures.

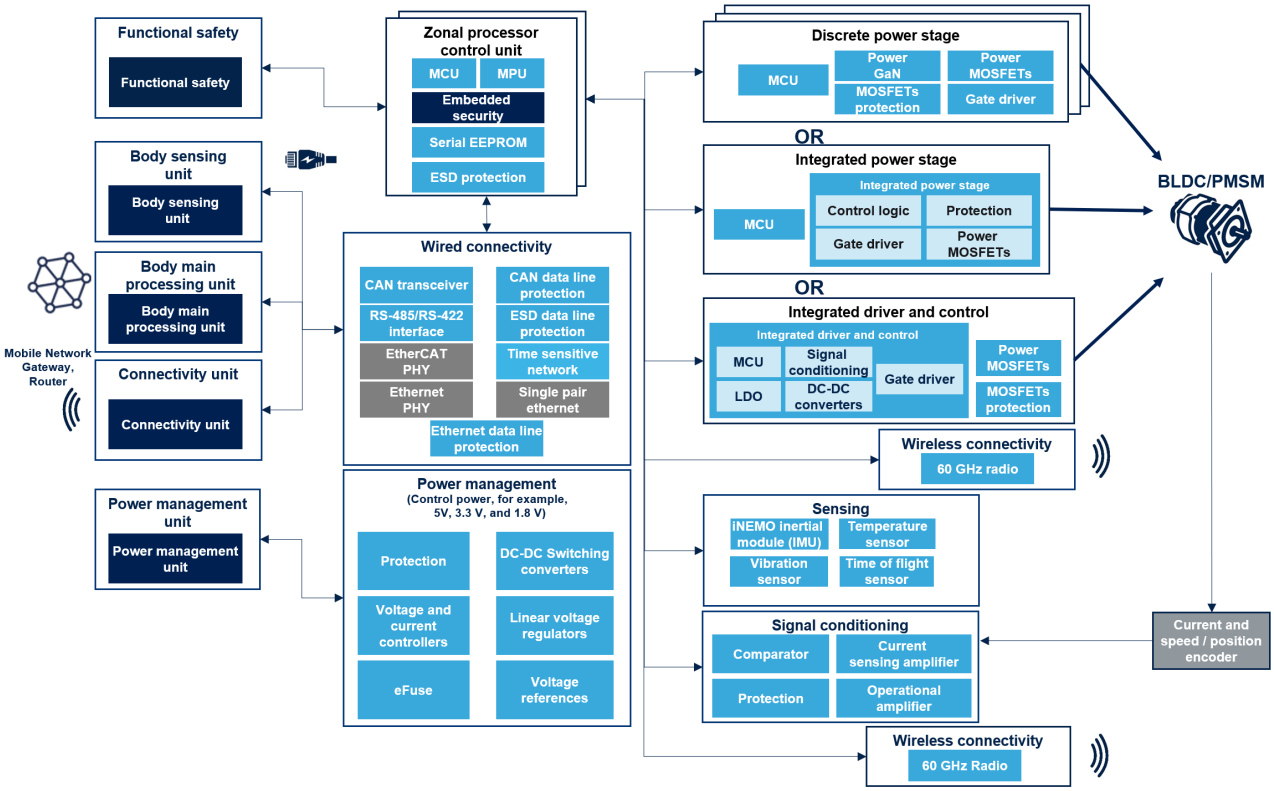
Table 4. System-level design priorities for humanoid motor drives

Design decision	Why it matters in humanoid robots	Architecture implication
PWM and current acquisition	Torque response and smooth motion depend on deterministic control and low-latency current feedback.	Select the MCU, PWM timers, ADCs, current-sensing topology, and power stage as one timing chain.
Position feedback	Position-sensor timing, resolution, and location affect commutation, joint-angle accuracy, and diagnostics. Motion sensing tracks whether movement is completed as expected or deviates from the original path.	Use one motor-side sensor where sufficient. Add output-side or redundant feedback where the mechanism or safety concept requires it.
Power-stage integration	Voltage, current, switching loss, cooling, and available volume determine the optimum implementation.	Use integrated stages for compact low-current designs, semi-discrete stages for intermediate requirements, and discrete devices for high-current joints.
Communication and local fault reaction	Distributed joints require coordinated commands, but they must not rely on the central controller for immediate protection.	Use CAN FD for local actuator connectivity and EtherCAT, or Ethernet TSN where deterministic synchronization is required. Detect local faults and safely shut down the power stage in hardware.

