

RCSIM Deployment Tool — ilustrowana instrukcja użytkownika

Ilustrowany przewodnik użytkownika: karty konfiguracji, akcje wdrożenia i procedury krok po kroku. Zrzuty przedstawiają angielski interfejs.

Spis treści

1. Przeznaczenie narzędzia i zakres instrukcji	4
2. Uruchomienie i pierwszy przebieg pracy	4
Otwórz właściwy panel	4
Zalecana kolejność pierwszego wdrożenia	4
3. Połączenie — przygotowanie dostępu do Raspberry Pi	5
3.1. Adres, użytkownik i hasło	6
3.2. Prywatny klucz SSH	7
3.3. New SSH Password (Optional)	7
3.4. Scan Hardware (I2C/UART)	8
3.5. Zapis ustawień a wykonanie operacji	8
4. Ustawienia wdrożenia — platforma, sieć i wyposażenie	9
4.1. Project Source Path — co faktycznie wysyłasz	9
4.2. Application Platform Type — dwa różne wdrożenia	9
4.3. Sieć i protokół standardowego RCSIM	10
4.4. IMU, GPS, LiDAR i ELRS / MAVLink Serial	11
4.5. Kamera — port, sensor i profil obrazu	12
4.6. NTRIP RTK — dane korekcyjne dla zgodnego GPS	13
4.7. Ostateczna kontrola formularza	14
5. Przygotowanie i wdrożenie	14
Co uruchamiasz i gdzie powstaje instalacja	14
Lista przed rozpoczęciem	15
Dwa typy aplikacji — wybór zmienia cel instalacji	15
Co trafia do paczki	15
Pełne wdrożenie Docker — pierwsza instalacja	16
Fast Deploy oraz Docker Rebuild & Update	16
Wdrożenie i aktualizacja MCS	17
6. Aktualizacja, kontrola wyniku i odzyskiwanie	17
Hot Deploy Code — kiedy wystarcza	17

Aktualizacja tylko kamery	18
Kopia, restart i logi	18
Jak rozpoznać zakończenie	18
Błąd lub przerwanie	19
7. Usługi, logi i diagnostyka	19
Co oznaczają wskaźniki u góry	20
Test Connection — sprawdź SSH przed naprawą	20
Restart Service — restart programu	20
Reboot Pi — restart całego Raspberry Pi	21
App Logs i Build Logs	21
Diagnostics — zestaw odczytów systemowych	21
Konsola, postęp i zamknięcie narzędzia	22
8. Wideo i pozostałe narzędzia obsługowe	23
Video & Streaming — pola kamery	23
Detect CSI Cameras	24
Low Latency Profile	24
Cam Config Only — zastosuj ustawienia strumienia	24
Backup Docker Vol — co rzeczywiście obejmuje kopia	24
Aktualizacja a instalacja od nowa	25
9. Rozwiązywanie problemów i źródła	25

1. Przeznaczenie narzędzia i zakres instrukcji

Deployment Tool przygotowuje paczkę programu na komputerze operatora, łączy się przez SSH z Raspberry Pi i uruchamia wybraną procedurę instalacji lub aktualizacji. Pozwala też skonfigurować komunikację, sensory i kamerę standardowego RCSIM, odczytać logi oraz wykonać wybrane czynności serwisowe.

Rozróżniaj dwa cele: **Standard RCSIM (Docker & MediaMTX)** to oprogramowanie komputera pokładowego, a **RCSIM MCS (Direct Services)** to stacja operatora. Raspberry Pi z MCS nie staje się przez to platformą autonomii Tier 4. Wybór aplikacji musi odpowiadać katalogowi źródłowemu i roli docelowego hosta.

Obecny program otwiera panel przeglądarkowy. Jest to **jeden ekran z kartami**, a nie zestaw osobnych zakładek. Na szerokim ekranie po lewej znajdują się dane SSH, w środku konfiguracja, po prawej przyciski wdrażania i narzędzia; konsola jest na dole. Na wąskim ekranie karty układają się pionowo — przewijaj stronę, aby dotrzeć do kolejnych pól.

Ilustracje: rzeczywisty HTML, CSS i JavaScript projektu uruchomiono lokalnie 14 października 2026, w języku English, z publicznymi wartościami domyślnymi. Podgląd dokumentacyjny nie łączył się z Raspberry Pi i blokował akcje SSH oraz wdrażania. Status Offline/Off jest celowy; puste dane logowania oraz przykładowe porty nie są gotową konfiguracją Twojego zestawu. To zrzuty interfejsu, nie wygenerowane makiety. Powiększ PDF, aby odczytać drobne podpisy.

Sprawdzono kod i układ interfejsu. Nie wykonywano instalacji na urządzeniu, aktualizacji systemu, restartów Pi ani fizycznych testów kamery, radia i czujników. Ograniczenia widocznych opcji wyjaśniamy przy odpowiednich procedurach.

2. Uruchomienie i pierwszy przebieg pracy

Otwórz właściwy panel

Uruchom przygotowaną aplikację Deployment Tool na komputerze, z którego będziesz wysyłać źródła. Aktualny launcher otwiera przeglądarkę i lokalny serwer. Zaczyna od portu 5000 i szuka dostępnego portu w zakresie **5000–5009**. Użyj adresu faktycznie podanego przez program, np. `http://localhost:5000`.

Serwer domyślnie nasłuchuje na `127.0.0.1`, czyli komputerze z narzędziem. To nie panel MCS na porcie 8080 i nie adres Raspberry Pi. Jeżeli karta przeglądarki została zamknięta, działający lokalny serwer można ponownie otworzyć pod jego adresem. Jeśli sam serwer został wyłączony, uruchom aplikację ponownie.

W przygotowanym środowisku źródłowym punktem wejścia jest `RCSimRPi5deploymentapp.py` w katalogu `RCSIM_deployment_tool/RCSIM_deployment_tool`. Nie uruchamiaj równocześnie kilku wdrożeń do tego samego hosta. Przed instalacją przygotuj dostęp SSH i `sudo` na Pi, stabilne zasilanie, dostęp do wymaganych pakietów oraz kopię potrzebnej konfiguracji.

Lista **English / Polski** w nagłówku zmienia język. Ta instrukcja zachowuje angielskie nazwy z ilustracji i objaśnia je po polsku. Niektóre dodatkowe pola występują w starszym interfejsie desktopowym; nie szukaj ich automatycznie w obecnej wersji Web.

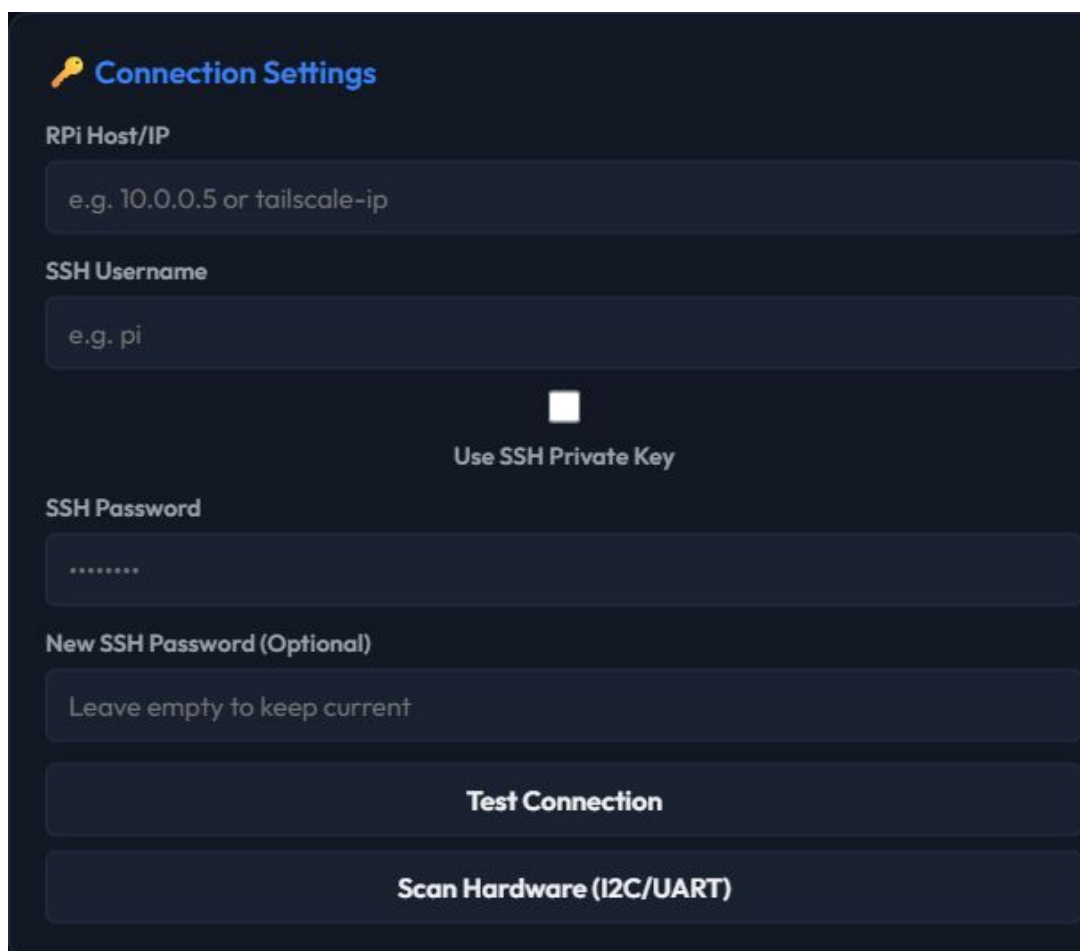
Zalecana kolejność pierwszego wdrożenia

1. Zatrzymaj model i zabezpiecz napęd. Ustal, czy instalujesz komputer pokładowy RCSIM, czy stację MCS.
2. W **Connection Settings** wpisz host, użytkownika i właściwy sposób uwierzytelniania. Zakończ edycję pola, aby wysłać zmianę.

3. Użyj **Test Connection** i odczytaj wynik w konsoli. Sam zielony ping nie potwierdza logowania SSH ani działania programu w pojeździe.
4. Wskaż właściwy **Project Source Path** i **Application Platform Type**. Zakończ edycję i sprawdź zapis po odświeżeniu.
5. Dla standardowego RCSIM ustaw sieć, protokół, czujniki, kamerę i ewentualny RTK zgodnie z faktycznym wyposażeniem. Dla MCS przygotuj własną konfigurację oraz gotowy frontend w katalogu źródłowym MCS.
6. Jeśli użyjesz **Scan Hardware**, przeczytaj wynik i odśwież formularz przed dalszą edycją; skan może zmienić ustawienia na serwerze.
7. Wybierz właściwy rodzaj operacji. Przy pierwszej pełnej instalacji sprawdź i odznacz **Fast Deploy (Skip OS Config)**; domyślne zaznaczenie nie oznacza przygotowanego systemu Pi.
8. Uruchom jedną akcję i obserwuj **Deployment Live Console**. Czytaj końcowy wynik, nie tylko procent postępu lub ponownie aktywny przycisk.
9. Po zakończeniu sprawdź właściwą usługę, połączenie programu docelowego, świeżą telemetrię i obraz, jeżeli ich używasz. W MCS sprawdź profile i mapowanie. Dopiero osobna próba całego toru potwierdza gotowość modelu.

Przy błędzie zachowaj końcowe komunikaty i wróć do procedur odzyskiwania w rozdziale 6. Narzędzie nie zapewnia automatycznego cofnięcia każdej zmiany. Zamknięcie karty przeglądarki nie jest anulowaniem działającego polecenia na Pi.

3. Połączenie — przygotowanie dostępu do Raspberry Pi



The screenshot shows a dark-themed web form titled "Connection Settings" with a key icon. It contains several input fields and buttons. The "RPi Host/IP" field has a placeholder "e.g. 10.0.0.5 or tailscale-ip". The "SSH Username" field has a placeholder "e.g. pi". Below these is a checkbox labeled "Use SSH Private Key". The "SSH Password" field is masked with dots. Below it is a field for "New SSH Password (Optional)" with the placeholder "Leave empty to keep current". At the bottom are two large buttons: "Test Connection" and "Scan Hardware (I2C/UART)".

