

Tier 3: ELRS/mLRS, CRSF/MAVLink i mody EdgeTX/ExpressLRS

Instrukcja oparta na researchu kodu źródłowego. Procedury, warianty i ograniczenia opisano oddzielnie od deklaracji wymagających testu fizycznego zestawu.

Spis treści

1. Co obejmuje Tier 3	3
2. Najpierw wybierz rzeczywisty tor sygnału	3
3. Wymagania i karta konfiguracji	3
4. CRSF Direct w GCS	4
4.1. Nomad — przygotowanie konkretnego wariantu CRSF	4
4.2. GCS — USB, TCP i UDP krok po kroku	5
4.3. Backpack CRWF i Airlink — nie myl protokołów	6
5. MAVLink RF	6
5.1. Połączenie szeregowe i Airport	7
5.2. Firmware mLRs/ELRS i odzyskanie modułu	7
6. Mod EdgeTX: CRSF Trainer przez USB-VCP	7
6.1. Instalacja i powrót do własnego firmware	8
6.2. Menu i tor trenera	8
7. Mod ExpressLRS ER Series: przykład ER5C V2	8
7.1. Montaż ER5C V2 GPS + IMU	9
7.2. Wgranie wsadu ER5C V2	9
7.3. Funkcje wyjść i pierwsze sensory	10
8. Diagnostyka	10
8.1. Kierownica, pedały i Force Feedback	11
8.2. Próba całego zestawu przed jazdą	11
9. Granice potwierdzenia	12
10. Źródła researchu	12

Podręcznik użytkownika: montaż, firmware, konfiguracja i próby stanowiskowe. Stan: 14 października 2026. Podstawą są lokalne implementacje, patche, narzędzia testowe i dokumentacja konkretnych wsadów. Nie przeprowadzono testów fizycznego radia ani kompilacji wszystkich firmware.

1. Co obejmuje Tier 3

Tier 3 to sterowanie i telemetria przez systemy modelarskie ELRS/mLRS, z interfejsami CRSF lub MAVLink oraz projektowymi modyfikacjami EdgeTX i ExpressLRS, m.in. dla ER5C V2. Nazwa nie oznacza jednego zestawu nadajnik–odbiornik ani jednego baudrate.

Komputer pokładowy Raspberry Pi nie jest wymaganiem tego Tiera. Gdy radio jest jednym z interfejsów platformy Raspberry Pi, kompletna platforma należy do Tier 4; jej stronę radiową nadal konfiguruje się według właściwego wariantu z tej instrukcji.

PPM z karty dźwiękowej należy do Tier 0. Mostek Arduino/RP2350 Digital to PPM należy do Tier 1. Nie należy nazywać aparatury z modem CRSF przez USB-VCP mostkiem PPM.

2. Najpierw wybierz rzeczywisty tor sygnału

- **CRSF Direct:** GCS → zgodny interfejs CRSF → moduł radiowy → odbiornik. Dane zwrotne zależą od sensora i firmware całego toru.
- **MAVLink RF:** GCS → port lub mostek sieciowy → radio ELRS/mLRS → endpoint MAVLink. Samo radio nie tworzy danych sensorów.
- **EdgeTX CRSF VCP mod:** GCS → USB-VCP aparatury → trener/mikser → moduł RF → odbiornik. FullDuplex dodaje zwrotne odbicie telemetrii do USB.
- **ExpressLRS ER Series mod:** zgodny odbiornik z dodatkowym GPS/IMU i zmienionymi funkcjami pinów.

CRSF jest formatem ramek/interfejsem, MAVLink protokołem wiadomości, a ELRS/mLRS określają system radiowy i jego firmware. Zgodność na jednym odcinku nie dowodzi zgodności całego toru.

Wideo nie jest automatycznym składnikiem CRSF/MAVLink. Osobny system FPV i grabber HDMI/USB można skonfigurować jako źródło obrazu w GCS. Widoczny obraz nie potwierdza działania sterowania ani telemetrii.

3. Wymagania i karta konfiguracji

Przed podłączeniem zanotuj:

- model nadajnika/modułu i odbiornika;
- nazwę oraz wersję firmware każdego urządzenia;
- wariant GCS: CRSF Direct albo MAVLink RF;
- port PC i prędkość danego odcinka USB/UART;
- mapowanie skrętu, gazu, uzbrojenia i dodatkowych kanałów;
- źródło każdego odczytu: GPS, IMU, bateria, statystyki łącza;
- sposób przejęcia kontroli i zachowanie odbiornika po utracie łącza.

Nie zmieniaj jednocześnie firmware, pinoutu, baudrate i mapowania. Po każdej zmianie sprawdzaj osobno najbliższy odcinek toru.

4. CRSF Direct w GCS

CRSFTransceiver przyjmuje port, baudrate, częstotliwość nadawania i transport. Oprócz portu szeregowego implementuje warianty TCP, UDP i Backpack UDP. Ich obecność w klasie nie oznacza, że dowolny moduł radiowy obsłuży każdy z nich.