4 Designing for next-generation joint requirements

A joint-control node must be designed as one signal chain. MCU performance, ADC timing, position feedback, inverter switching, communication, and local protection all work together to determine torque accuracy, dynamic response, and fault response behavior.

Figure 4. Example of an actuator system architecture



4.1 PWM, current control, and synchronized acquisition

Precise torque control requires deterministic field-oriented control (FOC) execution and pulse-width modulation (PWM)-synchronized current acquisition. PWM frequency and current-loop update rate are related, but not necessarily identical. Both depend on the motor dynamics, sensing latency, switching loss, and the required control bandwidth.

Three-shunt sensing provides broad phase-current observability. One-shunt and two-shunt implementations reduce area and cost, but they require current reconstruction and impose tighter sampling windows. Inline current sensing can simplify algorithms by offering a highly accurate and continuous reading of phase currents. Regardless of topology, shunt value, amplifier settling, analog-to-digital converter (ADC) timing, and printed circuit board (PCB) layout must be evaluated as one measurement chain. Low-side current sensing is a cost-effective alternative that performs well in many scenarios. It provides an efficient way to monitor current, but it requires complex algorithms to estimate phase current accurately.

Current designs often operate in the 20 kHz to 60 kHz range. Some compact or high-dynamic next-generation joints operate in the 40 kHz to 100 kHz range, depending on application key performance indicators (KPIs) and the tradeoff between increased power density and higher switching losses for the selected power-device class.

4.2 Position feedback and calibration

One motor-side position sensor is sufficient for commutation and servo control in many joints. An output-side or second encoder can improve joint-angle accuracy, monitor transmission compliance or backlash, and provide diagnostic diversity. Persistent memory can retain encoder offsets, motor parameters, and joint calibration. It retains this information across power cycles or when replacing modules.

A joint-mounted inertial measurement unit (IMU) can complement encoder feedback by independently observing actual joint motion and orientation. IMU measurements can verify that commanded motion produces the expected physical response. They work well with encoder data, sensor fusion, or plausibility algorithms. This process can also detect faults in the encoder, in the mechanical transmission, or in the control system. In safety-related applications, this additional sensing may support motion plausibility checking and diagnostic diversity, subject to the required independence, diagnostic coverage, and system-level safety concept.

4.3 Power-stage integration and technology selection

Power-stage selection should consider several factors, including:

- DC bus voltage
- Continuous and peak current
- Switching frequency
- Available cooling
- Mechanical volume

Integrated stages suit compact lower-current joints. Semi-discrete stages add current and thermal flexibility. High-power joints generally require external switching devices to handle higher current and improve heat spreading.

Silicon MOSFETs remain cost-effective for many 48 V to 60 V drives. GaN is attractive where reduced switching loss, higher PWM frequency, or smaller power-stage volume and compact thermal design creates system value. The choice must also consider the following:

- Conduction loss
- RDS(on)
- Electromagnetic interference (EMI)
- Layout
- Thermal design
- Reliability and cost

4.4 High-speed communication and system integration

The CAN FD offers a compact and robust interface for local actuator communication. EtherCAT and Ethernet time-sensitive networking (TSN) serve applications that need tighter synchronization. They also provide higher bandwidth and integration with the robot backbone. Interface selection should follow:

- Required command-update rate
- Synchronization accuracy
- Diagnostic traffic
- Topology
- Fault-containment strategy

Where the MCU or processor and Ethernet PHY are separate, a serial gigabit media-independent interface (SGMII) can reduce the number of high-speed PCB connections compared with wider parallel interfaces. SGMII simplifies the media access control (MAC)-to-PHY connection but does not normally remove the external PHY or the physical network connection.

Sensor-rich and high-dynamic humanoid platforms can use concepts like the NVIDIA Holoscan Sensor Bridge architecture. This architecture helps to exchange synchronized sensor data. It also shares status and actuator data with the robot central computer or motion controller. This requires a high-bandwidth and low-latency path. It also needs to preserve the separation between local joint control and higher-level data processing. Current and torque loops should remain within the joint control perimeter.

Continuously rotating joints can use a hybrid slip-ring architecture that combines conventional power transfer with an ST60 short-range contactless data link. This separates power delivery from high-speed communication and can support gigabit Ethernet or other real-time industrial communication traffic across the rotating interface without relying on high-data-rate electrical slip-ring contacts. Local current and torque loops remain within the joint-control node. Use of the contactless link for commands, feedback, or synchronized data must be validated against the required latency, availability, and fault-response targets.