Deployment Tool — dane połączenia i uwierzytelnianie SSH

Ilustracje tego rozdziału pokazują podgląd interfejsu na ustawieniach przykładowych, bez wykonywania SSH i bez testu sprzętu. Puste pola oraz stan offline nie są wynikiem próby konkretnego Raspberry Pi.

Panel **Connection Settings / Ustawienia połączenia** opisuje dostęp administracyjny do Raspberry Pi. Nie konfiguruje sterowania kierownicą ani radiowego failsafe. Przed wdrożeniem zatrzymaj pojazd i zabezpiecz napęd: późniejsze operacje mogą zatrzymywać usługi lub restartować komputer pokładowy.

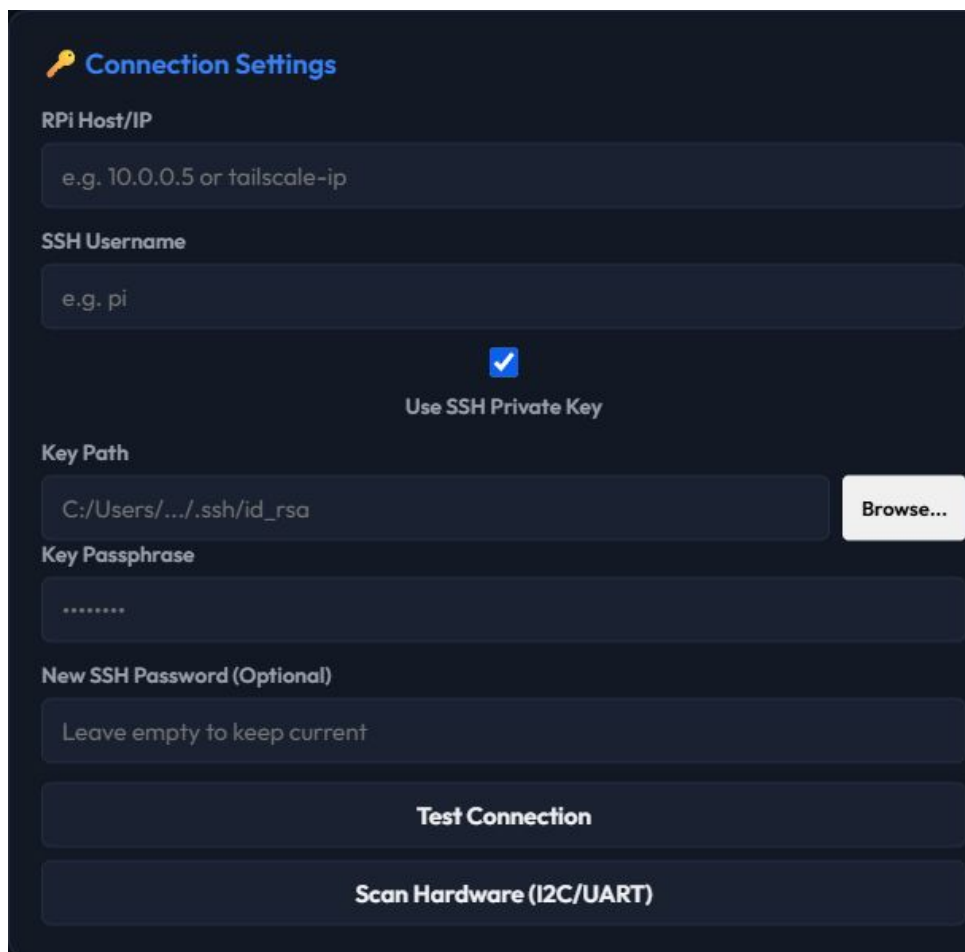
3.1. Adres, użytkownik i hasło

- **RPI Host/IP:** wpisz adres Raspberry Pi dostępny z komputera uruchamiającego Deployment Tool, np. adres LAN albo przygotowanej sieci Tailscale. Wpisuj host/IP bez `http://`, ścieżki strony i portu panelu MCS. Web UI nie ma osobnego pola niestandardowego portu SSH; nie wpisuj `:8080` do adresu SSH.
- **SSH Username:** istniejące konto użytkownika na Pi. `pi` w odpowiedzi jest przykładem, nie gwarancją istnienia takiego konta. Konto musi mieć uprawnienia wymagane przez wybrane wdrożenie.
- **SSH Password:** aktualne hasło tego konta, używane po wyłączeniu **Use SSH Private Key**. To nie hasło Wi-Fi, NTRIP ani kod parowania radia.
- **Test Connection / Testuj połączenie:** sprawdza dostęp SSH przy zapisanych danych. Poczekał na wynik w konsoli; samo odblokowanie przycisku nie oznacza sukcesu.

Pierwsza próba: wpisz host, użytkownika i aktualne hasło, zakończ edycję kliknięciem poza polem, zaczekaj na zapis, następnie wybierz Test Connection. Nie uruchamiaj wdrożenia, dopóki konsola nie potwierdzi połączenia.

Gdy test się nie udaje: sprawdź, czy Pi działa, czy komputer narzędzia ma trasę do podanego adresu i czy SSH jest włączone. Rozróżnij błąd sieci od odrzucenia użytkownika/hasła. Status **RPI: Online** na górze nie zastępuje próby uwierzytelnienia i nie potwierdza działania aplikacji.

3.2. Prywatny klucz SSH



Connection Settings

RPi Host/IP
e.g. 10.0.0.5 or tailscale-ip

SSH Username
e.g. pi

☒ **Use SSH Private Key**

Key Path
C:/Users/.../.ssh/id_rsa **Browse...**

Key Passphrase
.....

New SSH Password (Optional)
Leave empty to keep current

Test Connection

Scan Hardware (I2C/UART)

Logowanie kluczem — ścieżka prywatnego klucza i passphrase

1. Zaznacz **Use SSH Private Key / Użyj prywatnego klucza SSH**. Pole hasła konta zostanie zastąpione polami klucza.
2. W **Key Path** wpisz pełną ścieżkę prywatnego klucza na komputerze, na którym działa Deployment Tool. Wybieraj klucz prywatny, nie plik publiczny zakończony `.pub`.
3. **Browse / Przeglądaj** otwiera systemowe okno wyboru pliku na tym komputerze. Jeśli oglądasz panel przez inną przeglądarkę lub urządzenie, nie jest to przesyłanie pliku z telefonu: okno może otworzyć się na komputerze serwera narzędzia.
4. W **Key Passphrase** wpisz hasło szyfrujące klucz, jeśli je ma. Nie wpisuj tutaj automatycznie hasła konta SSH.
5. Zakończ edycję, następnie wykonaj **Test Connection**. Poprawne pole ścieżki nie dowodzi, że klucz jest uprawniony na Pi.

Przy błędzie sprawdź istnienie pliku na właściwym komputerze, uprawnienie klucza dla wskazanego konta i passphrase. Zmiana metody logowania nie dodaje samoczynnie klucza do Raspberry Pi.

3.3. New SSH Password (Optional)

To pole opisuje zamiar zmiany hasła podczas przygotowania systemu, a nie dane obecnego logowania. Puste pole oznacza zachowanie obecnego hasła w obsługiwanej ścieżce.

Ograniczenie Web UI: w sprawdzonej wersji wartość **New SSH Password (Optional)** nie jest zapisywana przez formularz konfiguracji. Nie zakładaj, że wpisanie tutaj nowego hasła je zmieniło. Po wdrożeniu nadal

sprawdź logowanie wcześniej działającymi danymi; jeśli chcesz celowo zmienić hasło, użyj obsługiwanej procedury administracyjnej i zweryfikuj nowe logowanie osobno.

Wersja okienkowa ma własną ścieżkę przekazywania tego ustawienia; nie przenoś jej zachowania automatycznie na panel przeglądarkowy. Wypełnienie pola nie jest potwierdzeniem wykonania zmiany na Pi.

3.4. Scan Hardware (I2C/UART)

1. Najpierw uzyskaj pozytywny wynik **Test Connection**.
2. Podłącz i zasil sensory zgodnie z ich instrukcją. Nie podłączaj lub przepinaj UART/I2C przypadkowo podczas skanowania.
3. Wybierz **Scan Hardware (I2C/UART) / Skanuj sprzęt** i przeczytaj wynik konsoli: adresy I2C, rozpoznane porty GPS/LiDAR i kamerę.
4. Po zakończeniu odśwież stronę, aby pobrać zaktualizowane ustawienia przed dalszą edycją. Skan może zapisać wykryte parametry w narzędziu, choć formularz nadal pokazuje wcześniejsze wartości.
5. Porównaj wynik z rzeczywistym modelem sensora. Wspólny adres I2C 0x68/0x69 nie rozstrzyga jednoznacznie, czy to MPU6050, MPU9250 lub inny zgodny moduł; wykryty sterownik jest propozycją do sprawdzenia.

Skan nie wgrywa firmware i nie dowodzi poprawnych jednostek ani orientacji sensora. Brak wykrycia nie uprawnia do losowej zmiany GPIO lub napięcia zasilania. Nie zostawiaj starego formularza po skanie bez kontroli: późniejsza zmiana innego pola może ponownie zapisać nieaktualne wartości.

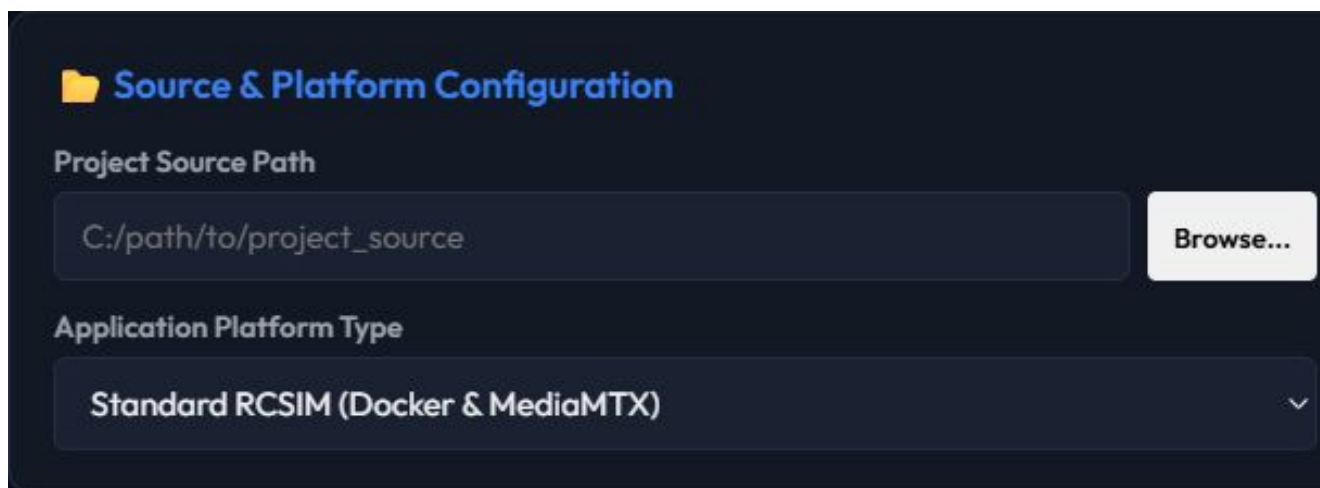
3.5. Zapis ustawień a wykonanie operacji

Panel przeglądarkowy zapisuje formularz automatycznie po zmianie pola/listy; w tekście trzeba zakończyć edycję, np. klawiszem Tab. Nie ma ogólnego przycisku Apply, który od razu aktualizuje Pi. Zapis oznacza zapamiętanie danych w Deployment Tool, a zastosowanie na Pi wymaga właściwego wdrożenia lub dedykowanej aktualizacji.

