

??????????

Параметры робота задаются двумя способами, которые можно комбинировать:

- 1. Файл `params.json` в **текущем рабочем каталоге** программы.
- 2. Словарь, переданный в конструктор: `Robot({...})`.

Значения из словаря имеют приоритет над `params.json`. Удобная схема: общие параметры (адрес сервера, логирование) — в `params.json`, а тип устройства — в коде.

```
robot = Robot({"type": "LED"})    # остальное — из params.json
```

??????????

Параметр	Тип	По умолчанию	Описание
<code>name</code>	str	<code>"DefaultName"</code>	Имя устройства, отображается в интерфейсе оператора
<code>type</code>	str	<code>"DefaultType"</code>	Тип устройства. Определяет страницу управления в интерфейсе: <code>Robot</code> , <code>LED</code> , <code>Drone</code>
<code>server</code>	str	<code>"http://artex.arcon"</code>	Адрес сервера платформы (схема + хост, без порта)
<code>port</code>	int	<code>80</code>	Порт сервера
<code>telem_can</code>	bool	<code>false</code>	Признак «устройство передаёт телеметрию» — передаётся серверу в паспорте
<code>video_can</code>	bool	<code>false</code>	Признак «устройство передаёт видео» — передаётся серверу в паспорте
<code>max_send_freq</code>	float	<code>0.05</code>	Период (в секундах) между отсылками телеметрии и видео. <code>0.05</code> — до 20 отправок в секунду

Параметр	Тип	По умолчанию	Описание
<code>max_main_freq</code>	float	<code>0.1</code>	Период (в секундах) паузы основного цикла. <code>0.1</code> — около 10 итераций в секунду
<code>logger_type</code>	str	<code>"console"</code>	Куда писать журнал: <code>console</code> — в стандартный вывод, <code>file</code> — в файлы <code>all.log</code> и <code>error.log</code>
<code>logger_level</code>	str	<code>"all"</code>	Уровень журнала: <code>all</code> — все сообщения, <code>skip_TODO</code> — все, кроме служебных сообщений <code>TODO</code>
<code>external</code>	dict	<code>{}</code>	Зарезервировано для внешних модулей (в версии 0.1.1 не используется)

Несмотря на слово `freq` в названии, параметры `max_send_freq` и `max_main_freq` задают период в секундах, а не частоту. Чтобы отправлять данные 10 раз в секунду, укажите `0.1`.

??????? ? ?????????? ????????????

При подключении SDK передаёт серверу **паспорт** (`s_meta`) — параметры `name`, `type`, `telem_can`, `video_can`. Сервер использует его для отображения устройства и выбора интерфейса управления.

Состояние (`d_meta`) меняется во время работы и возвращается серверу по запросу heartbeat:

Поле	Значения	Смысл
<code>sid</code>	идентификатор или <code>null</code>	Идентификатор подключения
<code>ready</code>	<code>init</code> → <code>ok</code>	<code>ok</code> после сигнала готовности от сервера; до этого команды игнорируются
<code>telem_run</code>	bool	Запущен ли поток отправки телеметрии
<code>video_run</code>	bool	Запущен ли поток отправки видео

????? ???????

Сценарий	max_main_freq	max_send_freq
LED, простые датчики	0.1	0.1
Колёсный робот с камерой (FPV)	0.03-0.05	0.033-0.05
Слабый канал связи / мобильная сеть	0.1	0.1-0.2

Реальная частота отправки не выше частоты основного цикла: данные отправляются, только если основной цикл успел обновить их в Vault. Размер одного кадра 640×480 в JPEG (качество 80) — около 30–60 КБ; при 20 кадрах/с это 5–10 Мбит/с. Для слабых каналов уменьшайте разрешение кадра перед отправкой.

??????? params.json

Колёсный робот:

```
{
  "name": "tank-01",
  "type": "Robot",
  "server": "http://192.168.1.10",
  "port": 80,
  "telem_can": true,
  "video_can": true,
  "max_send_freq": 0.05,
  "max_main_freq": 0.05,
  "logger_type": "file",
  "logger_level": "skip_TODO"
}
```

Светодиодная лампа:

```
{
  "name": "lamp-hall",
  "type": "LED",
  "server": "http://192.168.1.10",
  "telem_can": true,
  "logger_level": "skip_TODO"
}
```