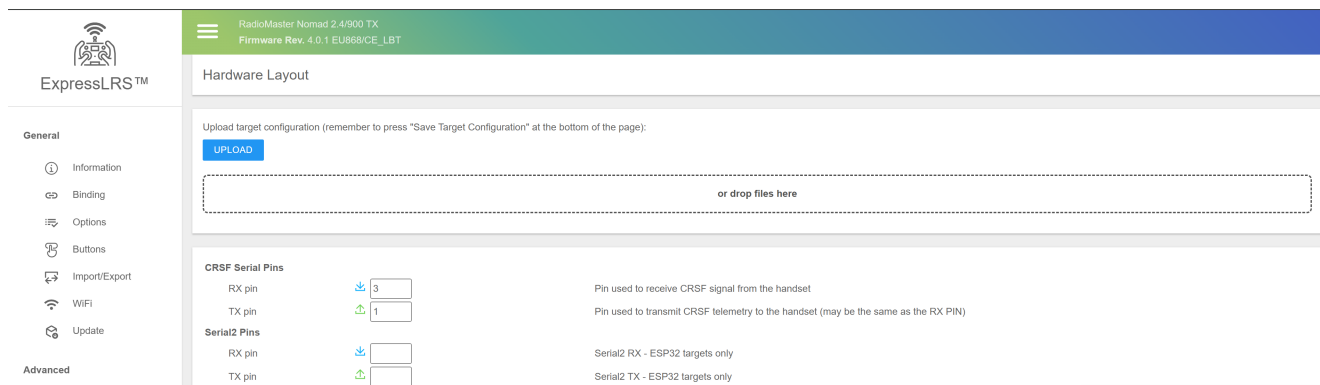
Aktywna strategia GCS CRSF nadaje z docelową częstotliwością 100 Hz. Sama klasa transceivera ma inne domyślne argumenty: 420000 bps i 250 Hz; fabryka GCS ma fallback prędkości 400000 bps. Żadna z tych liczb nie jest uniwersalnym wymaganiem radia. Mod EdgeTX USB-VCP opisany dalej używa 115200 bps. Dobierz prędkość do konkretnego interfejsu, a nie do samej nazwy ELRS.

Pierwszy test wykonaj bez możliwości uruchomienia napędu:

1. Uruchom GCS według osobnej instrukcji GCS i wybierz profil pojazdu.
2. Wybierz właściwy tryb CRSF/Direct ELRS oraz zidentyfikowany port urządzenia.
3. Ustaw zgodny baudrate i pozostaw model rozbrojony.
4. Sprawdź kanały w GCS, następnie monitor kanałów aparatury i odbiornik.
5. Porównaj środek, zakres oraz kierunek skrętu i gazu. Nie zakładaj, że gaz neutralny zawsze odpowiada tej samej wartości w każdym modelu.
6. Dopiero po zgodności kanałów sprawdź uzbrojenie i przejęcie sterowania.
7. Przerwij połączenie PC, a osobno połączenie RF, i sprawdź rzeczywiste zachowanie wyjść odbiornika.

CRSF Direct po rozbrojeniu nadaje kanały neutralne, ale CH5/AUX1 ustawia nisko. Po uzbrojeniu CH5 może być ustawiony wysoko; domyślna wartość środkowa tego kanału jest zastępowana wysoką. Uwzględnij tę rolę przed przypisaniem CH5 np. do innego urządzenia. Nie jest to odcięcie zasilania ESC ani gwarancja zgodnej semantyki ARM w każdym odbiorniku.

4.1. Nomad — przygotowanie konkretnego wariantu CRSF



Przykład WebUI Nomad: ExpressLRS 4.0.1, RX pin 3 i TX pin 1

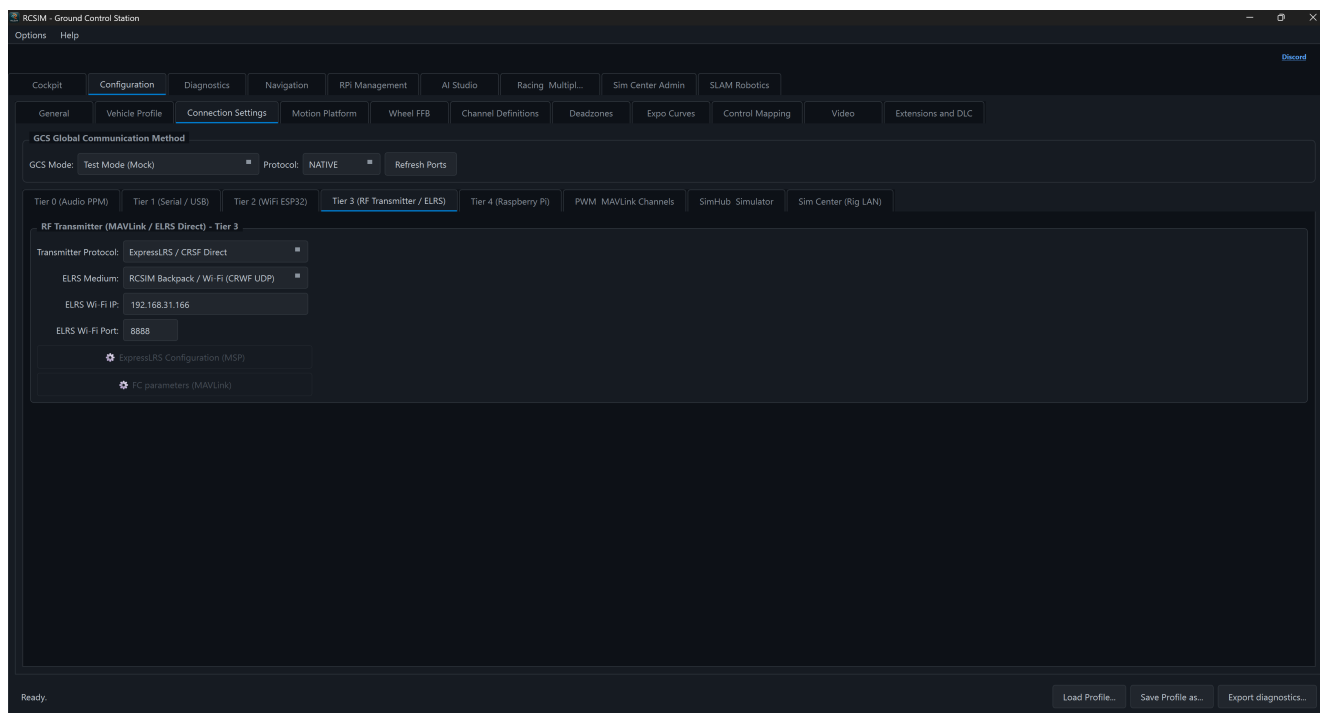
Ilustracja projektu przedstawia **RadioMaster Nomad 2.4/900 TX**, firmware **4.0.1 EU868/CE_LBT**. W **Hardware Layout** → **CRSF Serial Pins** zapisano **RX pin 3, TX pin 1**. Notatka projektu wskazuje zmianę z 4/4 dla tego zestawu. Nie jest to schemat lutowania CP2102 ani uniwersalny pinout wszystkich rewizji Nomad.

