UNITREE

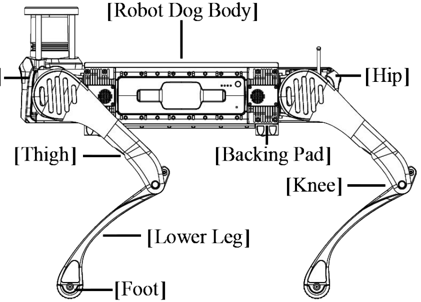
LOOKING GLASS XR

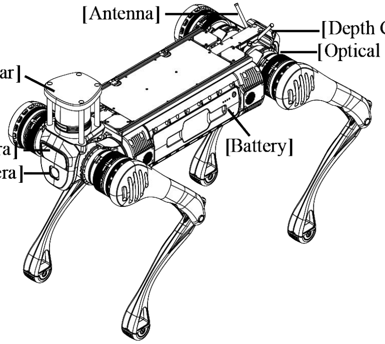
XR

WE MAKE REALITY...VIRTUAL.™

B2 Platform – Autonomy

B2 can be deployed in a semi-autonomous state for patrolling a large open area. The B2 robot is equipped with various advanced features and capabilities that make it suitable for such tasks.





Communication Capabilities: B2 can connect to cloud services for remote operations, system upgrades, and communication with the app and web front-end.

Sensor Integration: B2 collects sensor data such as motor and radar information, which can be crucial for navigation and obstacle detection in a large open area.

SLAM Services: B2 supports SLAM services for mapping, location, and navigation. This feature enables the robot to create maps of its environment, determine its position, and autonomously plan paths to navigate through the area.

Topology Navigation: B2 can utilize topology maps for navigation, allowing it to understand the layout of the environment and plan its movements accordingly.

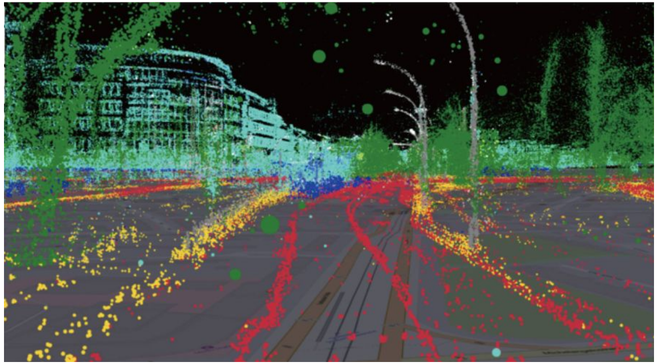
Obstacle Crossing Abilities: B2 has strong obstacle-crossing capabilities, including climbing stairs, which can be beneficial for patrolling different terrains in a large open area.

Autonomous Control: B2's WebRTC module facilitates data traffic with the app, enabling control commands to be issued remotely, supporting semi-autonomous operation.

Development Interfaces: B2 supports development interfaces like DDS and ROS2, providing flexibility for developers to customize and enhance its capabilities for unique tasks.

B2 Platform - SLAM

SLAM (Simultaneous Localization and Mapping) Services are divided into three main parts.



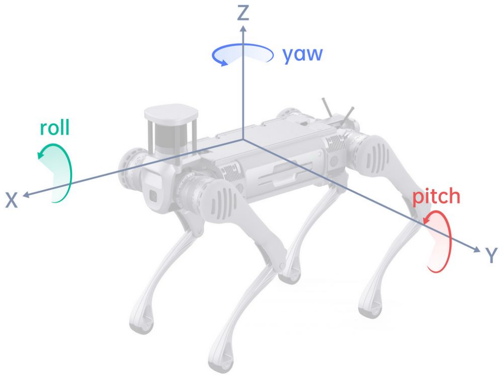
source

	Function	Illustrate
	Start navigation	After receiving relevant instructions, automatically open the navigation node
	Single node navigation	Receive a single navigation target node and automatically go to it
	Multi nodes loop navigation (default)	After receiving instructions, automatically proceed to all topology nodes in sequence
Navigation	Multi nodes loop navigation (setting)	After receiving instructions, automatically proceed to the received topology nodes in sequence
	Pause navigation	Pause the current navigation state and stop moving
	Restore navigation	Restore navigation status
	Return to the starting node	Return to the starting node position (default to the first node as the starting node)
	Close all nodes	After receiving instructions, close all functional nodes