Przed uruchomieniem operacji zaczekaj na zakończenie edycji i zapisu. Po istotnych zmianach odśwież stronę i porównaj wartości. Wyświetlenie tekstu w polu nie potwierdza poprawności; część nieprawidłowych liczb może zostać zastąpiona domyślnymi wartościami, a inne zatrzymać operację.

Narzędzie zapamiętuje dane logowania w lokalnych ustawieniach. Traktuj tę konfigurację jako poufną; nie dołączaj jej bez sprawdzenia do zgłoszeń i nie zakładaj, że ukrycie hasła kropkami oznacza jego zaszyfrowanie w pliku.

4. Ustawienia wdrożenia — platforma, sieć i wyposażenie



Deployment Tool — katalog źródłowy, platforma i komunikacja

4.1. Project Source Path — co faktycznie wysyłasz

Project Source Path to folder na komputerze uruchamiającym narzędzie, którego zawartość zostanie spakowana do wdrożenia. To nie ścieżka docelowa na Pi, archiwum ZIP ani lokalizacja pliku PDF z instrukcją.

1. Wybierz **Application Platform Type** przed ustaleniem katalogu.
2. Dla **Standard RCSIM (Docke & MediaMTX)** wskaż przygotowany katalog projektu pokładowego z plikami wymaganymi przez ten wariant, a nie dowolny katalog całego repozytorium.
3. Dla **RCSIM MCS (Direct Services)** wskaż katalog modułu MCS zawierający jego aplikację, wymagania, konfigurację i przygotowany interfejs. Nie wskazuj wyłącznie katalogu frontendu ani samego pliku wykonywalnego.
4. Użyj **Browse**, sprawdź ścieżkę, zakończ edycję i potwierdź ją po odświeżeniu. Wybieranie folderu odbywa się na komputerze narzędzia.
5. Sprawdź wersję i kompletność źródeł przed wdrożeniem. Narzędzie sprawdza istnienie katalogu, ale samo jego istnienie nie dowodzi, że zawiera właściwy projekt.

Gdy pakowanie się nie udaje: sprawdź pełną ścieżkę, dostęp do katalogu i zgodność wybranej platformy. Pusta ścieżka lub folder z innym modułem nie zostaną naprawione przez Test Connection; test SSH dotyczy innego etapu.

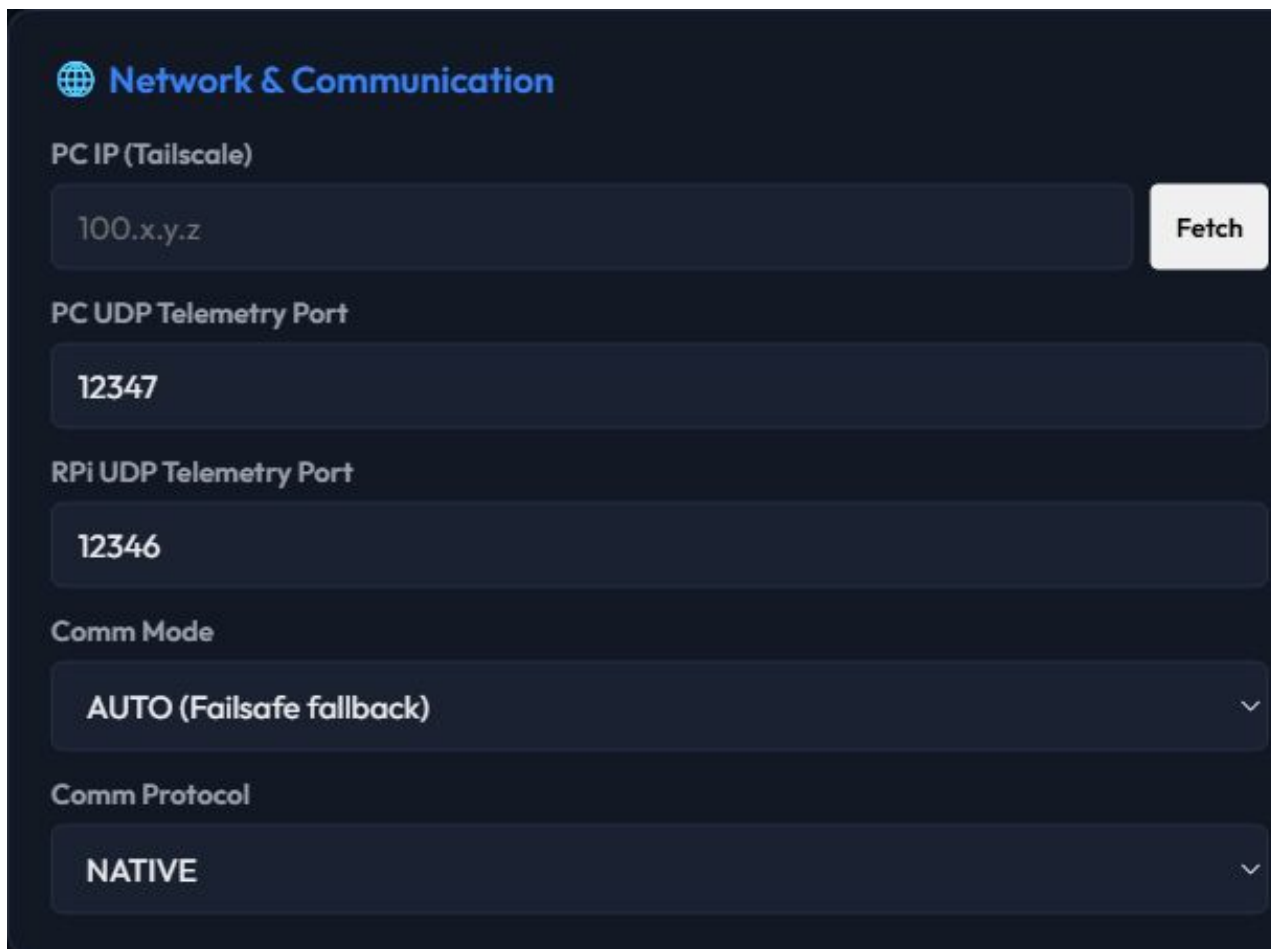
4.2. Application Platform Type — dwa różne wdrożenia

- **Standard RCSIM (Docke & MediaMTX):** wariant programu pokładowego i wideo. Formularz buduje ustawienia konfiguracji sprzętu, sieci, kamery i RTK przeznaczone do paczki tego wariantu.
- **RCSIM MCS (Direct Services):** wariant mobilnej stacji sterowania działającej jako usługi. Po wyborze znikają karty kamery i RTK; działania ściśle dockerowe nie są właściwym odpowiednikiem obsługi MCS.

Ważne dla MCS: widoczne ogólne pola IMU/GPS/LiDAR/ELRS i sieci nie są automatycznie przekształcane w pełną konfigurację wykonawczą MCS. Ta ścieżka pakuje wskazany katalog MCS; profile kontrolerów, wyjście i sprzęt skonfiguruj we właściwym panelu MCS. Nie zakładaj, że wybór MPU w Deployment Tool nadpisze profil MCS.

Narzędzie nie oferuje w tym Web UI osobnego menedżera nazwanych profili wdrożenia. Ustawienia ostatniego formularza nie są kilkoma niezależnymi presetami dla różnych pojazdów. Po zmianie platformy ponownie sprawdź katalog, adres, protokoły i sensory.

4.3. Sieć i protokół standardowego RCSIM



The screenshot shows the 'Network & Communication' settings panel. It contains the following fields and controls:

- PC IP (Tailscale):** A text input field containing '100.x.y.z' and a 'Fetch' button.
- PC UDP Telemetry Port:** A text input field containing '12347'.
- RPi UDP Telemetry Port:** A text input field containing '12346'.
- Comm Mode:** A dropdown menu with 'AUTO (Failsafe fallback)' selected.
- Comm Protocol:** A dropdown menu with 'NATIVE' selected.

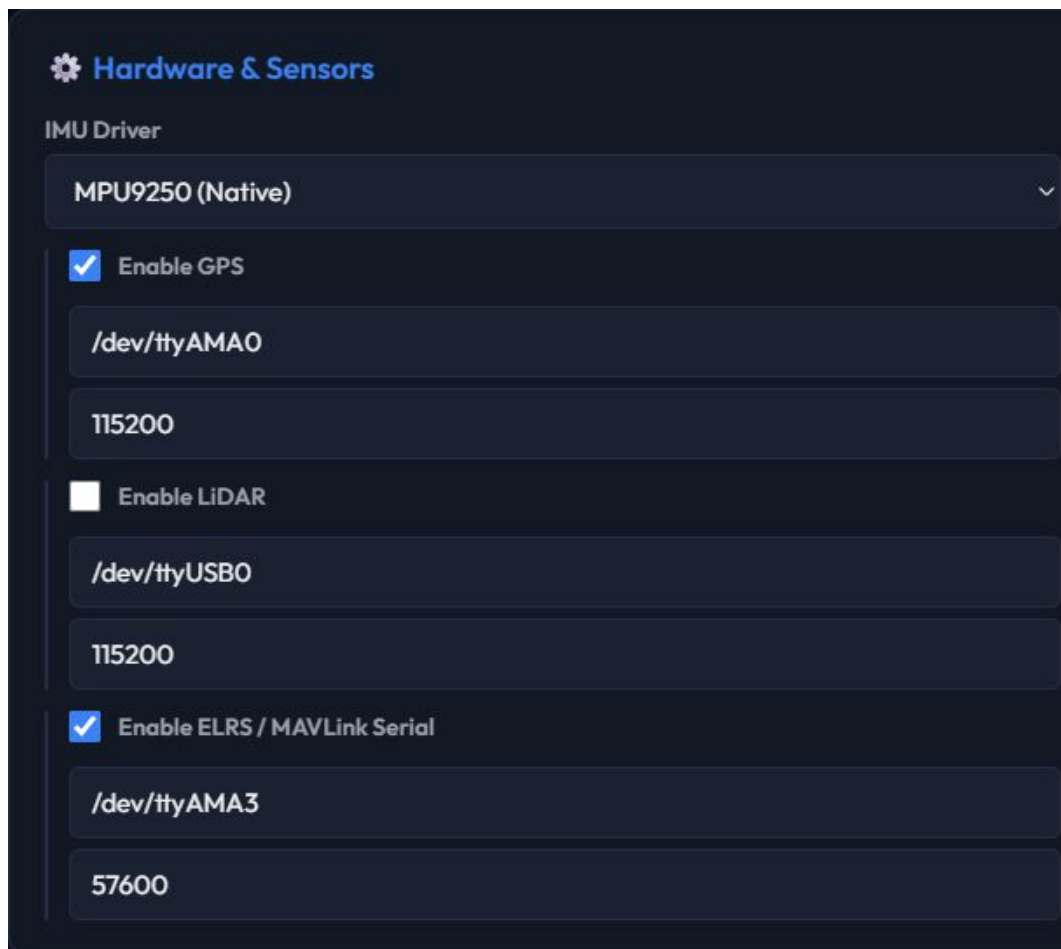
Ustawienia sieci — adres stacji PC, porty i protokoły

- **PC IP (Tailscale):** adres stacji PC, do której pojazd ma kierować odpowiednie dane. Nie jest adresem Pi wpisywanym w panelu SSH.
- **Fetch / Pobierz:** próbuje odczytać adres IPv4 Tailscale komputera narzędzia, a gdy to się nie uda może zwrócić zwykły adres tego komputera. Sprawdź wynik; nie dowodzi instalacji ani działania VPN na Pi.
- **PC UDP Telemetry Port:** port odbioru po stronie PC, np. 12347 tylko wtedy, gdy GCS nasłuchuje właśnie tam.
- **RPi UDP Telemetry Port:** port po stronie Raspberry Pi, np. 12346 dla zgodnej konfiguracji. Porty PC i Pi pełnią różne role; nie wyrównuj ich bez uzasadnienia.
- **Comm Mode:** AUTO, WEBRTC lub UDP. AUTO jest wyborem logiki używanej przez zgodne oprogramowanie, a nie obietnicą automatycznej naprawy każdego łącza ani zatrzymania napędu.
- **Comm Protocol:** NATIVE albo MAVLINK. Dobierz go do oprogramowania pokładowego oraz GCS. MAVLINK nie oznacza samoczynnego wgrania autopilota lub zmiany firmware radia.