1. Najpierw potwierdź dokładny model i aktualny firmware modułu oraz sposób wyprowadzenia jego USB/UART.
2. Zachowaj istniejącą konfigurację sprzętową przed zmianą pinów.
3. Tylko dla zgodnego wariantu z ilustracji ustaw RX 3 i TX 1 w CRSF Serial Pins.

4. Użyj **Save Target Configuration** na dole strony, następnie uruchom moduł ponownie zgodnie z jego komunikatem.
5. Sprawdź oddzielnie odbiór kanałów i powrót telemetry przez wybrany interfejs.

Przy zewnętrznym adapterze UART sygnał TX hosta prowadzi do RX urządzenia, a RX hosta do TX urządzenia, z poprawnie połączoną masą. Zasilanie modułu i napięcia logiczne muszą odpowiadać specyfikacji konkretnego urządzenia. Nie łącz wyjścia RS-232 z wejściem UART TTL i nie zasilaj nadajnika dużej mocy z przypadkowego pinu adaptera. Ten projekt nie dostarcza zweryfikowanego fizycznego schematu każdego takiego adaptera; bez identyfikacji pinów płytki nie wykonuj połączenia na podstawie numerów z WebUI. Przed nadawaniem zamontuj właściwą antenę i ustaw parametry RF odpowiednie do używanego sprzętu i regionu.

4.2. GCS — USB, TCP i UDP krok po kroku



Ustawienia Tier 3: przykład CRWF UDP; na rzucie aktywny pozostaje Test Mode (Mock)

Otwórz **Configuration** → **Connection Settings** → **Tier 3 (RF Transmitter / ELRS)**. Samo otwarcie karty Tier 3 nie przełącza globalnego **GCS Mode** z Test Mode. Wybierz rzeczywiste połączenie CRSF Direct w GCS Mode i **ExpressLRS / CRSF Direct** w **Transmitter Protocol**.

- **USB Serial (COM)**: użyj Refresh Ports, wybierz faktyczny Port i zgodny Baud. Zamknij inne programy używające tego COM.
 - **Airlink / Wi-Fi (TCP)**: wpisz adres mostka w ELRS Wi-Fi IP i jego port serwera TCP w ELRS Wi-Fi Port.
 - **Airlink / Wi-Fi (UDP)**: wpisz adres i port mostka odbierającego surowe ramki CRSF przez UDP.
 - **RCSIM Backpack / Wi-Fi (CRWF UDP)**: używa osobnego formatu CRWF, opisanego poniżej; nie zastępuje zwykłego UDP.
1. Wybierz medium zgodne z firmware drugiego końca, nie tylko z dostępną siecią.
 2. Dla USB sprawdź port oraz prędkość; dla sieci IP i port. Domyślne 192.168.4.1 i 8888 to przykłady programu.
 3. Zakończ edycję pól, zachowaj profil i wróć do **Cockpit**.
 4. Wybierz **Manual (Driver)**, pozostaw rozbrojenie i użyj **Connect**.

5. Sprawdź monitor kanałów, potem rzeczywisty odbiornik i powrót telemetry.
6. Jeżeli Connect tylko otwiera UDP, nie traktuj tego jako potwierdzenia odpowiedzi drugiego urządzenia.

ExpressLRS Configuration (MSP) wymaga aktywnego zgodnego CRSF Direct. Udostępnia konfigurację tylko wtedy, gdy moduł odpowiada na wymagane polecenia. Przycisk nie instaluje firmware i nie jest dostępny dla transportu Backpack CRWF. **Raw IMU via CRSF (0x13)** otwiera informacje o modzie i wymaganiach; nie dodaje sensora do fabrycznego odbiornika.

4.3. Backpack CRWF i Airlink — nie myl protokołów

CRWF wymaga **zmodyfikowanego RCSIM Backpack**, zgodnego **ExpressLRS 4**, **Link Mode: Normal**, trybu trenera EdgeTX **Master/CRSF** oraz miksów wejść **TR** na właściwych kanałach. To inny wariant trenera niż opis USB-VCP dalej; nie przenoś między nimi nazw trybów bez sprawdzenia. Zwykły fabryczny Backpack nie staje się mostkiem CRWF przez samo wybranie tej opcji w GCS.

1. Potwierdź, że wgrano pakiet Backpack przeznaczony dla dokładnego targetu urządzenia i zgodny z CRWF.
2. Przygotuj model EdgeTX: Master/CRSF, miks TR każdego używanego kanału i przełącznik przejęcia sterowania.
3. Sprawdź połączenie Wi-Fi, adres Backpacka i port nasłuchu; wpisz je w GCS.
4. Wybierz CRWF UDP, połącz i sprawdź kanały w monitorze radia przed próbą modelu.
5. Sprawdź powrót telemetry. Po utracie łącza albo przełączeniu head trackingu rozłącz i połącz RCSIM ponownie.