1. **Mapping**: The mapping module is used to obtain feature information such as corner and face features in the current environment. It generates a current environment map for relocation. This process involves converting motion data into standard Odom and IMU data, obtaining point cloud data from LiDAR, loading marked navigation points, and controlling the robot to reach the designated target endpoint.
2. **Positioning**: The positioning module is based on the map generated by the mapping module. It combines the perception of the current sensor to determine the robot's pose in the current environment. The process involves obtaining motion data and LiDAR data, providing an initial pose for the robot, and matching perception data with the map to calculate the pose result. This provides position feedback and information for navigation.
3. **Navigation**: The navigation module automatically plans the path to the target point based on the positioning results and the deviation from the target point. It allows the robot to autonomously move to the target point by following the planned path. The process involves using the positioning results and marked topology points to guide the robot's motion towards the target point.

B2 Platform - Topology Navigation

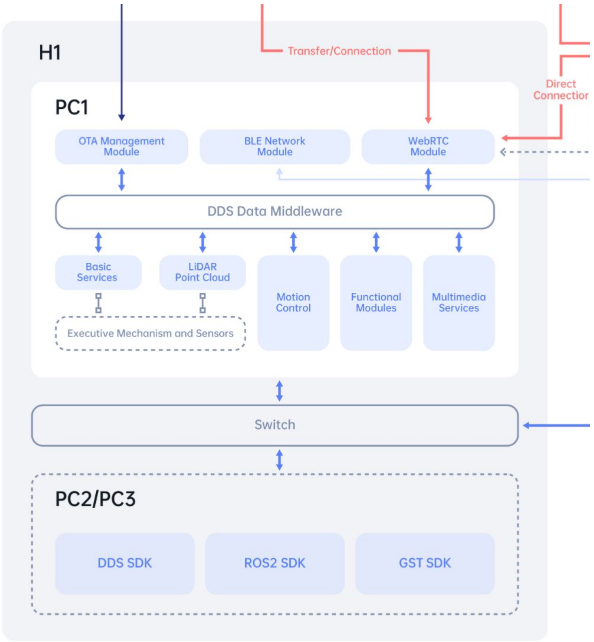
Topology Navigation involves creating a map that consists of **nodes** and **edges**. Nodes represent key locations in the environment, while edges represent the connections between nodes, indicating the paths that robots can move along.



1. **Establishing a Topology Map**: By creating a topology map, robots gain an understanding of the environment's layout and connection relationships. This enables them to conduct navigation and path planning effectively.
2. **Understanding Location Layout**: Robots use the topology map to comprehend the location layout and connection relationships in the environment. This understanding helps in navigating and planning paths efficiently.
3. **Navigation and Path Planning**: Robots select appropriate paths for movement based on the nodes and edges between their current position and the target position. This process involves autonomously planning motion based on the deviation between the robot's position and the target point, ensuring safe movement towards the target.
   Edges can be bidirectional or unidirectional, depending on navigation limitations and robot capabilities. Additionally, edges may have weights representing distance or cost, aiding in path planning and navigation decisions.
4. **Choosing Paths**: Robots can navigate to specific topology nodes, with the edge between the current position and the target node highlighted in red. This indicates that the robot is moving along the selected edge towards the target node. During movement, the robot's gait, speed, and other parameters are constrained by the corresponding edges.

Architectural Overview (H1 Example)

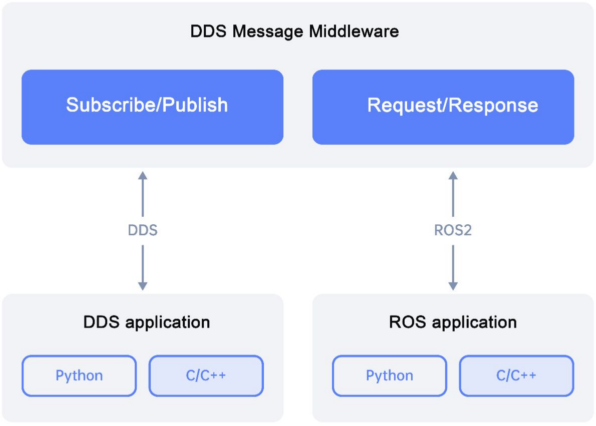
UNITREE



The **Unitree Explorer App** uses a WebRTC based module to facilitate key data exchange, managing streams for audio, video, radar data, motion status, and control commands, allowing for real-time interaction and feedback.