Trusted operation requires protection of the joint firmware, configuration, and communication throughout the actuator life cycle. Secure boot, authenticated firmware updates, protected calibration data and device, or module identity can help prevent unauthorized software, invalid parameters, or untrusted replacement modules from affecting joint behavior. These features require a system-level cybersecurity architecture and should only be claimed where the corresponding hardware, software, and key-management processes are implemented.

5 ST reference architectures for humanoid robot joints

Two complementary architectures, developed with partners, illustrate how the joint-control platform can scale across the robot. A compact 48 V GaN node addresses density-focused compact mid-power joints, while a discrete node provides the current and thermal scalability required by high-power joints. Both architectures use common control, sensing, communication and software concepts.

5.1 Compact 48 V GaN joint-control

This architecture targets compact midpower joints such as elbows, wrists, and selected neck or shoulder axes. An indicative application envelope includes a nominal 48 V bus. It ranges from approximately 50 W to 500 W. The current varies from about 5 A to 20 A. This variation depends on the current definition, duty cycle, and thermal design.

An STM32 motor-control MCU executes FOC, generates PWM, synchronizes current acquisition, processes position feedback, and manages communication and supervision. An integrated GaN power stage reduces inverter footprint and can support higher switching frequencies where these provide a system-level benefit. Its fast-switching edges can also increase dV/dt, voltage overshoot, EMI and motor-insulation stress, requiring coordinated design of the gate drive, PCB, motor interconnect, and winding insulation. CAN FD connects the local joint node, while EtherCAT or Ethernet TSN may be implemented at joint, gateway or zonal level.

Features and benefits:

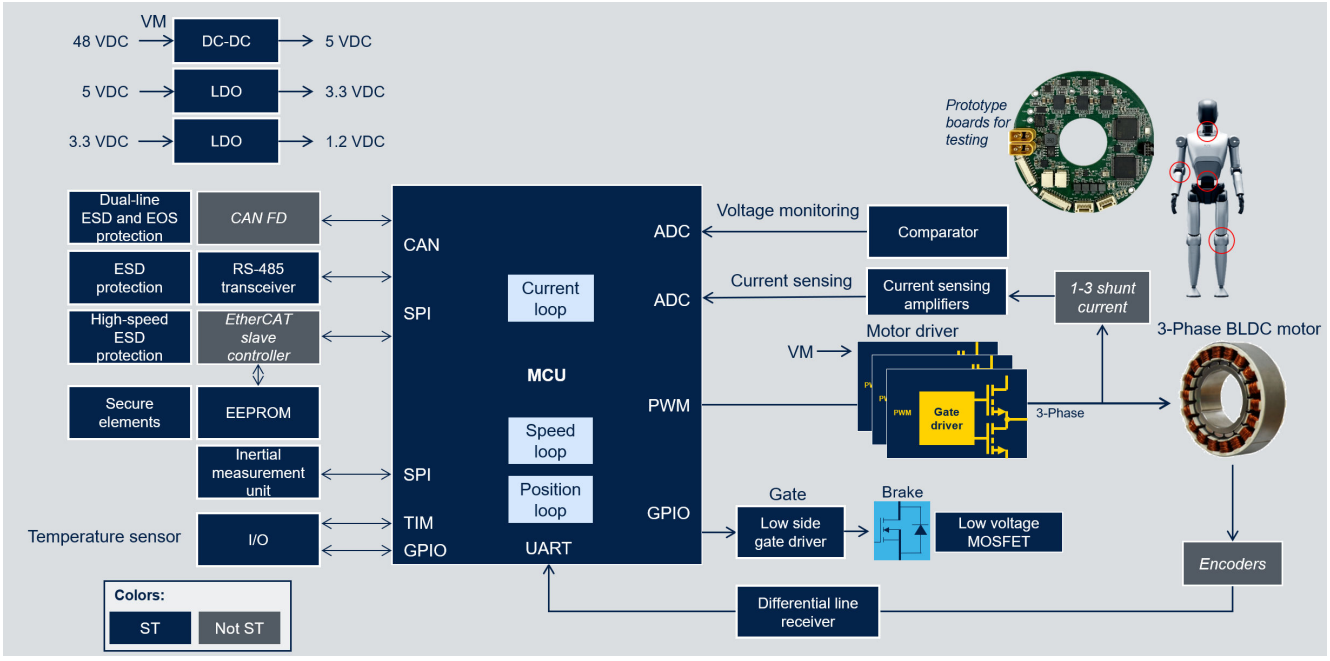
- Compact power stage for 48 V joints
- PWM-synchronized current and position acquisition for responsive torque control
- Reduced inverter footprint where GaN provides a system-level integration advantage
- Reusable STM32 control, communication, and software architecture across multiple joint variants

