# LOW\_LEVEL-CONTROL\_NODE\_WITHOUT\_HANDS\_NODE – НИЗКОУРОВНЕВОЕ УПРАВЛЕНИЕ МОТОРАМИ РОБОТА UNITREE H1

Для работы с роботом Unitree H1 в рамках олимпиадных заданий предоставляется реализованный программный стек. Его основная задача — упростить взаимодействие с моторми робота, обеспечивая при этом высокий уровень надежности и безопасности.

**Важно:** Использование данного стека является рекомендуемым, но не обязательным. Участники вольны применять как предложенные инструменты, так и создавать собственные решения.

На GitHub представлена инструкция по установке нашего репозитория, управляющего моторами робота Unitree H1. Выполните всё до 6 шага не включая. В этой же инструкции ниже можно найти описание всех модулей включая указанный. Мы будем рассматривать только работу с нодой low\_level\_control\_without\_hands\_node, которая реализует безопасное управление моторами робота.

## Что делает этот узел (low\_level\_control\_without\_hands\_node)?

Этот Python-скрипт — низкоуровневый ROS 2-узел, который:

1. Принимает команды для движения моторов через специальный топик.

2. Обеспечивает плавное и безопасное управление положением моторов.

3. Ограничивает скорость движений, чтобы робот не дёргался резко.

4. Использует обратную связь от робота (его текущее положение суставов), чтобы корректировать команды.

## Основные компоненты

### 1. Топик для управления: positions\_to\_unitree

* Это входной топик, через который вы отправляете желаемые позиции суставов.
* Формат сообщения:

{"0": 0.5, "1": -0.3, ...}$0.8

– Левая часть — JSON-объект: ключи — номера суставов (см. таблицу ниже), значения — углы в радианах. Есть библиотека для python для работы с JSON-объектами (import python json). Которая позволяет “упаковать” словарь в JSON.

– Правая часть после $ — “impact” (влияние): число от 0.0 (робот не двигается) до 1.0 (полный контроль). (Присоединяется к уже “запакованному” словарю с помощью обычной конкатенации (сложения) строк)

Пример:

{"12": 1.0, "15": -0.5}$1.0

— поднять правое плечо и согнуть правый локоть. —

### 2. Нумерация суставов (важно!)

В коде есть словарь **FROM\_NAMES\_TO\_INDEXES**. Вот ключевые суставы:

|  |  |  |
| --- | --- | --- |
| Номер | Название сустава | Где находится |
| 12 | right\_shoulder\_pitch\_joint | Правое плечо (вверх/вниз) |
| 13 | right\_shoulder\_roll\_joint | Правое плечо (вперёд/назад) |
| 15 | right\_elbow\_joint | Правый локоть |
| 16–19 | Левые плечо и локоть | Аналогично правой руке |
| 0–2 | Правая нога (бедро, колено) | Нижняя часть тела |
| 3–5 | Левая нога |  |
| 6 | torso\_joint | Поворот корпуса |
| 9 | IMPACT | **Не сустав!** Управляет “влиянием” пользовательских команд |

В этом узле пальцы и запястья отключены (как указано в аннотации), хотя в словаре они есть. Управление идёт только по **active\_joints\_H1**.

Как управлять роботом? Пошагово

### Шаг 1: Запустите узел

В терминале:

Управление руками во время ходьбы (реальный робот):

ros2 run low\_level\_control low\_level\_control\_without\_hands\_node

# (по умолчанию target\_topic\_param = "arm\_sdk")

Полный контроль (реальный робот в development mode или MuJoCo):

ros2 run low\_level\_control low\_level\_control\_without\_hands\_node \

--ros-args \

-p target\_topic\_param:="lowcmd"

### Шаг 2: Отправьте команду через терминал

Откройте новый терминал и отправьте JSON-сообщение:

ros2 topic pub /positions\_to\_unitree std\_msgs/msg/String "data: '{\"12\": 0.8, \"15\": -1.0}\$1.0'"

Что это значит: - "12": 0.8 → правое плечо поднимается на 0.8 радиан (~45°) - "15": -1.0 → правый локоть сгибается - $1.0 → максимальное влияние (робот точно выполнит команду)

Совет: начинайте с малых углов (0.1–0.5 рад), чтобы не повредить робота!