Próba konfiguracji sieci: wpisz właściwy adres odbiorcy, ustaw porty po obu stronach, dobierz tryb/protokół, zakończ edycję i dopiero potem wdrażaj. Po zakończeniu sprawdź świeżą telemetrię w GCS oraz osobno sterowanie i obraz. Brak obrazu nie jest podstawą do losowej zmiany UDP Telemetry Port.

W panelu web nie ma osobnych pól **MAVLink System ID** i **MAVLink Throttle Hz** znanych z wersji okienkowej. Nie szukaj ich na ilustracji. Wersja okienkowa ma identyfikator systemu i listę częstości 1, 2, 5, 10, 20, 50 Hz; nie należy mylić tej częstości z prędkością portu szeregowego.

4.4. IMU, GPS, LiDAR i ELRS / MAVLink Serial



Deployment Tool — wybór sterowników i portów wyposażenia

IMU Driver: wybierz rzeczywiście zamontowany moduł: MPU9250 (Native), MPU6050 (Native), GY-91, BMX160 + BMP388, BMX160 lub BNO08x. **Simulation (Dummy IMU)** jest źródłem demonstracyjnym, a nie dowodem działania fizycznego czujnika. Po skanie I2C potwierdź model ręcznie.

Enable GPS: zaznacz, jeśli pojazd ma obsługiwany odbiornik. **GPS Serial Port** to ścieżka urządzenia na Pi, np. `/dev/ttyAMA0`, nie port COM komputera Windows. **Baudrate** wpisz jako dodatnią liczbę całkowitą zgodną z aktywnym wyjściem GPS, np. 115200 tylko dla tak ustawionego urządzenia.

Enable LiDAR: włącza konfigurację sensora. **LiDAR Port** to jego ścieżka na Pi, a **Baudrate** prędkość konkretnego urządzenia, nie częstość skanowania w Hz. `/dev/ttyUSB0` jest przykładem, który po zmianie urządzeń USB może oznaczać inny adapter.

Enable ELRS / MAVLink Serial: włącza odpowiednią konfigurację szeregową. **ELRS Serial Port** to port Pi, **Baudrate** dotyczy odcinka Pi–moduł. 57600 widoczne jako wartość domyślna nie jest uniwersalną prędkością CRSF, ELRS lub mLRs; musi odpowiadać używanemu firmware i protokołowi.

Procedura dla każdego sensora:

1. Sprawdź fizyczny model, zasilanie i interfejs według instrukcji platformy.
2. Użyj skanu jako pomocy, odśwież formularz i sprawdź zaproponowany port/sterownik.

3. Wyłącz funkcje urządzeń, których nie ma w pojeździe. Nie pozostawiaj domyślnie włączonego GPS/ELRS tylko dlatego, że tak otworzył się formularz.
4. Wpisz właściwy port na Pi i dodatni Baudrate bez dopisku bps i bez separatorów tysięcy. Nie przypisuj tej samej fizycznej linii GPS, LiDAR i radiu.
5. Zastosuj konfigurację przez właściwe wdrożenie. Po uruchomieniu sprawdź czujnik w danych aplikacji, jego aktualność, jednostki i kierunki; nie tylko komunikat o wykryciu portu.

Przy braku danych sprawdź uprawnienia do portu, jego zajęcie przez inną usługę i prędkość urządzenia. Wybranie sterownika nie instaluje sensora fizycznie i nie kalibruje jego orientacji.

4.5. Kamera — port, sensor i profil obrazu

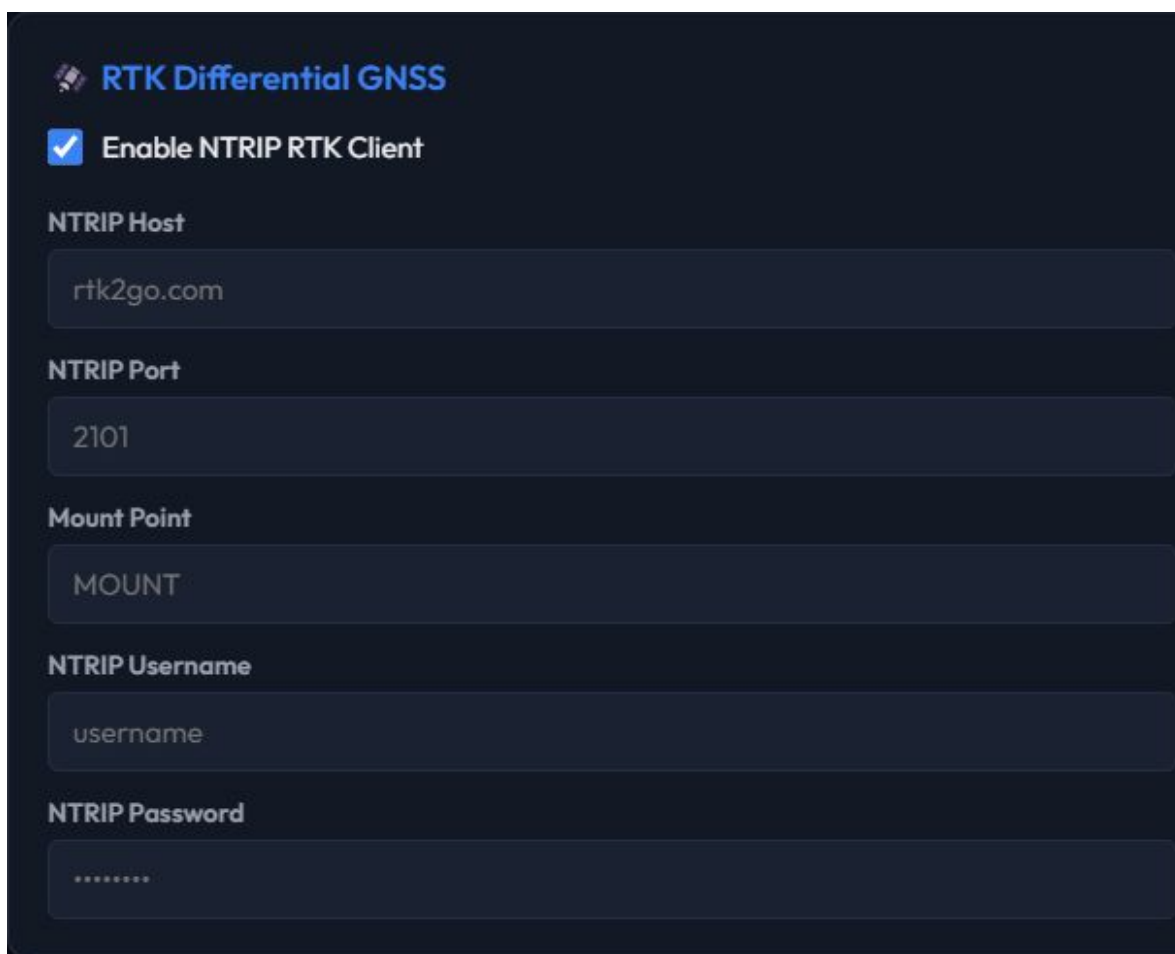
Karta wideo jest widoczna dla standardowego RCSIM, a ukrywana dla MCS. **Camera CSI Port** wybiera cam0, cam1, /dev/video0 albo /dev/video1; dobierz rzeczywisty tor urządzenia.

- **Camera Sensor Type:** AUTO (Detect), imx219, imx708, ov5647, imx477 lub imx519. Dobierz sensor do kamery; wybór innego modelu nie zapewnia zgodności.
- **Resolution:** wpisuj szerokość i wysokość w pikselach, np. 1920x1080, z małym x, bez jednostki i dopisku Full HD.
- **FPS:** dodatnia całkowita liczba klatek na sekundę, np. 30. Jest parametrem zadany, nie gwarancją uzyskanego obrazu.
- **Video Bitrate:** użyj zapisu zgodnego z formularzem, np. 5 Mbps. W tej wersji oczekiwana jest całkowita liczba Mbps na początku; 2.5 Mbps, 5000000 lub zapis z przecinkiem nie są równoważnymi bezpiecznymi formatami.
- **Detect CSI Cameras:** odczytuje wykrywanie kamer z Pi i wypisuje wynik. Przeczytaj konsolę; samo kliknięcie nie oznacza, że wybrano właściwy sensor/port.
- **Low Latency Profile:** ustawia lokalnie **800x600, 30 FPS, 2 Mbps** i zapisuje formularz. To preset, nie pomiar opóźnienia i nie natychmiastowe przełączenie strumienia na Pi.

Procedura: sprawdź kamerę przez Detect CSI Cameras, wybierz zgodny port i sensor, ustaw rozdzielczość/FPS/bitrate, zakończ edycję i wdroż konfigurację albo użyj właściwej dedykowanej aktualizacji kamery. Następnie otwórz rzeczywisty strumień w GCS i sprawdź aktualny obraz, płynność i opóźnienie ruchu.

Jeśli chcesz zacząć od mniejszego obciążenia transmisji, użyj Low Latency Profile i świadomie zaakceptuj niższą rozdzielczość. Przy czarnym obrazie najpierw sprawdź wykrycie sensora, złącze, usługę wideo i jej log, zamiast wyłącznie zwiększać bitrate.

4.6. NTRIP RTK — dane korekcyjne dla zgodnego GPS



RTK Differential GNSS

☒ **Enable NTRIP RTK Client**

NTRIP Host

rtk2go.com

NTRIP Port

2101

Mount Point

MOUNT

NTRIP Username

username

NTRIP Password

NTRIP RTK — serwer korekt, mountpoint i dane logowania

Enable NTRIP RTK Client jest dostępne w wariantcie standardowym. Do poprawy pozycji potrzebny jest odbiornik obsługujący RTK oraz właściwy tor przekazania korekt; sam przełącznik nie zamienia zwykłego GPS w odbiornik RTK.

- **NTRIP Host:** nazwa lub adres serwera korekt bez adresu strony WWW i ścieżki.
- **NTRIP Port:** numer portu usługi, np. 2101 tylko dla tak skonfigurowanego dostawcy.
- **Mount Point:** dokładna nazwa strumienia korekt udostępnionego przez dostawcę, z zachowaniem pisowni.
- **NTRIP Username / Password:** dane tego serwisu; są niezależne od logowania SSH i sieci Wi-Fi.

Procedura: uzyskaj dane strumienia, potwierdź zgodny odbiornik i internet na Pi, wypełnij pola, zakończ edycję i zastosuj konfigurację we wdrożeniu. Po uruchomieniu sprawdź połączenie z casterem, napływ korekt i rzeczywisty stan rozwiązania GPS. Same współrzędne, liczba satelitów albo napis „RTK enabled” nie potwierdzają RTK FIX ani określonej dokładności.

Jeśli RTK nie jest używane, wyłącz klienta. Przy błędzie uwierzytelniania sprawdź konto i mountpoint; przy braku połączenia host, port i dostęp Pi do internetu. Nie zmieniaj na tej podstawie prędkości sterowania radiowego.

4.7. Ostateczna kontrola formularza

Przed wdrożeniem potwierdź: właściwy host i konto, udany test SSH, poprawną platformę i istniejący katalog źródłowy. Dla RCSIM sprawdź adres PC, porty, protokoły, faktyczne sensory, formaty liczb, kamerę i opcjonalne NTRIP. Dla MCS sprawdź kompletność źródeł i przygotuj późniejszą konfigurację w panelu MCS. Po zakończeniu edycji zaczekaj na zapis; dopiero wtedy wybierz działanie wdrożeniowe opisane w kolejnej części instrukcji.