The compact node includes fast local power-stage protection and a hardware shutdown path. Where the application requires certified or advanced safety functions, the design may add independent monitoring or a dedicated safety controller according to the system-level safety concept.

Architecture validation platform

The architecture validation platform used to model the joints is illustrated in Figure 5.

Figure 5. Example of a compact 48 V-class GaN joint-control node



The list of ST products which are stated in the above figure are listed in Table 5.

Table 5. Selected ST products for compact 48 V GaN joint-control architecture

Function block	ST building block	Purpose
STM32 motor-control MCUs	STM32G4 series	Real-time FOC execution, multiaxis coordination, communication handling
STM32 high-performance MCUs	STM32H5 series STM32H7 series	Coordinates motor-control nodes, aggregates sensors and calibration data, manages diagnostics, and connects joints to a zonal controller
Motor driver (power stage)	GaNSPIN1102x(2) GaNSPIN1105(1) or silicon based drivers	Integrated GaN-based motor driver based on gate driver and GaNFETs for compact, efficient 48 V/100 V-class joint servo designs. Pin to pin compatible to scale up power level.
Current sensing	TSC202x TSC21x TSC240(1)	High-accuracy current measurement for torque control
Motion sensing	IIS3DWB10IS ISM330DHCXTR(1) ISM330IS LSM6DSV16X NMH1000	Robot joint IMU, vibration, or magnetic switch
Contactless rotating-interface communication	ST60 series	High-speed contactless data transfer across continuously rotating joints in a hybrid slip-ring architecture
Nonvolatile memory	M24C16-RMN6P(1) standard serial EEPROM	Stores motor parameters, encoder offsets, joint calibration, and module identity
Secure element	STSAFE-A120(1)	Device authentication, secure attachment, system integrity, including provisioning services
Voltage monitoring	TS3011(1) TS3121	Comparator/local protection for fast overcurrent and fault detection
Local power management	DCP10001Y(1) LD39015M12R(1) LD39100PU33R(1)	Compact DC-DC regulators and linear regulators for the MCU, gate driver, sensing, and communication sections
Break	STL125N10LF8AG(1) STripFET F7 and F8	100 V low voltage MOSFET in small QFN package to break due to inductive voltage spikes (back EMF), thermal overload from slow switching, or stalled mechanical loads
Low side gate driver	PM8834(1)	A flexible, high-frequency dual low-side driver for MOSFETs and IGBT in a small SO-8 package.

1. *Used in the application as shown in Figure 5.*
2. *In development, engineering samples are available.*

In a 48 V laboratory configuration, a package temperature of 63.3°C was measured without heatsink in the following operating conditions:

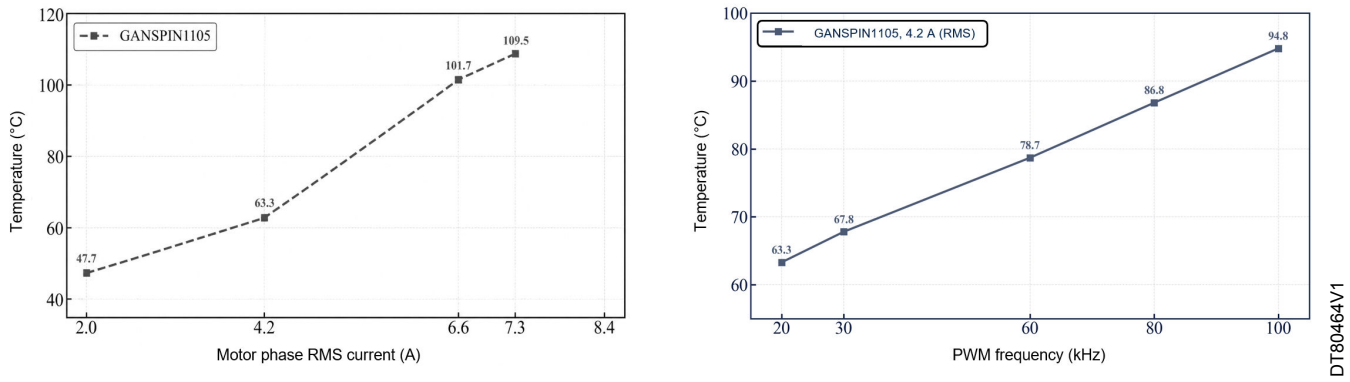
- 200 W
- 4.2 A RMS phase current
- 20 kHz PWM

