



Что такое LowCmd и LowState?

- **LowCmd** — это **сообщение, которое вы отправляете роботу**.
→ Вы говорите: «Мотор №3, повернись на 0.5 радиан с такой жёсткостью!»
- **LowState** — это **ответ от робота**, в котором он рассказывает:
→ «Вот мои текущие углы, скорости, температуры, данные с гироскопа и т.д.»

Эти сообщения работают на **низком уровне (low-level)** — вы управляете моторами напрямую, без абстракций вроде «идти вперёд».



Структура LowCmd — команда для робота

```
uint8[2] head
uint8 level_flag
uint8 frame_reserve
uint32[2] sn
uint32[2] version
uint16 bandwidth
MotorCmd[20] motor_cmd      → **Команды для 20 моторов** (главное поле!)
BmsCmd bms_cmd
uint8[40] wireless_remote
uint8[12] led
uint8[2] fan
uint8 gpio
uint32 reserve
uint32 crc                  → Контрольная сумма
```



Что внутри MotorCmd (один мотор)?

```
uint8 mode
float32 q      → Целевой угол (в радианах). Например: 0.0 = нейтраль, 1.0 ≈ 57°
float32 dq     → Целевая скорость вращения (рад/с). Обычно 0.
float32 tau    → Целевой момент (Н·м). Используется в режиме 1.
float32 kp     → "Жёсткость" — насколько сильно мотор сопротивляется отклонению от угла `q`
float32 kd     → "Амортизация" — насколько сильно мотор тормозит при движении
uint32[3] reserve
```



Пример:

Если вы хотите, чтобы мотор держал угол 0.3 рад и был "упругим", как пружина:

```
mode = 10
q = 0.3
```

```

dq = 0.0
kp = 50.0
kd = 2.0

```

📁 Структура LowState — состояние робота

```

uint8[2] head
uint8 level_flag
uint8 frame_reserve
uint32[2] sn
uint32[2] version
uint16 bandwidth
IMUState imu_state          → **Данные с гироскопа и акселерометра**
MotorState[20] motor_state  → **Состояние всех 20 моторов**
BmsState bms_state
int16[4] foot_force
int16[4] foot_force_est
uint32 tick
uint8[40] wireless_remote
uint8 bit_flag
float32 adc_reel
int8 temperature_ntc1/2
float32 power_v
float32 power_a
uint16[4] fan_frequency
uint32 reserve
uint32 crc

```

III Что внутри MotorState (один мотор)?

```

uint8 mode          → Текущий режим мотора
float32 q            → Текущий угол (рад)
float32 dq           → Текущая скорость (рад/с)
float32 ddq          → Ускорение (редко используется)
float32 tau_est      → Оценка момента, который сейчас приложен
float32 q_raw        → "Сырой" угол (до фильтрации)
float32 dq_raw       → "Сырая" скорость
int8 temperature     → Температура мотора (°C)
uint32 lost          → Сколько пакетов потеряно для этого мотора

```

🧭 Что внутри IMUState?

```

float32 quaternion[4] → Ориентация робота в пространстве: [w, x, y, z]
float32 gyroscope[3]  → Угловая скорость: [ωx, ωy, ωz] (рад/с)
float32 accelerometer[3] → Ускорение: [ax, ay, az] (м/с²)

```

```
float32[3] rpy  
int8 temperature
```

Как это использовать? (Кратко)

- Чтобы **управлять моторами** → публикуйте в топик `/lowcmd`, заполняя `motor_cmd[i]`.
 - Чтобы **читать состояние** → подписывайтесь на `/lowstate`, читайте `motor_state[i].q`, `imu_state.gyroscope` и т.д.
-