CRWF przenosi kanały i telemetry, nie polecenia konfiguracji CRSF/MSP ani MAVLink. Nie należy oczekiwać działania konfiguratora parametrów przez ten transport. Sesja transportu jest identyfikatorem połączenia, nie uwierzytelnieniem użytkownika. Kod próbuje odnowić sesję po przerwie nadawania lub utracie dotychczas odbieranej telemetry; nie zastępuje to próby failsafe i ponownego połączenia z punktu widzenia operatora. W lokalnym projekcie nie znaleziono kompletnego, dopasowanego pakietu firmware Backpack ani uniwersalnej procedury jego odtworzenia. Do instalacji lub recovery wymagany jest pakiet i instrukcja dokładnego targetu; nie używaj binarki ER5C albo moda EdgeTX.

5. MAVLink RF

Strategia MAVLinkRFStrategy używa biblioteki pymavlink i wymusza MAVLink 2.0. W kodzie istnieją osobne mechanizmy parametrów, kalibracji, dekodowania telemetry oraz identyfikacji kontrolera.

Dobierając konfigurację:

1. Sprawdź, czy docelowe urządzenie wysyła heartbeat i wiadomości MAVLink, a nie wyłącznie CRSF.
2. Dopasuj parametry portu do aktywnego firmware radiowego.
3. Odróżnij identyfikator modułu radiowego od właściwego pojazdu/kontrolera. Kod zawiera filtrowanie identyfikatorów modułów mLRS podczas routingu.
4. Zweryfikuj dostępność parametrów i sensorów na podstawie odebranych wiadomości, nie samej obecności portu COM.
5. Skontroluj reakcję na zerwanie portu, zanik RF i utratę heartbeat jako oddzielne przypadki.

Nie ma jednej prędkości „MAVLink dla wszystkich”. Prędkości odcinków po obu stronach modułu mogą być różne, jeśli jego firmware wykonuje odpowiednie mostkowanie. Każdy odcinek musi być zgodny ze swoim endpointem.

5.1. Połączenie szeregowe i Airport

W **Transmitter Protocol** wybierz **MAVLink RF (Airport / Telemetry)** i globalny tryb MAVLink RF. **MAVLink Medium** wybiera Serial, Airport Wi-Fi TCP albo UDP. Serial wymaga Port/Baud; TCP i UDP pokazują **MAVLink Airport IP** oraz port. Dla TCP pole oznacza port serwera Airport; domyślne 5760 jest wartością programu, nie regułą każdego mostka. Dla UDP **Destination Port (Backpack Listen Port)** oznacza port odbierający pakiety na urządzeniu, a **Local UDP Port (Backpack Send Port)** port komputera, na który wraca telemetria. Przykład zgodnej pary to docelowy 14555 i lokalny 14550, jeżeli tak skonfigurowano Backpack. Wartość lokalna 0 pozostawia automatyczny wybór portu; nie używaj jej, gdy urządzenie wysyła wyłącznie na konkretny port PC.

SysID dotyczy docelowego kontrolera pojazdu, nie numeru COM ani identyfikatora samego modułu radiowego. Wartość 0 uruchamia automatyczny dobór celu; nie przyjmuj, że każde urządzenie używa SysID 1 lub 10.

ArduPilot Telemetry Profile ogranicza stawki wybranych wiadomości po potwierdzeniu kontrolera; nie zwiększa pasma RF i nie gwarantuje dostępności każdej wielkości.

1. Zapisz porty i prędkości obu stron mostka oraz SysID pojazdu.
2. Ustaw medium i odpowiednie pola, połącz z kokpitu i poczekaj na heartbeat pojazdu.
3. Sprawdź, czy rozpoznano kontroler, a nie tylko moduł mLRS.
4. Otwórz **FC parameters (MAVLink)** dopiero po uzyskaniu aktywnego, właściwego celu.
5. Odczytaj parametry przed zmianą; zachowaj dotychczasowe wartości i zmieniaj pojedyncze ustawienia.
6. Sprawdź kierunki i reakcję RC override na zabezpieczonym stanowisku, potem potwierdzenie ARM/DISARM kontrolera.

GCS kolejkuje najnowsze sterowanie do nadawania 20 Hz; strategia przyjmuje 1–18 wartości RC override. To nie opis ośmiokanałowego wyjścia MCS: oba programy mają odrębne implementacje. ARM/DISARM w GCS jest komendą MAVLink do właściwego endpointu i wymaga uprawnień sterowania; nie zakładaj uzbrojenia ani wyłączenia silnika bez odpowiedzi kontrolera i obserwacji modelu. Przerwanie RC override, utrata heartbeat i failsafe autopilota również wymagają osobnej konfiguracji i próby.

5.2. Firmware mLRS/ELRS i odzyskanie modułu

Nazwa Nomad nie wybiera automatycznie ELRS lub mLRS. Sprawdź firmware nadajnika i odbiornika, pasmo, target oraz obsługiwany interfejs przed próbą zmiany protokołu. Lokalny projekt zawiera integrację GCS, lecz nie kompletny zestaw instalacyjny mLRS dla każdej rewizji Nomad/XR4. Nie ma podstaw do wskazania jednej komendy flash, adresu pamięci lub sekwencji BOOT dla tych wszystkich urządzeń. Przed aktualizacją zachowaj ustawienia, przygotuj oryginalny firmware dokładnego targetu i jego procedurę recovery. Po aktualizacji ponownie sprawdź zgodność pary radiowej, binding, porty i failsafe. Nie wgrywaj moda odbiornika ER5C V2 do nadajnika, Backpacka ani odbiornika z innym układem MCU.