H1 includes three built-in computing units:

- PC1 is dedicated exclusively to Unitree's motion control programs and is restricted from public access.
- PC2 and PC3 are available for secondary development, allowing external developers to implement custom applications or modifications.

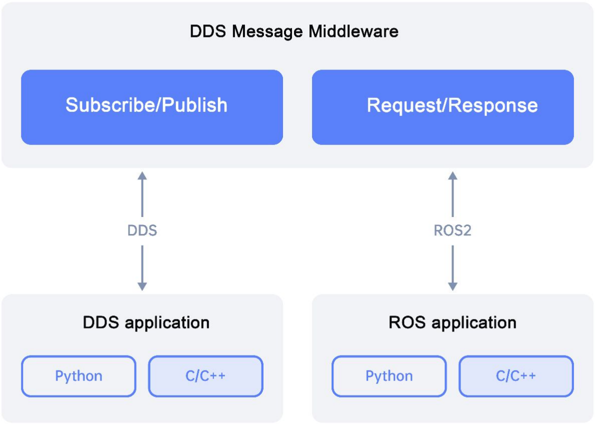


H1's data communication system uses **DDS** (Data Distribution Service) as the core middleware, handling data flows via two primary patterns: **Subscription/Publish** and **Request/Response**.

This approach balances continuous data flow needs with on-demand requests, improving system flexibility and usability.

DDS Middleware acts as the main communication layer between modules, compatible with ROS2 for extended development flexibility (developers must choose the appropriate RMW for ROS2 compatibility). EDU users can access interfaces via either DDS or ROS2.

Control SDK Overview



The Data communication system uses **DDS** (Data Distribution Service) as the core middleware, handling data flows via two primary patterns: **Subscription/Publish** and **Request/Response**.

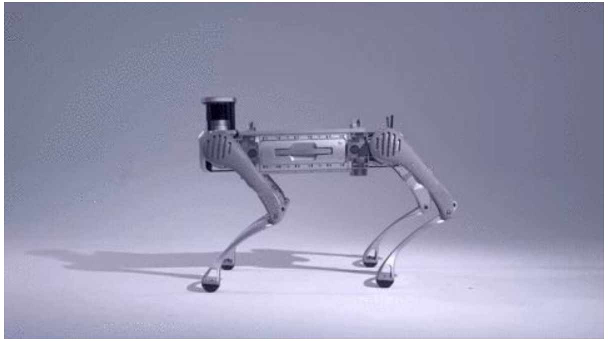
1. **Subscription/Publish**: Here, data is streamed from the sender to subscribers, fitting well with high-frequency, continuous updates. This pattern is ideal for scenarios where data needs to be constantly refreshed across multiple receivers.
2. **Request/Response**: Designed for on-demand data retrieval, this pattern supports lower-frequency, event-based interactions. It operates like a Q&A, suitable for specific data requests or functional changes.

For implementation, two main methods are outlined:

1. **API Call**: Resembling REST APIs, requests are broadcasted and paired with responses using unique identifiers (UUIDs) to maintain correct request-response matching.
2. **Functional Call**: A simplified API wrapper, offering function-like syntax for easier access and usability, especially for frequent interactions.

This approach balances continuous data flow needs with on-demand requests, improving system flexibility and usability.

Deploy Custom Code with DDS and ROS2



Unitree Development Resources

1. **Network Configuration**: Ensure that the devices are connected to the same local network. Modify the IP addresses of the devices if necessary to ensure proper communication.
2. **Accessing the Docking PC**: Connect to the docking PC using SSH with the default password. Select the ROS2 environment on the docking PC.
3. **Checking DDS Configuration**: Verify the network card parameters and the network card with the correct IP address on the docking PC. Check the ifconfig output to confirm the IP address.
4. **Checking Data Communication**: Use the ROS2 topic list command to view the message topics being sent by the robot's main PC. This confirms that data communication is functioning correctly.
5. **Custom Message Format**: Define custom message formats using IDL files. These files specify the structure of the messages exchanged between different components of the system. Examples include defining node information, edge information, and unique identifiers for instructions.
6. **Developing Custom Code**: Write custom code that subscribes to or publishes custom messages using the defined message formats. This code can interact with the SLAM and navigation services by sending or receiving relevant instructions and data.
7. **Compiling and Running Custom Code**: Compile the custom code using the necessary dependencies and tools provided by the Unitree ROS2 package. Run the custom code on the docking PC to interact with the SLAM and navigation services.

UNITREE

LOOKING GLASS XR

XR

WE MAKE REALITY...VIRTUAL.™