This result demonstrates the feasibility at one operating point only. It is not a guaranteed product or system specification.

Thermal performance depends on several factors. These factors include PCB layout, ambient temperature, and airflow. It also depends on switching strategy, enclosure, cooling, and duty cycle.

For instance, the motor temperature rises in line with motor phase current and PWM frequency as illustrated in Figure 6.

Figure 6. **GANSPIN1105 measurements of temperature as a function of current and frequency**



5.2 High-current discrete joint-control

This architecture targets high-power joints such as hips, knees, ankles, and load-bearing shoulder axes. Many concepts use 48 V to 60 V buses with 80 V-class to 100 V-class switching devices. Depending on duty cycle and cooling, power can extend from approximately 500 W to several kilowatts, and demanding designs may require a peak current above 100 A.

The controller, gate driver, and external power devices are separate. This setup helps to optimize switching devices, current margin, cooling, and protection for each joint. One-shunt, two-shunt, or three-shunt measurements can be selected according to observability, accuracy, and PCB constraints. Motor-side and output-side position feedback can support commutation, joint-angle accuracy, and diagnostics.

Independent overcurrent, voltage, and temperature supervision provide a hardware power-stage shutdown path. CAN FD, EtherCAT, or Ethernet TSN connectivity is selected according to the robot network.

Features and benefits:

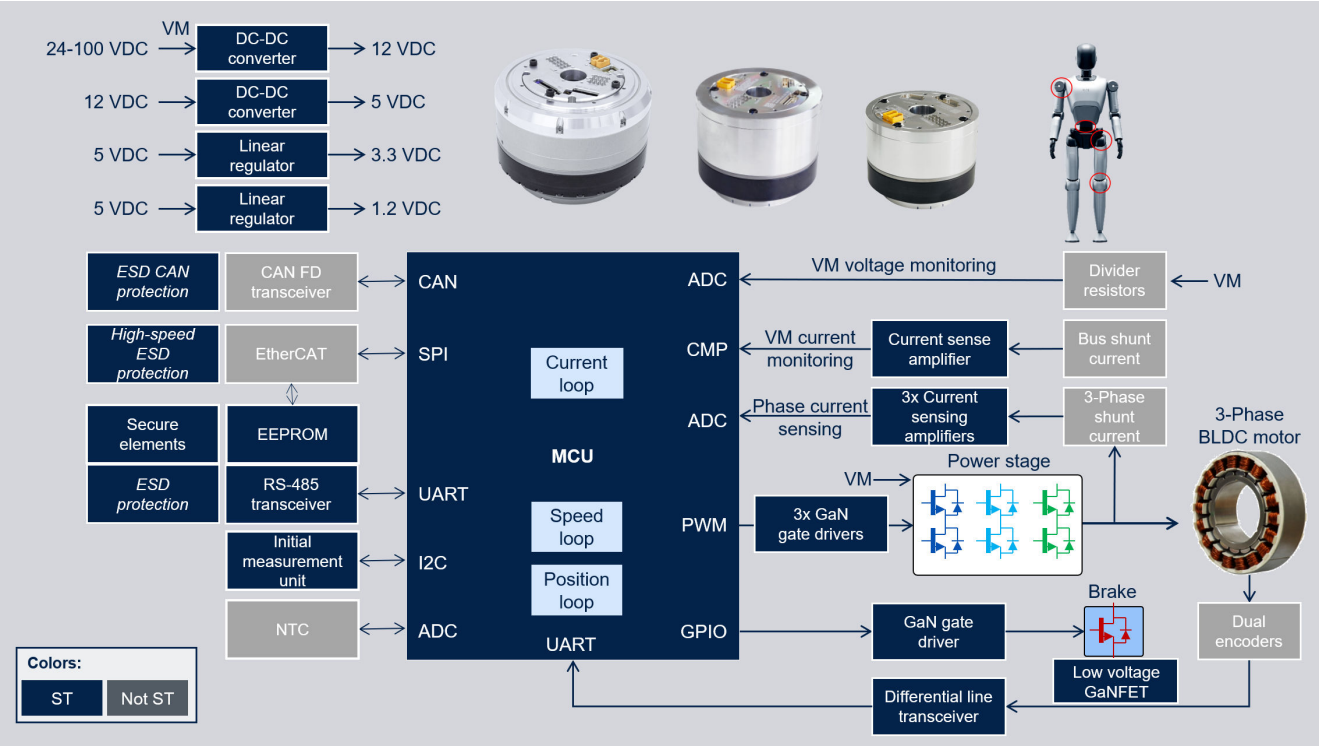
- Scalable external power stage for high continuous and peak current
- Flexible heat spreading and cooling implementation
- PWM-synchronized multichannel current acquisition
- Support for motor-side and output-side position feedback
- Local hardware fault reaction and deterministic communication
- Common STM32 control and software architecture across different joint power classes