6. Mod EdgeTX: CRSF Trainer przez USB-VCP

Pliki projektu znajdują się w katalogu:

```
RCSIM_PC/pc_app/ESP32Arduino/RCSIM_Arduino_ESP32/EdgeTX CRSF VCP mod/
```

Lokalna instrukcja rozróżnia wsady FullDuplex i starsze simplex. Według tej instrukcji MT12 i Pocket mają status przetestowanych przez autora moda, a część innych modeli ma status eksperymentalny. Niniejszy research nie stanowi ponownego testu sprzętowego ani potwierdzenia binarnej zawartości każdego wsadu.

Przed flashowaniem wykonaj kopię własnego firmware, ustawień modelu i karty SD. Wybierz wsad odpowiadający dokładnemu modelowi aparatury; nazwa innego radia nie jest kompatybilnym zamiennikiem.

6.1. Instalacja i powrót do własnego firmware

W katalogu moda dostępne są pliki EdgeTX_v2.10_MODEL_CRSF_VCP_FullDuplex.bin dla MT12, Pocket, Boxer, TX16S, TX12, TX12MK2 i Zorro. TX12 i TX12MK2 to osobne pliki. Nie wybieraj po podobnym wyglądzie aparatury.

1. Zabezpiecz model, odłącz napęd i zachowaj kopię konfiguracji aparatury, firmware oraz karty SD.
2. Wybierz USB Storage (SD) albo czytnik karty; skopiuj właściwy .bin do FIRMWARE/ na karcie.
3. Bezpiecznie wysuń pamięć i uruchom bootloader zgodnie z instrukcją swojego modelu radia.
4. W bootloaderze wybierz **Write Firmware**, właściwy plik i potwierdzenie zapisu.
5. Poczekaj na zakończenie, wybierz **Exit** i sprawdź uruchomienie radia oraz wybrany model.
6. Przygotuj konfigurację trenera z następnego podrozdziału i wykonaj próbę monitora kanałów bez napędu.

Gest wejścia do bootloadera zależy od aparatury; opis dwóch trymerów w starszej notatce nie jest tu zaleceniem dla każdego modelu. Jeżeli nowy wsad działa nieprawidłowo, użyj bootloadera i zachowanego poprawnego wsadu dokładnego targetu, a ustawienia przywróć z własnej kopii. Jeżeli bootloader nie jest dostępny, użyj procedury recovery właściwej dla radia; ten pakiet nie stanowi uniwersalnej instrukcji DFU dla wszystkich siedmiu modeli.

6.2. Menu i tor trenera

Wymagania zapisane w narzędziu crsf_vcp_test.py:

- tryb USB aparatury: Serial/VCP;
- konfiguracja USB-VCP jako CRSF Trainer;
- tryb trenera modelu: CRSF;
- aktywna funkcja specjalna Trainer, aby kanały dotarły do miksera.

Nazwy i położenie pozycji menu zależą od konkretnej kompilacji. Po zmianie sprawdź monitor kanałów aparatury z wyłączonym RF i odłączonym napędem.

1. W **SYS** → **Radio Setup** ustaw **USB Mode: VCP** albo Ask i wybieraj VCP/Serial po podłączeniu kabla.
2. W **SYS** → **Hardware** przypisz **USB-VCP: CRSF Trainer**.
3. W **MDL** → **Model Setup** → **Trainer** ustaw **Mode: CRSF** i używane kanały, do CH1–CH16.
4. W **Special Functions** dodaj Trainer, Value Sticks/Axis, Enable ON, najlepiej pod świadomym przełącznikiem przejęcia.
5. Sprawdź miksery modelu, kierunki i zakresy, a potem monitor kanałów.
6. W GCS wybierz CRSF Direct, USB Serial, właściwy COM i 115200 bps. Sprawdź transmisję i powrót telemetry osobno.

Mod FullDuplex odbija istniejącą telemetry. Funkcja **Discover new sensors** aparatury, jeżeli dostępna w danej kompilacji, służy wykryciu odbieranych sensorów; nie tworzy pomiaru IMU lub GPS. Jeśli sterowanie działa bez danych zwrotnych, sprawdź FullDuplex zamiast legacy simplex oraz aktywną telemetry RF.

Oba narzędzia crsf_vcp_test.py i crsf_vcp_duplex_test.py aktywnie nadają kanały; nie są biernymi monitorami. Wariant duplex domyślnie nadaje 100 Hz i przemiata kanały 1–4 w pełnym zakresie, równocześnie odbierając ramki. Nie uruchamiaj ich na aktywnym modelu z założeniem, że tylko odczytują telemetry.

7. Mod ExpressLRS ER Series: przykład ER5C V2

Katalog źródeł i artefaktów:

Repozytorium zawiera instrukcje, pliki binarne i patch ExpressLRS v4.1.0. Patch m.in. rozszerza GPS, obsługę IMU i konfigurację pinów. Zastosowanie patcha do innej wersji upstream wymaga osobnej weryfikacji; obecność pliku nie oznacza udanego builda na dowolnym checkoutcie.

Instrukcja wariantu ER5C V2 opisuje:

- CH1 jako PWM skrętu;
- CH2 jako PWM ESC;
- CH3 jako wejście RX dla TX modułu GPS;
- CH4/CH5 jako SDA/SCL dla zgodnego IMU.

Ten podział dotyczy wskazanego wsadu i konfiguracji WebUI. Nie stosuj go automatycznie do ER4, ER8, fabrycznego ER5C V2 lub innej rewizji sprzętu. Zmiana CH4/CH5 na I2C oznacza zmianę ich roli względem zwykłych wyjść PWM. Zasilanie i poziomy logiczne sensorów sprawdź w dokumentacji użytych modułów.