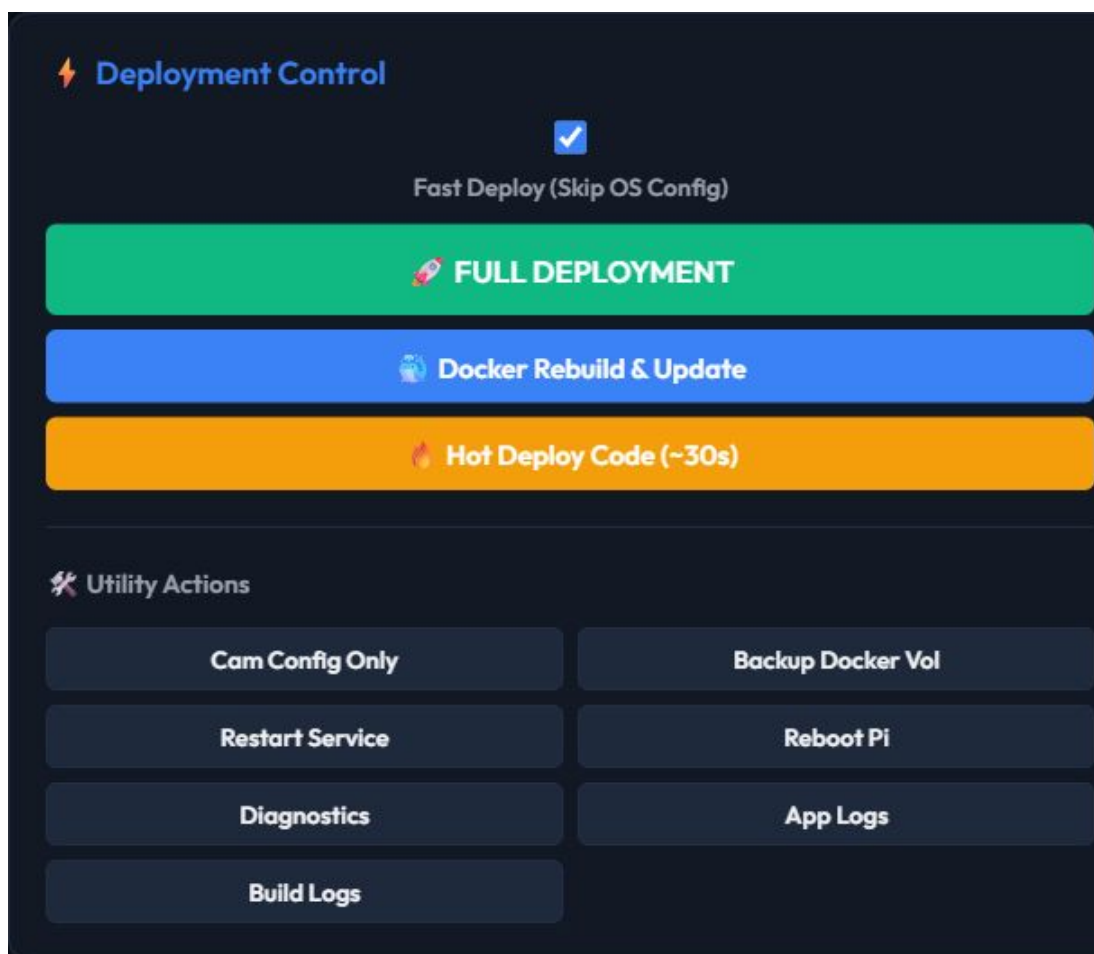
5. Przygotowanie i wdrożenie

Co uruchamiasz i gdzie powstaje instalacja

Aktualny Deployment Tool uruchamia lokalny serwer i otwiera panel w przeglądarce. Aplikacja wybiera wolny port z zakresu **5000–5009** na 127.0.0.1. Używaj adresu podanego przy starcie; nie jest to WebUI MCS na porcie 8080 ani pokładowa usługa WebRTC.

Źródła i archiwum przygotowuje komputer, na którym działa narzędzie. Przesyłanie następuje przez SSH/SFTP do Raspberry Pi, a instalacja i budowanie obrazu Docker wykonują się na Pi. W aktualnym panelu nie ma osobnego wyboru „build lokalny / zdalny”. Przeglądarka nie kompiluje obrazu, a wybór katalogu wskazuje folder na komputerze uruchamiającym narzędzie.

Akcje są w prawym panelu **Deployment Control**, nie w osobnej zakładce Deployment.



Panel sterowania wdrożeniem

Ilustracja pokazuje rzeczywisty frontend bez aktywnego połączenia SSH. Na potrzeby dokumentacji nie wykonywano wdrożenia.

Lista przed rozpoczęciem

1. Zatrzymaj model, rozbrój i fizycznie zabezpiecz napęd. Aktualizacja przerywa usługi sterowania lub obraz; nie wykonuj jej podczas jazdy.
2. Sprawdź docelowy host, konto i sposób logowania SSH. Musi istnieć dostęp sudo do instalowania pakietów, usług, konfiguracji systemu i Dockera.
3. Zapewnij stabilne zasilanie, miejsce na karcie i pamięć RAM. Docker jest budowany w katalogu RAM /dev/shm; wolne miejsce na karcie nie rozwiązuje braku RAM.
4. Zapewnij dostęp Pi do repozytoriów pakietów oraz potrzebnych zależności. Działające SSH nie oznacza dostępu do Internetu z Pi.
5. Zachowaj własną kopię aktywnej konfiguracji i potrzebnych danych z urządzenia przed ich zastąpieniem. Nie zakładaj automatycznej kopii ani automatycznego powrotu do starej wersji.
6. Wybierz właściwy **Application Type** i katalog **Project Source**; po wybraniu folderu lub pobraniu IP kliknij poza pole i sprawdź zapis. Te przyciski uzupełniają pole, ale samo uzupełnienie nie gwarantuje wysłania nowej wartości do ustawień serwera.
7. Zakończ edycję każdego pola przed uruchomieniem akcji. Zmiany są zapisywane przy zdarzeniu zmiany pola; nie rozpoczynaj wdrożenia w trakcie wpisywania adresu lub ścieżki.
8. Wykonaj **Test Connection**, a dopiero po potwierdzeniu SSH uruchamiaj kolejne czynności.

Dwa typy aplikacji — wybór zmienia cel instalacji

RCSIM Docker służy komputerowi pokładowemu Tier 4. Wskaż

RCSIM_RPI/RCSIM_rpi_embedded/rpi_project_source, czyli katalog zawierający Dockerfile, docker-compose.yml, entrypoint.sh, config.json oraz kod aplikacji. Nie wybieraj całego repozytorium ani katalogu MCS.

RCSIM MCS (Direct Services) instaluje stację MCS jako usługę systemową. Wskaż katalog RCSIM_MCS zawierający requirements.txt, systemd/usb_rc.service, src/, config/ i gotowe pliki src/ui/web/. To inny program i inny cel niż obraz pokładowy Tier 4.

Dla MCS panel ukrywa ustawienia wideo i RTK oraz próbuje zablokować przyciski kopii i logów budowania. Okresowe odświeżanie statusu może jednak ponownie uaktywnić te przyciski po około 2,5 s. **Backup Docker Vol** i **Hot Deploy Code** nadal dotyczą ścieżki Docker. **Build Logs** ma w backendzie osobny odczyt logu instalacji MCS; szczegóły opisuje rozdział 7. Aktywny przycisk sam nie potwierdza zgodności pozostałych funkcji z platformą.

Co trafia do paczki

Narzędzie tworzy lokalny katalog roboczy temp_deploy i archiwum ZIP. Pomija m.in. repozytorium Git, środowiska Python, node_modules, cache, logi oraz obrazy/archiwa. Nie przesyła automatycznie całego środowiska uruchomieniowego komputera.

W wariancie Docker kopia źródłowego config.json jest łączona z ustawieniami panelu. Zachowywane są istniejące klucze, a nowe wartości nadpisują odpowiednie pola; nie jest to bezwarunkowe zachowanie wszystkich ręcznych ustawień z Pi. Powstaje też mediamtx.yml na podstawie parametrów kamery. Kamera aplikacji jest ustawiana na RTSP rtsp://127.0.0.1:8554/camera_ai.

W wariancie MCS paczka kopiuje konfigurację z wybranego katalogu źródłowego. Pokładowe pola sensorów/RTK z Deployment Tool nie są konwerterem profili MCS. Uważaj na lokalny config/config.json: może zastąpić profile zapisane wcześniej na Pi. Zbuduj frontend MCS przed pakowaniem i sprawdź obecność aktualnego src/ui/web/; samo wysłanie źródeł React nie wykonuje kompilacji na urządzeniu.

Pełne wdrożenie Docker — pierwsza instalacja

1. Wybierz Docker, właściwy katalog i host. Ustaw komunikację, wyposażenie oraz kamerę zgodnie z rzeczywistym zestawem.
2. Na pierwszą instalację odznacz **Fast Deploy (Skip OS Config)**. Przycisk pełnego wdrożenia korzysta z aktualnej wartości tego checkboxa; sam napis FULL nie wymusza pełnej konfiguracji systemu.
3. Wybierz **FULL DEPLOYMENT** i obserwuj dolną konsolę. Najpierw powstaje archiwum, potem następuje połączenie, przesłanie i wykonanie skryptu na Pi.
4. Jeżeli pojawi się żądanie logowania Tailscale, dokończ je dla właściwego urządzenia. Otworzenie strony logowania nie jest końcem instalacji.
5. Poczekaj na wynik skryptu i komunikat końcowy; nie uruchamiaj kolejnej akcji po samym przesłaniu ZIP.
6. Sprawdź po wdrożeniu aplikację, MediaMTX, telemetrię i obraz, a potem osobno bezpieczny sygnał sterowania.

Rzeczywisty zakres zmian: skrypt instaluje/weryfikuje pakiety i zależności, konfiguruje usługi, instaluje lub aktualizuje hostowy MediaMTX, buduje obraz na Pi, zastępuje katalog `/home/USER/rcsim_project` i odtwarza kontenery z Compose. Może wyłączyć starszą usługę `rtsp-camera.service`. Instalator posiada także mechanizm zmiany hasła SSH, ale w aktualnej ścieżce WWW pole **New SSH Password** nie jest skutecznie przekazywane do operacji; nie oczekuj zmiany hasła po samym jego wypełnieniu (patrz rozdział 3.3).

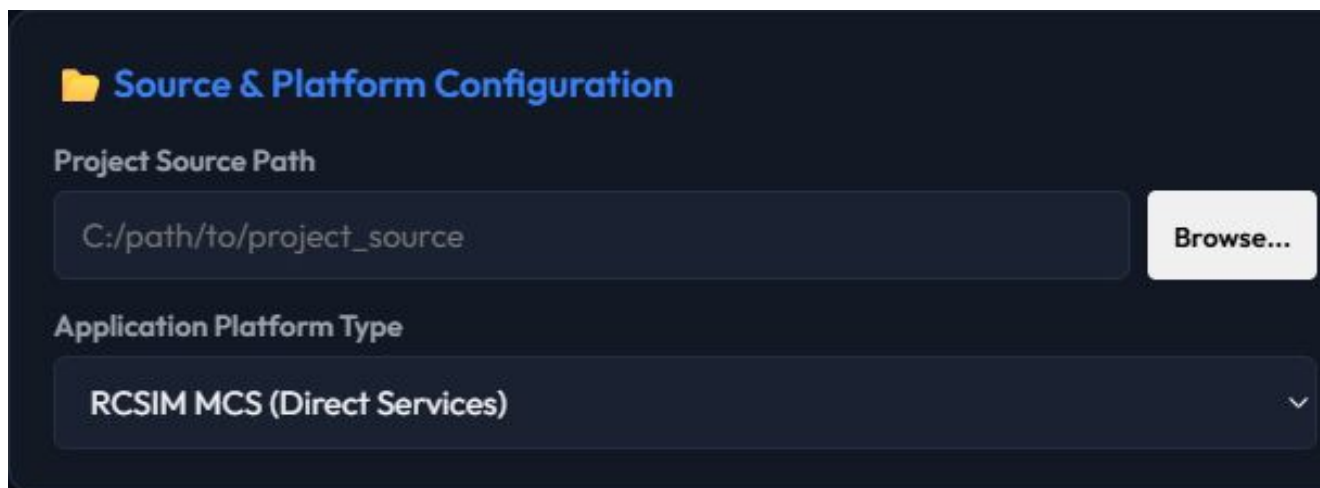
Przed budowaniem skrypt czyści `/tmp/*` **całego hosta**, wykonuje `docker system prune -f` i restartuje usługę Docker. To nie jest czyszczenie wyłącznie plików RCSIM: może usunąć tymczasowe dane innych programów i nieużywane zasoby Docker oraz przerwać pracę innych kontenerów. Używaj tej ścieżki na przygotowanym hoście i zachowaj potrzebne dane wcześniej.