The discrete implementation provides greater flexibility to separate motor control, power-stage protection, and safety-related monitoring. High-power joints may include a dedicated safety MCU. They can also have redundant feedback. Additionally, an independent gate-disable pathway may exist. This path supports safe torque off (STO) or safely limited functions. These features depend on the required safety integrity target and validation.

Higher loop rates, dual position feedback, richer diagnostics and deterministic networking increase MCU, memory and peripheral demands. A future-ready implementation should reserve sufficient timer, ADC, communication, and memory resources without implying support from an unreleased device.

An example of architecture used to model the joints is illustrated in Figure 7.

Figure 7. Example of an high-current discrete joint-control node



A high-current discrete joint-control node may include:

- An STM32 motor-control MCU for deterministic FOC, PWM generation, synchronized current acquisition, position feedback, and communication
- A three-phase gate-driver implementation matched to external MOSFET or GaN devices
- One-shunt, two-shunt, or three-shunt current measurement selected, using low-side or phase-current measurement according to accuracy, observability, and PCB constraints
- Independent overcurrent, bus-voltage, and temperature supervision with a hardware power-stage shutdown path
- Brake or torque-disable control is defined as a part of the system-level fault response.
- CAN FD, EtherCAT, or Ethernet TSN connectivity selected according to the robot network architecture

The list of ST products which are stated in Figure 7 are listed in Table 6.

Table 6. Selected ST products for high current discrete joint-control architecture

Function block	ST building block	Purpose
STM32 motor-control MCU	STM32G4 series(1)	Real-time FOC execution, multiaxes coordination, communication handling
STM32 high-performance MCUs	STM32H5 series STM32H7 series	Coordinates motor-control nodes, aggregates sensors and calibration data, manages diagnostics, and connects joints to a zonal controller
STDRIVE-GaN or MOSFET gate-driver	STDRIVEG212(1)	200 V high current half bridge gate driver with strong protection for 5 V GaN-HEMT transistors
Power stage (GaN or Si) and brake	SGT1D5R10MEA(1)	External 100 V-class GaN transistor for discrete 3-phase inverter with very low RDs(on)
Current sensing	TSC240A1IPT(1)	High-accuracy current measurement amplifier for motor current and phase current sensing
IMU, motion sensor	ISM330DHCXTR(1) ISM330IS LSM6DSV16X IIS3DWB10IS NMH1000	Robot joint IMU, vibration, or magnetic switch Control loop feedback, hinge angle detection, and measurement (see ST edge-ai-explained)
Transceiver (RS-485)	ST4E1240IQT(1)	Robust real-time communication with high electrostatic discharge (ESD) immunity
Contactless module interface	ST60 series	Short-range high-speed data transfer across detachable, rotating, or sealed joint interfaces
Nonvolatile memory	M24C16-RMN6P(1) standard serial EEPROM	Stores motor parameters, encoder offsets, joint calibration, and module identity
Secure element	STSAFE-A120(1)	Device authentication, secure attachment, system integrity, including provisioning services
Local power management	DCP10001Y(1) DCP3603(1) LD39015M12R(1) LD39100PU33R(1)	Compact and efficient DC-DC regulators, linear regulators to supply MCU, driver, EtherCAT, CAN FD section