Patch zawiera zmianę SERIAL_RX_ONLY, która w wariantcie GPS zwalnia pin TX dla funkcji PWM ESC. Nie usuwaj jej, zakładając, że nie wpływa na napęd.

7.1. Montaż ER5C V2 GPS + IMU

Poniższy pinout dotyczy wyłącznie ER5C V2 z opisanym modem:

- CH1/GPIO0, sygnał → serwo skrętu; zachowaj funkcję PWM.
- CH2/GPIO1, sygnał → ESC; zachowaj PWM, jeżeli to wyjście napędu.
- TXD GPS → sygnał CH3/GPIO3 odbiornika; RXD GPS pozostaje niepodłączony w wariantcie jednokierunkowym.
- SDA IMU → sygnał CH4/GPIO9; SCL IMU → sygnał CH5/GPIO10.
- GND odbiornika, GPS i IMU → wspólna masa.

Szyna + złączy serw może przenosić napięcie BEC, które nie jest bezpieczne dla każdego sensora. Zasil GPS i moduł IMU zgodnie z ich specyfikacją; surowy układ MPU nie jest automatycznie modułem tolerującym 5 V. Podciąganie I2C oraz sygnał UART muszą być zgodne z logiką 3,3 V. Odczytuj oznaczenia, nie kolory przewodów. Mocuj IMU sztywno i zanotuj orientację osi; luźna płytką może dawać szum i błędne siły FFB. Obsługiwane rodziny IMU to MPU6050/6500/9250/9255, adresy 0x68 lub 0x69; nie oznacza to obsługi dowolnego Grove IMU.

7.2. Wgranie wsadu ER5C V2

Pakiet łączony to **ELRS_V4.1_RadioMaster_ER5Cv2_RCSIM_GPS_IMU**: .bin.gz dla zgodnego WebUI, .bin dla odzyskiwania przez UART. GPS-only i IMU-only są odrębnymi wariantami; starej nazwy MPU9250_CRSF13 nie interpretuj jako pełnego opisu zawartości. Istnieją też binarki ER4/ER5A_C/ER6/ER6G/ER6GV/ER8/ER8G/ER8GV. Ich obecność nie potwierdza pinoutu ER5C V2 na tych płytkach; ER5 opiera się na ESP8285, a część ER6/ER8 na ESP32.

1. Zachowaj własne ustawienia odbiornika i oryginalny firmware dokładnego modelu.
2. Przy wyłączonym napędzie uruchom Wi-Fi odbiornika; lokalna instrukcja opisuje oczekiwanie około 60 s bez połączenia z TX.
3. Dla ustawień fabrycznych połącz z **ExpressLRS RX**, hasło **expresslrs**, otwórz <http://10.0.0.1> lub właściwy adres przydzielony odbiornikowi.
4. W **Update / Firmware Update** wybierz właściwy plik ER5Cv2 .bin.gz i uruchom Update.

5. Poczekać na końcowy wynik i restart; czas zależy od urządzenia. Nie odłączaj zasilania podczas zapisu.
6. Ponownie otwórz WebUI i potwierdź firmware oraz funkcje wyjść przed podłączeniem napędu.

Jeżeli potrzebne jest odzyskanie ER5C V2 przez UART, użyj adaptera TTL z logiką 3,3 V, TX adaptera → RX odbiornika i RX adaptera ← TX odbiornika oraz wspólnej masy. Odłącz na ten czas GPS od współdzielonego RX i ESC od współdzielonego TX/PWM. Ustal fizyczne punkty UART/BOOT dla konkretnej płytki; numer CH nie zastępuje dokumentacji adaptera i odbiornika. Przytrzymaj właściwy BOOT podczas włączania zgodnie z instrukcją ER5C V2, a po wykryciu portu wykonaj:

```
esptool.py --port COMX --baud 115200 --chip esp8266 write_flash -fm dout 0x000000  
ELRS_V4.1_RadioMaster_ER5Cv2_RCSIM_GPS_IMU.bin
```

Uruchom z katalogu pliku i zastąp COMX faktycznym portem. Ta komenda dotyczy surowego .bin ER5C V2, nie .gz, MCU ESP32, nadajnika Nomad ani bootloadera EdgeTX. Po sukcesie uruchom odbiornik normalnie, sprawdź konfigurację, a dopiero potem przywróć sensory i ESC. Nie wykonuj połączenia, jeżeli nie potwierdzono fizycznego dostępu do UART/BOOT i napięcie.

7.3. Funkcje wyjść i pierwsze sensory

1. W WebUI **Connections / PWM Pin Functions** ustaw Output 1 i 2 jako PWM zgodny z serwem/ESC; zacznij od 50 Hz, jeżeli urządzenia to obsługują.
2. Ustaw **Output 3: Serial RX** i główny **Serial Protocol: GPS**.
3. Ustaw **Output 4: I2C SDA**, **Output 5: I2C SCL** i użyj SAVE pod odpowiednimi tabelami.
4. Zrestartuj odbiornik i sprawdź, czy CH2 nadal jest PWM.
5. Zestaw zgodną parę radiową; sprawdź powrót GPS i IMU w GCS oddzielnie.

GPS musi już nadawać NMEA GGA/RMC/VTG lub obsługiwany UBX. Autowykrywanie sprawdza 115200, 9600, 38400, 57600, 230400 i 19200 bps. Wykryty baud nie oznacza fixa; pozycja wymaga odbioru satelitów i odpowiednich warunków. Jednokierunkowy GPS nie otrzymuje komend autokonfiguracji. Jeżeli ustawisz CH2 jako Serial TX i podłączysz go do RX GPS, tracisz na CH2 wyjście ESC. Nie próbuj jednocześnie używać tego samego pinu jako UART TX i PWM napędu.