Fast Deploy oraz Docker Rebuild & Update

Fast Deploy (Skip OS Config) ogranicza część kroków systemowych, w szczególności początkowe czyszczenie/aktualizację APT. Nie jest obietnicą braku zmian w systemie. Pozostają m.in. kontrole zależności, obsługa MediaMTX, budowanie Docker, czyszczenie `/tmp`, `prune Docker`, restart Dockera i zastąpienie projektu.

Docker Rebuild & Update uruchamia tę samą procedurę wdrożenia z wymuszonym Fast Mode. Wybierz ją po zmianie `Dockerfile`, zależności systemowych/Python lub wtedy, gdy aktualizacja samego kodu nie wystarczy. Przygotuj kopię danych i sprawdź wynik tak samo jak po pełnym wdrożeniu.

W aktualnym panelu nie ma osobnego przycisku **Clean Build**, wyboru kompilacji na PC ani instalowania gotowego obrazu z lokalnego cache. Nie zastępuj ich znaczeniem podobnie brzmiących opcji; „Rebuild” dotyczy zdalnego budowania Compose.



Source & Platform Configuration

Project Source Path

C:/path/to/project_source Browse...

Application Platform Type

RCSIM MCS (Direct Services) ▼

Panel akcji dla wariantu MCS — podgląd bez połączenia SSH

1. Wybierz **RCSIM MCS (Direct Services)**, katalog MCS i właściwy host.
2. Zachowaj profile z Pi oraz sprawdź konfigurację w paczce. Potwierdź gotowy frontend.
3. Użyj **FULL DEPLOYMENT**. Fast Mode nie jest przekazywany do osobnego instalatora MCS i nie wyłącza jego kroków systemowych.
4. Zaczekaj na instalację pakietów, rozpakowanie do `/opt/usb_rc_converter`, przygotowanie venv i restart `usb_rc.service`.
5. Sprawdź log oraz usługę. Po wymaganym restarcie Pi otwórz WebUI MCS i potwierdź profil, urządzenia i protokół, zanim uzbroisz model.

Instalator MCS nadpisuje pasujące pliki w `/opt/usb_rc_converter`; nie usuwa całego katalogu przed rozpakowaniem, więc stare pliki niewystępujące w nowej paczce mogą pozostać. Wyłącza i usuwa starsze jednostki `usb_rc_core.service` oraz `usb_rc_web.service`, po czym instaluje wspólną `usb_rc.service`.

Modyfikuje też konfigurację startową I2C/UART, wyłącza konsolę szeregową na `ttyAMA0` i przygotowuje interfejs Wi-Fi `uap0` oraz wpisy `rc.local`. To ingerencja w hosta, nie sama podmiana strony. Zmiany UART/I2C wymagają restartu Pi. Utworzenie interfejsu nie potwierdza działającego AP ani kompatybilności każdego adaptera Wi-Fi.

Przycisk **Docker Rebuild & Update** przy typie MCS również trafia do instalatora MCS; nie przebudowuje wtedy obrazu. Dla czytelności używaj jawnie pełnego wdrożenia MCS. Nie używaj **Hot Deploy Code** do aktualizacji MCS.

6. Aktualizacja, kontrola wyniku i odzyskiwanie

Hot Deploy Code — kiedy wystarcza

Ta ścieżka wymaga wcześniejszej instalacji Docker i istniejącego kontenera `rcsim_industrial`. Służy aktualizacji kodu oraz konfiguracji bez budowania nowego obrazu. Nie dostarcza nowych zależności z `Dockerfile` ani `requirements.txt`.

1. Wybierz typ Docker, właściwe źródła i host; zachowaj potrzebne dane z aktywnej instalacji.
2. Zatrzymaj model i uruchom **Hot Deploy Code (~30s)**. Czas w nazwie jest orientacyjny, nie gwarantowany.
3. Poczekaj na zatrzymanie kontenera, synchronizację i ponowne uruchomienie.

4. Sprawdź log pod kątem pojedynczych błędów kopiowania i restartu, a następnie rzeczywiste działanie usług.

Hot deploy może zastąpić `config.json`, `docker-compose.yml` i konfigurację MediaMTX. Wymienia katalogi **core**, **logic**, **modules**, **tools**, **models**, **web_assets** w projekcie hosta; własne pliki w tych katalogach mogą zniknąć. Aktualizuje także wybrane pliki startowe. Gdy Compose się zmienił, usuwa i odtwarza kontener; w przeciwnym razie restartuje istniejący. Następuje przerwa w sterowaniu i obrazie.

Komunikat końcowy Hot Deploy nie jest pełnym testem zdrowia. Skrypt sprawdza obecność uruchomionego kontenera, a nie działanie PWM, sensorów, inferencji czy kanału WebRTC. Niektóre błędy kroków są wypisywane bez bezwarunkowego przerwania całej procedury. Czytaj log, nie tylko ostatni zielony wiersz.

Aktualizacja tylko kamery

Cam Config Only służy ustawieniom hostowego MediaMTX. Po zmianie parametrów kamery użyj tej akcji zamiast przebudowy całego obrazu. Zmienia konfigurację i restartuje usługę, więc obraz może zniknąć na czas operacji. Nie aktualizuje kodu aplikacji, firmware kamery ani profili MCS. Po powrocie usługi sprawdź rzeczywistą rozdzielczość/obraz oraz zgodność źródła w GCS.

Kopia, restart i logi

Backup Docker Vol pobiera archiwum katalogu `rcsim_project` z katalogu domowego użytkownika Pi na komputer. W panelu WWW zapisuje je jako `rcsim_docker_backup.tar.gz` w katalogu domowym użytkownika komputera; kolejna kopia używa tej samej nazwy, więc wcześniej zachowaj potrzebny starszy plik. Nazwa nie oznacza obrazu całej karty SD, kompletnej kopii systemu ani wszystkich wolumenów Docker innych aplikacji. Sprawdź plik po zakończeniu. Kopię konfiguracji MCS wykonaj osobno. Choć odświeżanie panelu może ponownie uaktywnić przycisk w trybie MCS, ta akcja nadal nie kopiuje `/opt/usb_rc_converter` i nie jest kopią MCS.

Restart Service dobiera usługę do typu aplikacji: kontener Docker albo usługę MCS. Nie jest restartem całego Pi. **Reboot Pi** restartuje host, przerywa SSH i wszystkie jego usługi; poczekaj na powrót połączenia przed następną akcją.

App Logs pokazuje log aplikacji odpowiedniego typu. **Build Logs** dotyczy budowania Docker, natomiast **Diagnostics** zbiera informacje diagnostyczne. Log budowania i log uruchomionej aplikacji odpowiadają na różne pytania: poprawna kompilacja nie potwierdza sprawnych sensorów.

Jak rozpoznać zakończenie

Pasek postępu jest przybliżeniem etapów. W pełnym wdrożeniu może dojść do 100% również przed zgłoszeniem błędu skryptu. Odblokowanie przycisków albo pojawienie się archiwum nie oznacza udanej instalacji.

Za wynik instalatora przyjmij komunikat sukcesu po zakończeniu zdalnego skryptu kodem 0. Następnie wykonaj niezależną kontrolę:

- **Docker:** kontener nie wpada w pętlę restartów, MediaMTX jest aktywny, log aplikacji nie zgłasza krytycznych błędów, GCS otrzymuje świeże dane i właściwy obraz.
- **MCS:** `usb_rc.service` pozostaje aktywna, panel odpowiada, profil po odświeżeniu jest właściwy, host widzi kontrolery, a ustawiony port i protokół odpowiadają sprzętowi.
- **Oba warianty:** na zabezpieczonym stanowisku potwierdzono neutralny gaz, kierunek skrętu, uzbrojenie i reakcję na utratę łącza. Żaden zielony wskaźnik nie zastępuje tych prób.

Błąd lub przerwanie

W obecnym panelu nie ma przycisku anulowania aktywnego wdrożenia. Zamknięcie karty przeglądarki nie jest poleceniem wycofania zmian. **Shutdown Server** kończy lokalny program, lecz nie zapewnia przerwania skryptu na Pi ani przywrócenia poprzedniej instalacji. Nie używaj go jako odpowiednika Cancel.

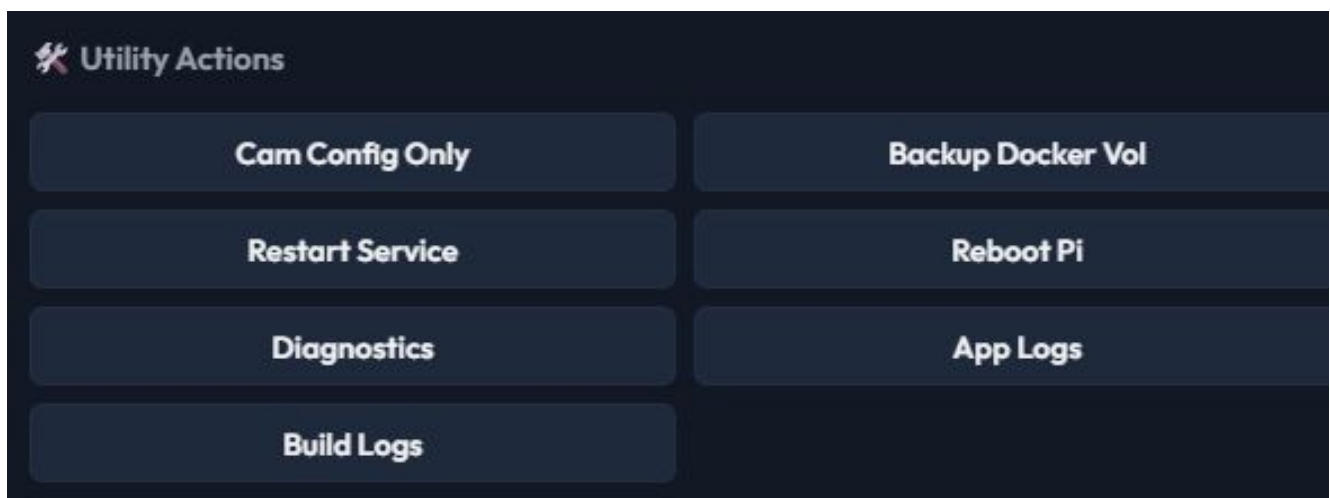
Jeżeli postęp stanął, najpierw odczytaj konsolę i log: pobieranie pakietów lub budowanie może trwać bez częstych zmian procentu. Nie uruchamiaj drugiego wdrożenia. Narzędzie blokuje równoległe akcje, ale utrata przeglądarki lub ponowne uruchomienie programu nie dowodzi zakończenia procesu zdalnego.

Po błędzie sprawdź **App Logs** oraz **Build Logs**. Pełne wdrożenie próbuje także pobrać końcówkę `rpi_setup_docker.log` lub `rpi_setup_mcs.log`, a dla Docker `docker_build.log`, z katalogu użytkownika Pi. Zachowaj pierwszą przyczynę błędu, nie tylko końcowy komunikat.