### Шаг 3: Наблюдайте за движением

Робот плавно начнёт двигать рукой. Движение замедлено в начале (за счёт **TIME\_TO\_CHANGE\_VELOCITY** = 20 сек), но через 20 секунд станет быстрее.

Если отправить impact = 0.0, робот перестанет реагировать на команды и “отдаст контроль” обратно (например, системе балансировки).

Режимы управления Unitree H1: что важно знать

Управление роботом Unitree H1 зависит не только от того, работаете ли вы в симуляции или с реальным роботом, но и от режима работы самого робота. Есть два ключевых режима:

### 1. Обычный режим (preparation / sport)

* Робот стоит или ходит под управлением встроенного контроллера (например, с пульта).
* Вы можете управлять положением ТОЛЬКО верхней частью тела: руками и торсом.
* Для этого ваши команды должны публиковаться в топик arm\_sdk.
* Ноги остаются под контролем встроенной системы балансировки — вы их не трогаете.
* Режим разработчика (development mode) НЕ требуется.
* Подходит для безопасного взаимодействия: робот ходит, а вы управляете жестами рук.

Используйте **target\_topic\_param:=arm\_sdk** — это значение по умолчанию в коде.

### 2. Режим разработчика (development mode)

* Робот полностью отдаёт контроль внешней системе (вашему ROS-узлу).
* Вы можете управлять всеми суставами, включая ноги.
* В этом режиме робот обычно подвешен на страховке — ходить самостоятельно он не будет!
* Команды должны публиковаться в топик lowcmd.
* Обязательно включите development mode на роботе (см. Нашу документацию).
* Без этого режима публикация в lowcmd будет проигнорирована.

Используйте **target\_topic\_param:=lowcmd**, если работаете в этом режиме.

### 3. Симуляция в MuJoCo

* В симуляторе нет “обычного режима” — Вы всегда управляете всеми поторами робота напрямую.
* Поэтому работает ТОЛЬКО топик lowcmd.
* Топик arm\_sdk в MuJoCo не поддерживается и не будет иметь эффекта.

Запускайте узел с **target\_topic\_param:=lowcmd**.

### Итоговая сводка

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Сценарий** |  | **Топик для публикации** | **Требуется development mode?** | **Управление** |
| Реальный робот, ходьба + управление руками |  | arm\_sdk | ❌ Нет | Только руки + торс |
| Реальный робот, полный контроль (подвешен) |  | lowcmd | ✅ Да | Все суставы |
| Симуляция (MuJoCo) |  | lowcmd | ❌ (не применимо) | Все суставы |

## **Безопасность**

* Код автоматически ограничивает углы суставов (через limits из h1\_info\_library), так что вы не сможете выйти за физические пределы.
* При завершении (Ctrl+C) узел плавно снижает impact до 0, чтобы робот не “уронил” руки.

## Частые вопросы новичков

### Q: Где взять текущие значения углов?

Используйте ros2 topic echo /lowstate, чтобы посмотреть текущие углы суставов.

### Q: Почему робот не двигается?

* Проверьте, что impact > 0.
* Убедитесь, что робот включен и подключён.
* Проверьте, что номера суставов правильные (только из active\_joints\_H1).

### Q: Можно ли управлять ногами?

Да! Например: {"0": 0.2, "1": -0.3, "2": 0.5}$0.5 — немного пошевелить правой ногой. (Но в development-режиме или в симуляции MuJoCo)

## Итог

* Этот код — мост между вашими командами и железом робота.
* Вы говорите: «рука в позицию X», а узел безопасно и плавно это выполняет.
* Главное — помните: - Используйте номера суставов, а не названия. - Всегда указывайте impact (иначе ничего не произойдёт). - Начинайте с малых углов и низкого impact.

Удачи в управлении Unitree H1!