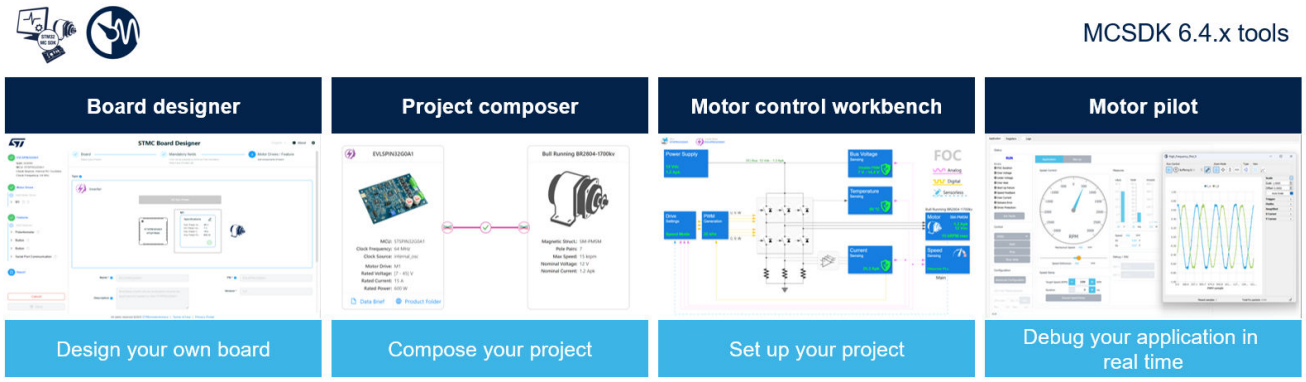
1. *Used in the application as shown in Figure 7.*

6 Software enablement for humanoid joint control

Robotic-joint developers use proprietary software, open motor-control platforms, and vendor-supported software packages. They choose these tools based on the existing architecture and the development stage.

The motor control software development kit (X-CUBE-MCSDK) is an ST motor-control software expansion package. In major robotic joints, its main value lies in the ability to run advanced control algorithms on high-performance STM32 microcontrollers. The enhanced processing capability, analog peripherals, and timing resources make it possible to implement sophisticated motor-control functions with high dynamic performance and tight control-loop execution. Some robotics developers use other platforms, such as ODrive. The appropriate approach depends on the existing software architecture, required feedback interfaces, performance targets, and validation needs.

Figure 8. Key components of X-CUBE-MCSDK version 6



From an algorithmic point of view, X-CUBE-MCSDK supports several control strategies depending on the motor type, sensing method, and application constraints. The package can implement six-step commutation, although this approach is generally less relevant for robotics, where smooth torque production and low-speed precision are usually required. The core strength of the package is the FOC, which can be deployed in both sensored and sensor-less configurations. Sensored FOC can use incremental encoders or Hall sensors. Sensor-less operation can rely on estimators. The estimators include the *Luenberger* observer (STO), and advanced approaches like High Sensitivity Observer (HSO), and ZeST. This flexibility allows designers to select the most appropriate control scheme for each joint architecture, balancing cost, performance, and mechanical integration constraints.

A key advantage of X-CUBE-MCSDK is that it is not a standalone tool. It is fully integrated into the ST ecosystem and works together with STM32CubeMX, which generates the project configuration and the underlying libraries. This significantly simplifies system integration with other ST products and helps streamline development from initial setup to final firmware generation. For robotic applications, motor control often needs to work with communication interfaces. It also interacts with sensing, safety logic, and higher-level motion layers. This ecosystem-based approach reduces integration effort and improves consistency across the entire embedded stack.

ST MEMS sensors can complement the motor-control software through embedded processing and STM32 motion-sensor libraries. Machine learning core, finite state machine (FSM), or intelligent sensor processing unit (ISPU)-enabled sensors can execute functions such as motion and shock detection, vibration monitoring, and activity recognition close to the sensing point. Additionally, STM32 software libraries can support sensor calibration, sensor fusion, and real-time motion detection. This processing can reduce bridge-MCU and network load while providing joint-motion information for state estimation, diagnostics, and transmission monitoring. These functions complement, rather than replace, encoder feedback; support must be verified for the selected sensor, MCU, and released software package.