Jeśli zabrakło pamięci/miejsca, sprawdź oddzielnie RAM i kartę. Jeśli zawiodło SSH, sprawdź host, konto, klucz/hasło i ewentualną zmianę hasła wykonaną przez instalator. Jeśli kontener nie startuje, sprawdź zależności i hostowy MediaMTX. Jeśli brakuje kontenera dla Hot Deploy, wykonaj poprawne pełne wdrożenie Docker.

Odzyskiwanie polega na świadomym ponownym wdrożeniu sprawdzonych źródeł/konfiguracji albo odtworzeniu własnej kopii. Panel nie udostępnia automatycznego rollbacku. Po naprawie powtórz kontrolę całego toru przed jazdą. Opisane procedury zweryfikowano w kodzie; przy tworzeniu podręcznika nie wykonano wdrożenia na rzeczywistym Pi.

7. Usługi, logi i diagnostyka



Podgląd przycisków obsługowych; bez połączenia SSH i bez wykonania operacji

Deployment Tool ma jedną stronę z kartami ustawień i przyciskami obsługowymi. Nie ma osobnej zakładki „Services”. Obsługę wykonuje się w **Utility Actions / Narzędzia**, a wyniki odczytuje w **Deployment Live Console / Konsoli wdrożenia** na dole. Przed każdą operacją sprawdź adres Raspberry Pi i wybraną platformę. Restart lub aktualizacja podczas jazdy przerywa usługi sterowania; najpierw zabezpiecz pojazd i zakończ sesję.

Co oznaczają wskaźniki u góry



Podgląd wskaźników bez połączenia SSH; nie jest to pomiar stanu fizycznego Pi

RPI: Online oznacza udaną odpowiedź na ping, nie działającą aplikację ani zalogowanie SSH. Po udanym pingowaniu narzędzie próbuje sprawdzić usługi przez SSH. Przy zablokowanym pingowaniu pozostałe wskaźniki mogą być nieaktywne mimo dostępnego SSH. Nie używaj samego koloru znacznika jako rozstrzygającego testu połączenia.

Dla platformy Docker wskaźnik **Industrial** sprawdza główną aplikację `rcsim_industrial`, a **Video** usługę MediaMTX. Dla **RCSIM_MCS** znaczniki **Core Service** i **Web Service** pochodzą z tego samego sprawdzenia `usb_rc.service`. Nie są dwoma niezależnymi testami procesów ani dowodem otwarcia strony w przeglądarce. Stan usługi nie potwierdza obrazu kamery, odbioru kanałów ani fizycznego działania czujników.

Test Connection — sprawdź SSH przed naprawą

1. Sprawdź host, użytkownika oraz hasło albo plik klucza SSH.
2. Naciśnij **Test Connection** / **Testuj połączenie**.
3. Odczytaj wynik w konsoli; przy błędzie popraw połączenie przed ponawianiem operacji instalacji.
4. Po zmianie ustawień poczekaj na ich zapis i wykonaj test ponownie.

To właściwy pierwszy krok przy problemach dostępu. Pomyślne SSH nie potwierdza jeszcze uprawnień do każdej komendy `sudo` ani gotowości usługi aplikacji.

Restart Service — restart programu

Restart Service / **Restart usługi** dotyczy platformy wskazanej w formularzu:

- Docker: restartuje kontener `rcsim_industrial`.
- MCS: restartuje usługę `usb_rc.service`.

Nie jest to restart systemu Raspberry Pi ani osobny restart MediaMTX. Przycisk nie ma osobnego potwierdzenia w tej wersji panelu.

1. Zabezpiecz model i upewnij się, że wybrano właściwą platformę.
2. Naciśnij **Restart Service** i poczekaj na komunikat w konsoli.
3. Po chwili sprawdź status oraz **App Logs**.
4. Ponownie sprawdź rzeczywiste sterowanie lub panel MCS przed wznowieniem pracy.

Komunikat o wysłaniu restartu nie jest testem zdrowia programu po ponownym uruchomieniu. Jeżeli usługa wraca do błędu, odczytaj logi zamiast powtarzać restart w pętli.

Reboot Pi — restart całego Raspberry Pi

Reboot Pi / Restart RPi wysyła polecenie restartu systemu przez SSH po potwierdzeniu. Przerywa połączenia, obraz i działające programy na docelowym Pi.

1. Zakończ jazdę i upewnij się, że cel to właściwy host.
2. Naciśnij Reboot Pi i potwierdź zamiar.
3. Poczekaj na ponowne uruchomienie; chwilowy stan Offline jest oczekiwany.
4. Ponów Test Connection, sprawdź znaczniki usług i logi.

Reboot command sent oznacza wysłane polecenie, nie zakończony restart. Jeżeli host nie wraca, sprawdź zasilanie, sieć i ewentualną zmianę adresu.

App Logs i Build Logs

App Logs / Logi aplikacji pobiera ostatnie 100 linii: z dziennika `usb_rc.service` dla MCS albo z kontenera `rcsim_industrial` dla Docker. To jednorazowe pobranie, nie nieprzerwany strumień dziennika z hosta. Aby sprawdzić nowy błąd, uruchom pobranie ponownie. Aktualna ścieżka odczytu zwraca standardowe wyjście polecenia; część informacji dostępnych na standardowym strumieniu błędów może nie trafić do panelu. Pusty wynik nie dowodzi braku problemów.

Build Logs / Logi budowania pobiera zapis budowania/instalacji. Dla Docker jest to plik `~/docker_build.log` na Raspberry Pi. W trybie MCS odczytywany jest `~/rpi_setup_mcs.log`. Panel próbuje wyłączyć przycisk po wybraniu MCS, lecz okresowe odświeżanie statusu może ponownie go uaktywnić. Jeżeli przycisk jest dostępny, odczyt nadal wymaga obecności właściwego pliku na hoście. Nie szukaj tam automatycznie logów każdej aktualizacji: plik musi istnieć i odpowiadać właściwej operacji.

1. Po nieudanym uruchomieniu wybierz App Logs; po błędzie budowania wybierz Build Logs, jeśli dostępny.
2. Znajdź pierwszy konkretny błąd oraz komunikaty bezpośrednio go poprzedzające.
3. Zachowaj treść przed czyszczeniem konsoli; zanotuj platformę i wykonywaną operację.
4. Nie traktuj ostatniego zdania o niepowodzeniu jako jedynej informacji potrzebnej do naprawy.

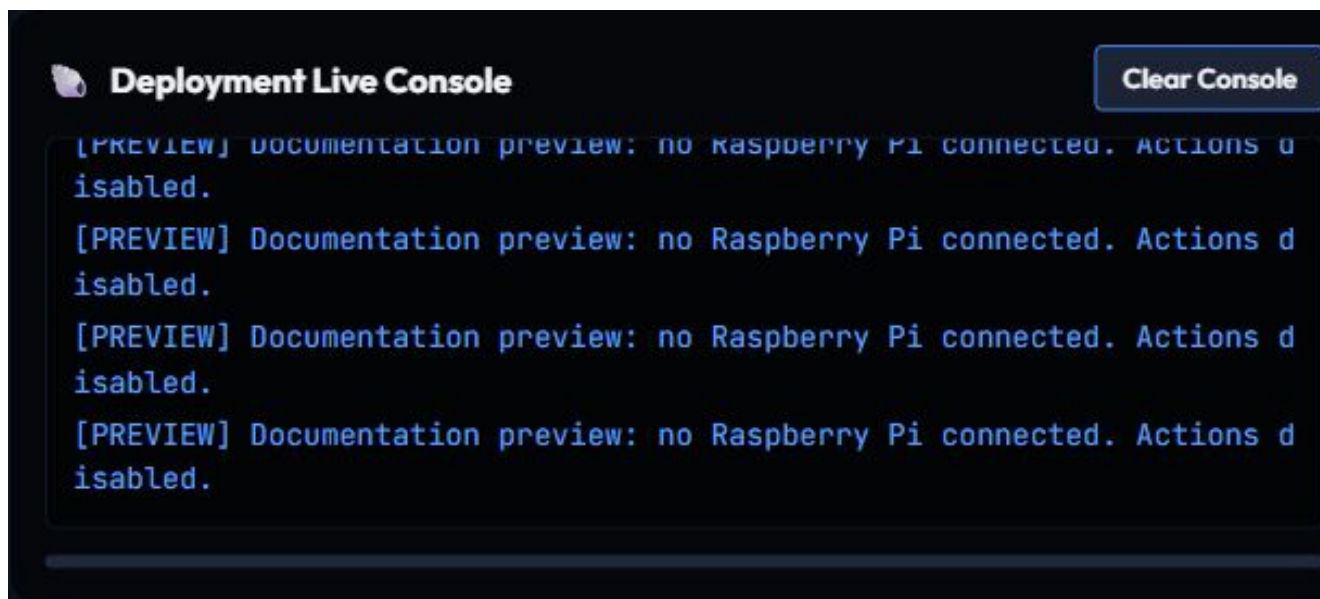
Diagnostics — zestaw odczytów systemowych

Diagnostics / Diagnostyka pobiera informacje o czasie pracy, dysku, pamięci, kontenerach Docker, stanie i ostatnich 20 liniach MediaMTX, ograniczeniach zasilania/temperatury, temperaturze CPU i stanie Tailscale. Wyniki pojawiają się kolejno w konsoli, z nazwą sprawdzanego obszaru.

Zestaw jest obecnie stały i obejmuje Docker/MediaMTX również wtedy, gdy wybrano MCS. Brak takiej usługi w instalacji MCS nie oznacza automatycznie uszkodzenia MCS. Narzędzie nie ocenia osobno kodu wyjścia każdej z tych komend i może zakończyć zestaw komunikatem OK mimo błędu pojedynczego odczytu. Puste sekcje również wymagają sprawdzenia; nie oznaczają poprawnego testu.

1. Uruchom Diagnostics po sprawdzeniu SSH.
 2. Porównaj sekcje z wyposażeniem i sposobem instalacji hosta.
 3. Przy braku miejsca sprawdź, co zajmuje dysk; przy problemie MediaMTX przejdź do jego logów.
 4. Przy błędzie sterowania sprawdź dodatkowo App Logs i faktyczny odbiór kanałów.
- Diagnostyka nie testuje fizycznie serw, failsafe odbiornika, czujników ani opóźnienia jazdy.