Surowe IMU to ramka **0x13**, payload **18 bajtów**, źródło **crsf_raw**; obejmuje czas próbki, gyro, przyspieszenie i temperaturę, bez magnetometru. **0x1E** zawiera orientację, **0x02** GPS, **0x03** czas GPS. Nie szukaj przestarzałego dekodera 0x86. Pętla IMU ma cel 100 Hz, ale faktyczny odbiór zależy od pasma telemetry; częstotliwość RF nie jest częstotliwością nowych próbek. Błąd I2C nie powinien być interpretowany jako prawidłowe sztuczne 1 g.

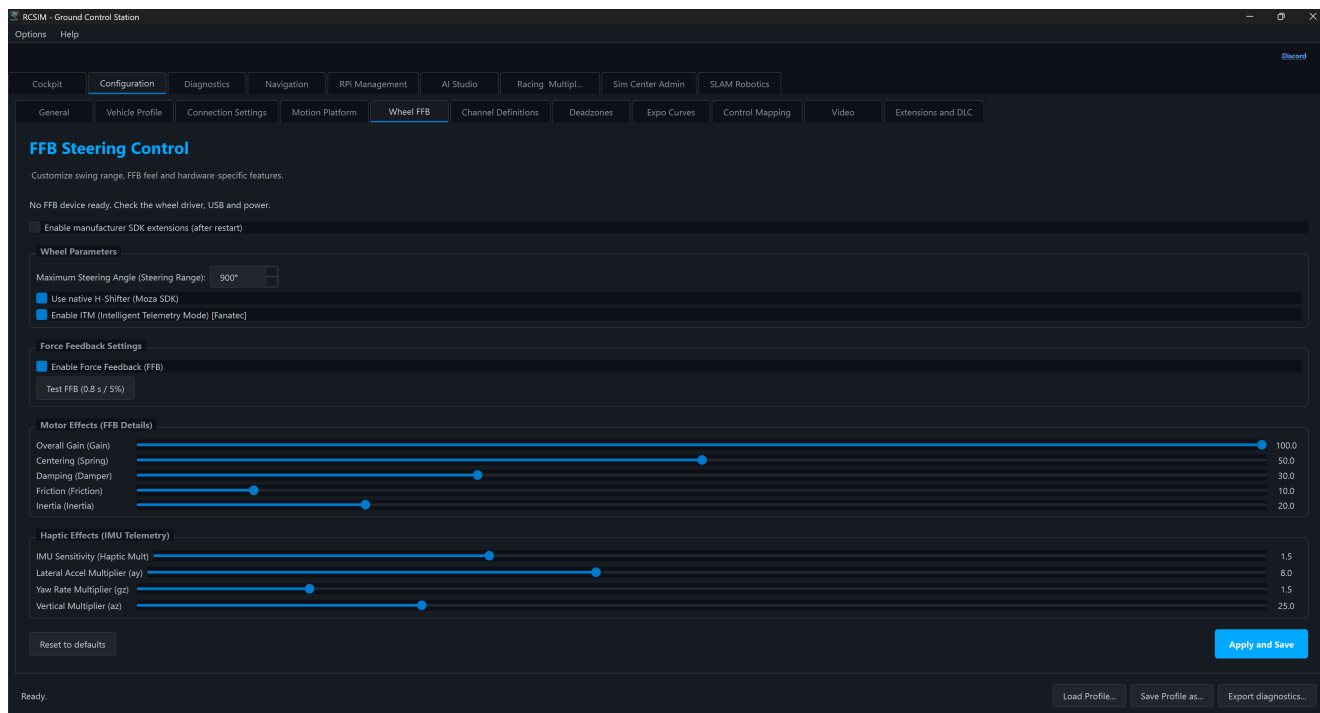
Odbiór GPS i IMU nie oznacza gotowej autonomii. ER5C V2 z tym modem pozostaje konfiguracją Tier 3. Zakres telemetry potwierdź na faktycznie odbieranych ramkach w GCS/MCS.

8. Diagnostyka

- **Jest COM, brak kanałów:** sprawdź protokół, baudrate, USB-VCP/Trainer i aktywną funkcję trenera.
- **Kanały działają, brak telemetry:** sprawdź FullDuplex zamiast simplex, tor telemetryczny, sensor i typ ramki.
- **GPS działa, ESC nie:** sprawdź wsad i konflikt Serial TX/PWM na CH2.
- **Brak IMU:** sprawdź funkcje CH4/CH5, adres i poziomy I2C; nie zmieniaj ich w ciemno na PWM.
- **RF działa, brak celu MAVLink:** sprawdź heartbeat właściwego kontrolera i SysID, nie tylko moduł radiowy.
- **Odłączenie PC daje nieoczekiwany gaz:** sprawdź cały tor trenera, miksera i odbiornika; sam stan ARM GCS nie wystarcza.

- **CRWF nie otwiera konfiguratora:** to ograniczenie medium; konfiguracja MSP nie jest w nim dostępna.
- **UDP działa w jedną stronę:** sprawdź osobno port docelowy i lokalny, ustawienia mostka i zaporę PC.