The package also includes a motor-control protocol interface, which communicates with the MCU over UART and enables real-time debugging of the control algorithm. Through this interface, users can monitor live variables, observe internal control states, and tune parameters during operation. This is especially useful during the commissioning of robotic joints. It helps to optimize their performance. Tuning current loops, speed loops, and position control is often essential. This tuning achieves the required precision and smoothness.

X-CUBE-MCSDK provides fundamental motor-control functions. These functions include FOC, position detection, and standard control loops.

However, X-CUBE-MCSDK does not support absolute encoders yet, required by robotic systems that need immediate knowledge of their position after they turn on, or recover from faults. The package does not include advanced application-specific features. It lacks moment-of-inertia compensation, cogging torque compensation, and torque vectoring. Calibration management is also not included.

These capabilities must still be implemented at the application level by the user, depending on the mechanical design and performance targets of the robot.

From the safety perspective, X-CUBE-MCSDK includes a broad set of protections and fault-handling mechanisms that help improve operational robustness. However, it does not by itself provide certification-oriented safety frameworks such as class B or higher. For applications that require safety-compliant software infrastructure, ST offers dedicated safety packages, including X-CUBE-CLASSB, which provide sets of safety functions aligned with specific STM32 microcontrollers. This separation allows developers to combine high-performance motor control with the appropriate safety layer when the target application demands it.

Overall, X-CUBE-MCSDK is best understood as a powerful motor-control foundation for robotics, particularly when used with high-end STM32 devices. It provides control algorithms and integration tools. It also offers to debug capabilities for building advanced three-phase motor drives. The package leaves room for application-specific optimization, and implementation of higher-level robotic functions above the motor-control layer.

For major robotic joints, this makes it a strong enabler for precise, efficient, and scalable actuator design.

7 Conclusion

A humanoid joint-control node must combine several functions:

- Accurately control the motors
- Sense movement in sync with the control loop
- Convert power efficiently
- Communicate reliably
- React quickly to faults
- Fit within a limited space

The ST scalable architecture approach allows for high-power joints. It also supports compact mid-power joints and auxiliary low-power joints. All these joints can share common STM32 control, sensing, and software concepts. They use the power stage integration level that best suits each application and its design parameters.

Revision history

Table 7. **Document revision history**

Date	Version	Changes
13-Aug-2026	1	Initial release.

Contents

1 General information 2
2 Glossary 3
3 Architecture of humanoid joint-control nodes 4
4 Designing for next-generation joint requirements 7
4.1 PWM, current control, and synchronized acquisition 7
4.2 Position feedback and calibration 8
4.3 Power-stage integration and technology selection 8
4.4 High-speed communication and system integration 8
5 ST reference architectures for humanoid robot joints 10
5.1 Compact 48 V GaN joint-control 10
5.2 High-current discrete joint-control 12
6 Software enablement for humanoid joint control 15
7 Conclusion 17
Revision history 18
List of tables 20
List of figures 21

List of tables

Table 1. Reference documents . . . 2
Table 2. Glossary . . . 3
Table 3. Humanoid joint classes and architecture directions for motor control . . . 5
Table 4. System-level design priorities for humanoid motor drives . . . 6
Table 5. Selected ST products for compact 48 V GaN joint-control architecture . . . 11
Table 6. Selected ST products for high current discrete joint-control architecture . . . 14
Table 7. Document revision history . . . 18

List of figures

Figure 1. Humanoid robot joint . . . 1
Figure 2. Typical humanoid joint locations and motor control classes . . . 4
Figure 3. Distributed control architecture with zonal nodes and motion coordination . . . 5
Figure 4. Example of an actuator system architecture . . . 7
Figure 5. Example of a compact 48 V-class GaN joint-control node . . . 10
Figure 6. GANSPIN1105 measurements of temperature as a function of current and frequency . . . 12
Figure 7. Example of an high-current discrete joint-control node . . . 13
Figure 8. Key components of X-CUBE-MCSDK version 6 . . . 15

IMPORTANT NOTICE – READ CAREFULLY

STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice.

In the event of any conflict between the provisions of this document and the provisions of any contractual arrangement in force between the purchasers and ST, the provisions of such contractual arrangement shall prevail.

The purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST’s terms and conditions of sale in place at the time of order acknowledgment.

The purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of the purchasers’ products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

If the purchasers identify an ST product that meets their functional and performance requirements but that is not designated for the purchasers' market segment, the purchasers shall contact ST for more information.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, refer to www.st.com/trademarks. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2026 STMicroelectronics – All rights reserved