Konsola, postęp i zamknięcie narzędzia



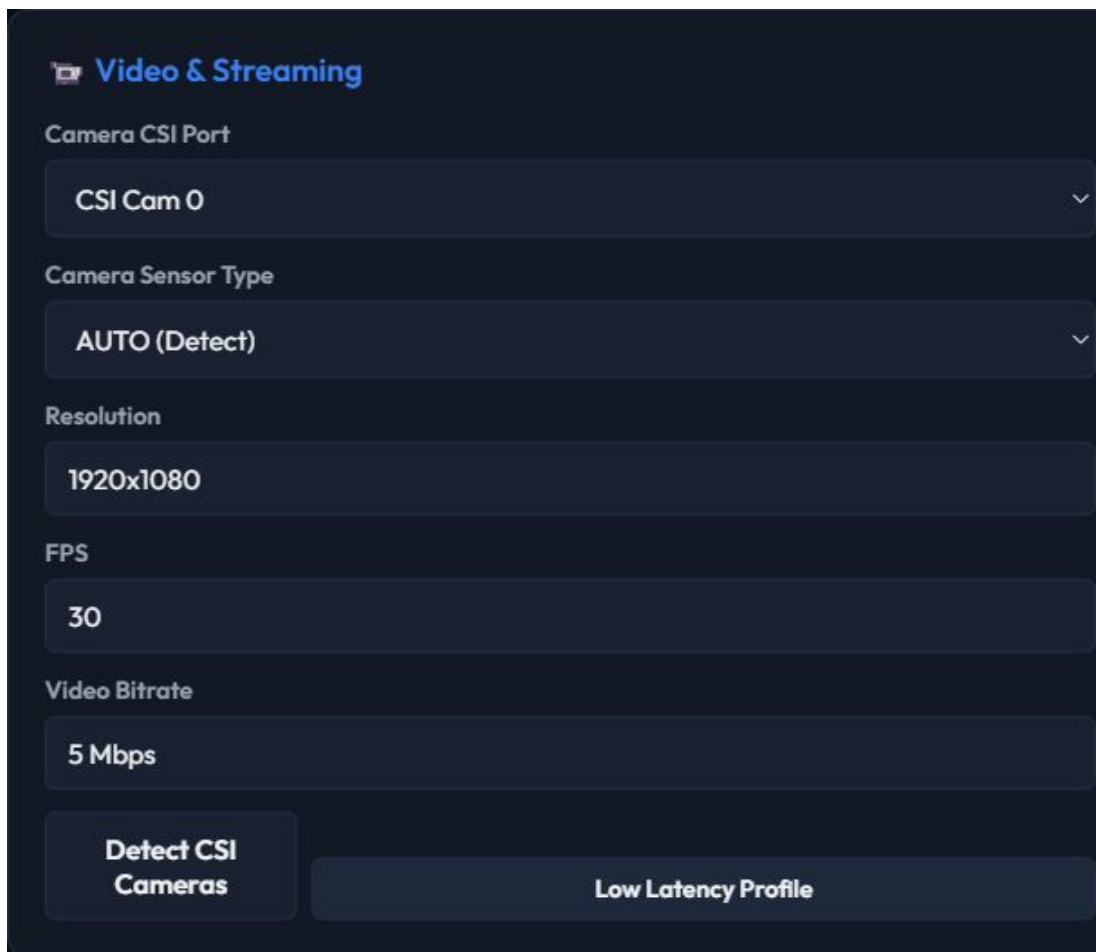
Konsola przed uruchomieniem zadań; ilustracja nie przedstawia wyniku wdrożenia

Konsola wyświetla komunikaty zadań i ich wyniki. Pasek postępu opisuje etapy operacji; nie jest precyzyjnym pomiarem czasu pozostałego do końca. Przy dłuższym kroku sprawdź nowe komunikaty zamiast uruchamiać tę samą operację ponownie. Odpowiedź na kliknięcie przycisku oznacza uruchomienie zadania w tle, nie jego powodzenie. Poczekaj na końcowy wynik konkretnej operacji.

Clear Console / Wyczyść konsolę usuwa widoczne wpisy z bieżącego widoku. Nie kasuje dzienników na Raspberry Pi, nie resetuje błędów usługi i nie przerywa zadania. Nie traktuj tego przycisku jako zatrzymania wdrożenia.

Shutdown Server / Wyłącz Serwer na górze zamyka lokalny serwer Deployment Tool po potwierdzeniu. Nie wyłącza systemu Raspberry Pi i nie jest poleceniem zamknięcia pojazdu. Po tej operacji panel przestaje odpowiadać, dopóki ponownie nie uruchomisz narzędzia. Nie używaj go w trakcie aktualizacji jako przycisku anulowania. W tym panelu nie ma osobnego polecenia zdalnego wyłączenia Pi ani instalacji AI/Hailo.

8. Wideo i pozostałe narzędzia obsługowe



Video & Streaming

Camera CSI Port

CSI Cam 0

Camera Sensor Type

AUTO (Detect)

Resolution

1920x1080

FPS

30

Video Bitrate

5 Mbps

Detect CSI Cameras

Low Latency Profile

Podgląd karty Video & Streaming; bez testu kamery

Video & Streaming — pola kamery

Karta jest widoczna dla platformy Docker; po wyborze MCS zostaje ukryta razem z kartą RTK. Nie jest odtwarzaczem wideo i nie pokazuje na żywo obrazu z kamery.

- **Camera CSI Port:** CSI Cam 0, CSI Cam 1, /dev/video0 albo /dev/video1.
- **Camera Sensor Type:** AUTO (Detect), imx219, imx708, ov5647, imx477 albo imx519.
- **Resolution:** szerokość i wysokość, np. 1920x1080, z literą x pomiędzy liczbami.
- **FPS:** żądana liczba klatek na sekundę, np. 30.
- **Video Bitrate:** żądana przepływność, np. 5 Mbps.

To żądane ustawienia. Faktyczna kombinacja zależy od kamery, sterownika i usługi na Pi. Zmiana wartości w formularzu nie potwierdza nowego strumienia i nie jest pomiarem rzeczywistego FPS. Zachowaj dotychczasowe nastawy przed zmianą kilku parametrów.

Detect CSI Cameras

Detect CSI Cameras / Wykryj kamery CSI łączy się z Pi i uruchamia wykrywanie kamer przez system.

Przeczytaj wynik w konsoli i porównaj z faktycznie zamontowanym sensorem. Funkcja nie jest uniwersalnym skanerem wszystkich źródeł USB, IP ani przechwytywania HDMI. Nie ustawia automatycznie całego profilu transmisji na podstawie nazwy kamery. Jeśli wynik jest pusty, sprawdź narzędzia kamery na hoście, taśmę, złącze i zasilanie.

Low Latency Profile

Low Latency Profile / Profil niskiej latencji wpisuje **800x600, 30 FPS i 2 Mbps**, a następnie zapisuje ustawienia narzędzia. Nie wysyła samodzielnie konfiguracji do działającej kamery. Nie jest gwarancją zmierzonej latencji i nie poprawia łącza radiowego. Po jego wyborze wykonaj odpowiednią aktualizację konfiguracji kamery i sprawdź obraz w GCS.

Cam Config Only — zastosuj ustawienia strumienia

1. Sprawdź adres Pi i rzeczywisty wariant wdrożenia.
2. Ustaw Resolution, FPS i Video Bitrate albo użyj Low Latency Profile.
3. Naciśnij **Cam Config Only / Tylko konfiguracja kamery**.
4. Poczekać na końcowy wynik. Usługa obrazu zostanie zrestartowana, więc strumień może zniknąć na chwilę.
5. Sprawdź wskaźnik Video, logi MediaMTX i rzeczywisty obraz w GCS.

Ta szybka operacja generuje konfigurację MediaMTX, zastępuje `/etc/mediamtx.yml` i restartuje MediaMTX. Uwzględnia rozdzielczość, FPS i bitrate; nie jest pełną aktualizacją sterowników ani ustawień systemowych kamery. W szczególności nie traktuj jej jako potwierdzenia zastosowania każdego wyboru CSI Port i Sensor Type. Generowany strumień korzysta ze źródła Raspberry Pi Camera; samo wybranie `/dev/video0` nie potwierdza skonfigurowania kamery USB.

Backup Docker Vol — co rzeczywiście obejmuje kopia

Mimo nazwy **Backup Docker Vol / Kopia Docker Vol** bieżąca funkcja archiwizuje katalog `rccsim_project` w katalogu domowym użytkownika Raspberry Pi, a następnie pobiera archiwum na komputer uruchamiający Deployment Tool. Nie jest to automatyczna kopia wszystkich woluminów Docker, obrazów kontenerów ani całej karty SD. W trybie webowym zapisuje plik `rccsim_docker_backup.tar.gz` w katalogu domowym lokalnego użytkownika narzędzia. Nie trafia on automatycznie do folderu pobierania telefonu, jeśli przeglądarka działa na telefonie. Kolejna kopia używa tej samej nazwy; zachowaj poprzedni plik pod inną nazwą, jeśli jest potrzebny.

1. Sprawdź docelowy host i konto oraz dostępne miejsce po obu stronach.
2. Uruchom Backup Docker Vol i poczekaj na zakończenie pobierania.
3. Odczytaj ścieżkę pliku z końcowego komunikatu.
4. Sprawdź obecność i zawartość archiwum przed uznaniem kopii za przydatną do odtworzenia.

Panel nie udostępnia osobnego przycisku automatycznego odtworzenia tej kopii. Po wybraniu MCS panel próbuje wyłączyć przycisk kopii, ale odświeżanie statusu może ponownie go uaktywnić. Jego dostępność nie oznacza obsługi kopii MCS: operacja nadal używa katalogu `rccsim_project` właściwego dla opisanej ścieżki Docker.

Aktualizacja a instalacja od nowa

Docker Rebuild & Update uruchamia ścieżkę wdrożenia z pominięciem części konfiguracji systemu. **Hot Deploy Code (~30s)** służy szybkiej podmianie kodu i konfiguracji w istniejącym środowisku; dla Docker wymaga wcześniej utworzonego kontenera. Liczba „~30s” w nazwie jest orientacyjna, nie gwarantuje czasu ani braku przerwy w sterowaniu. **Fast Deploy (Skip OS Config)** oznacza pomijanie konfiguracji systemowej, nie sprawdzenie, że system jej nie potrzebuje. Jeśli zmieniły się zależności, sprzęt lub wymagania systemu, nie zakładaj, że szybka podmiana wystarczy. **FULL DEPLOYMENT** jest szerszą operacją instalacji i konfiguracji, opisaną w procedurze wdrożenia. Wybór tych przycisków nie uruchamia treningu AI ani nie daje osobnej obsługi modeli Hailo w tym panelu.

9. Rozwiązywanie problemów i źródła

- **RPI Online, usługa nieaktywna:** sprawdź SSH, platformę i App Logs; ping nie testuje programu.
- **RPI Offline, ale SSH działa poza panelem:** host może nie odpowiadać na ping; sprawdź Test Connection.
- **MCS pokazuje dwa identyczne statusy:** Core/Web odnoszą się tu do jednej usługi, a nie dwóch testów.
- **Restart Service nie naprawia obrazu:** restart głównej aplikacji nie restartuje MediaMTX; sprawdź jego diagnostykę i konfigurację.
- **Diagnostics kończy się OK, ale część sekcji jest pusta:** odczytaj indywidualne wyniki i logi; końcowe OK nie gwarantuje powodzenia każdej komendy.
- **Brak App Logs:** upewnij się, że wybrano właściwą platformę; pusty odczyt nie wyklucza komunikatów na strumieniu błędów.
- **Dostępność Build Logs lub kopii zmienia się w MCS:** odświeżanie statusu może nadpisywać blokadę przycisków. Build Logs odczytuje plik MCS, ale kopia nadal dotyczy katalogu rcsim_project; aktywny przycisk nie potwierdza zgodności funkcji z platformą.
- **Low Latency Profile nie zmienił obrazu:** profil zmienia zapis formularza; zastosuj konfigurację kamery i sprawdź strumień w GCS.
- **Kamera USB nie działa po Cam Config Only:** szybki generator używa źródła Raspberry Pi Camera; sam wybór portu USB nie konfiguruje takiego źródła.
- **Nie widzę kopii na telefonie:** plik jest pobierany na komputer z serwerem Deployment Tool, pod ścieżką podaną w konsoli.
- **Po Shutdown Server strona nie działa:** ponownie uruchom lokalne narzędzie; ten przycisk zamyka jego server.
- **Operacja trwa dłużej niż opis przycisku:** sprawdź etap w konsoli i poczekaj na wynik, nie rozpoczynaj równoległego wdrożenia.

Ilustracje służą rozpoznaniu elementów panelu; nie potwierdzają wdrożenia na fizycznym Raspberry Pi. Źródła odczytane 14 października 2026 r., względne wobec katalogu `RCSIM_deployment_tool/RCSIM_deployment_tool/`:

- `web/templates/index.html` oraz `web/static/js/main.js`: karty, przyciski, profil kamery, potwierdzenia, konsola i statusy.
- `web_server.py`, linie 310–364, 446–624 i 638–645: monitoring, zadania i zamknięcie lokalnego serwera.
- `core/deployment_logic.py`: konfiguracja strumienia, wykrywanie kamer, instalacja, aktualizacje, kopia katalogu, diagnostyka, restart i logi. `core/service_monitor.py` dotyczy alternatywnego interfejsu desktopowego.