8.1. Kierownica, pedały i Force Feedback



Panel Wheel FFB: przykład ustawień; zrzut zgłasza brak gotowego urządzenia FFB

1. W **Channel Definitions** ustal kanał skrętu jako oś dwukierunkową i napęd zgodnie z elektroniką modelu.
2. Przy dwóch oddzielnych pedałach sterujących jednym ESC wybierz Split Axis; przypisz osobno Throttle i Brake w **Control Mapping**.
3. Ustaw kierunki przez Invert, sprawdź neutralny gaz oraz zakresy przy odłączonym napędzie.
4. Dopasuj Deadzones i Expo dopiero po prawidłowym przypisaniu osi; zapisz profil.
5. W **Wheel FFB** ogranicz siłę bazy sprzętowo, ustaw niski dodatni Gain i uruchom **Test FFB (0.8 s / 5%)** bez potrzeby telemetry.
6. Sprawdź źródło i świeżość IMU w Diagnostics, dopiero potem dodawaj osobno mnożniki ay, gz, az i Haptic Mult.

Pełna ramka IMU 0x13 daje inne dane niż sama orientacja 0x1E. Przy samej orientacji nie oczekuj rzeczywistych pomiarów nierówności z akcelerometru. Zanik dynamicznych sił przy utracie świeżej telemetry jest oczekiwany; nie kompensuj go podnoszeniem Gain. Pełny opis wszystkich suwaków i przycisków znajduje się w podręczniku GCS.

8.2. Próba całego zestawu przed jazdą

1. Sprawdź mocowanie, anteny, napięcia, mapowanie i zapisane wersje firmware, z odłączonym napędem.
2. Potwierdź ręczne kanały w GCS, radiu i odbiorniku oraz poprawny bieg neutralny.
3. Sprawdź ARM/DISARM i rolę CH5, potem fizyczny przełącznik przejęcia sterowania aparaturą.
4. Przerwij osobno USB/sieć PC, zasilanie hosta i RF; obserwuj rzeczywiste wyjście ESC.
5. Przywróć połączenie bez dotykania gazu i potwierdź brak niezamierzonego uruchomienia napędu.
6. Sprawdź GPS, IMU, baterię i statystyki łącza; brak sensora nie jest pomiarem zero.

7. Dopiero po przejściu prób przywróć napęd i wykonaj kontrolowany krótki przejazd z możliwością zatrzymania.

Nie ma jednej wartości failsafe właściwej dla wszystkich ESC, modeli i autopilotów. Konfiguracja ELRS/EdgeTX oraz lokalny stan GCS są różnymi warstwami; każdą zweryfikuj na swoim torze.

W logu diagnostycznym zapisz wersje firmware, protokół, port, ustawienia odcinka i moment zaniku danych. Usuń identyfikatory/loginy i inne dane, których nie chcesz udostępniać.

9. Granice potwierdzenia

Potwierdzono obecność implementacji strategii i narzędzi testowych oraz treść projektowych patchy. Nie potwierdzono zasięgu RF, czasu reakcji całego toru, poprawności wszystkich binariów, zgodności każdej rewizji sprzętu ani działania failsafe konkretnego ESC.

Przed jazdą muszą przejść testy: neutralny gaz, kierunki kanałów, przejęcie kontroli, utrata PC, utrata RF i powrót łączą bez niespodziewanego uruchomienia napędu.

10. Źródła researchu

Ścieżki są względne do katalogu projektu; numery linii wskazują stan sprawdzany podczas opracowania.

- `RCSIM_PC/pc_app/core/comm_manager.py:55` — rejestr strategii komunikacyjnych.
- `RCSIM_PC/pc_app/core/comm/crsf_transceiver.py:44` — parametry i wybór transportu CRSF.
- `RCSIM_PC/pc_app/core/comm/rf_link_strategy.py:43` — częstotliwość 100 Hz używana przez strategię GCS.
- `RCSIM_PC/pc_app/gui/tabs/connection_sections.py:196` i `connection_behavior.py:125` — pola oraz medium CRSF/MAVLink.
- `RCSIM_PC/pc_app/core/comm/transport/backpack_transport.py:25` — sesja CRWF i brak konfiguracji MSP.
- `RCSIM_PC/pc_app/core/comm/mavlink_rf_strategy.py:839` i `:917` — ARM/DISARM oraz kolejka RC override.
- `docs/manual/Zdjęcia/Nomad/Info.md` i `image.png` — przykład mapowania RX3/TX1, nie uniwersalny schemat elektryczny.
- `RCSIM_PC/pc_app/core/comm/crsf_transceiver.py:125` — uzbrojenie i rozbrojenie transceivera.
- `RCSIM_PC/pc_app/core/comm/mavlink_rf_strategy.py:20` — MAVLink 2.0 i identyfikatory modułów radiowych.
- `RCSIM_PC/pc_app/core/comm/mavlink_identity.py:1` — rozpoznawanie endpointów MAVLink.
- `RCSIM_PC/pc_app/ESP32Arduino/RCSIM_Arduino_ESP32/EdgeTX CRSF VCP mod/README_PL.md:1` — zakres moda, modele i deklarowany status testów.
- `RCSIM_PC/pc_app/ESP32Arduino/RCSIM_Arduino_ESP32/EdgeTX CRSF VCP mod/crsf_vcp_test.py:2` — wymagania konfiguracji aparatury.
- `RCSIM_PC/pc_app/ESP32Arduino/RCSIM_Arduino_ESP32/EdgeTX CRSF VCP mod/crsf_vcp_duplex_test.py:182` — argumenty, port 115200 bps i aktywne nadawanie testowe.
- `RCSIM_PC/pc_app/ESP32Arduino/RCSIM_Arduino_ESP32/ExpressLRS ER Series Tier 3 mod/README_PL_RadioMaster_ER5Cv2_GPS.md:13` — wariant GPS/IMU i mapowanie złączy.

- `RCSIM_PC/pc_app/ESP32Arduino/RCSIM_Arduino_ESP32/ExpressLRS_ER_Series_Tier_3/mod/elrs_v4.1_er5cv2_gps_imu.patch:7367` — zmiany RX i zwolnienie pinu dla PWM.