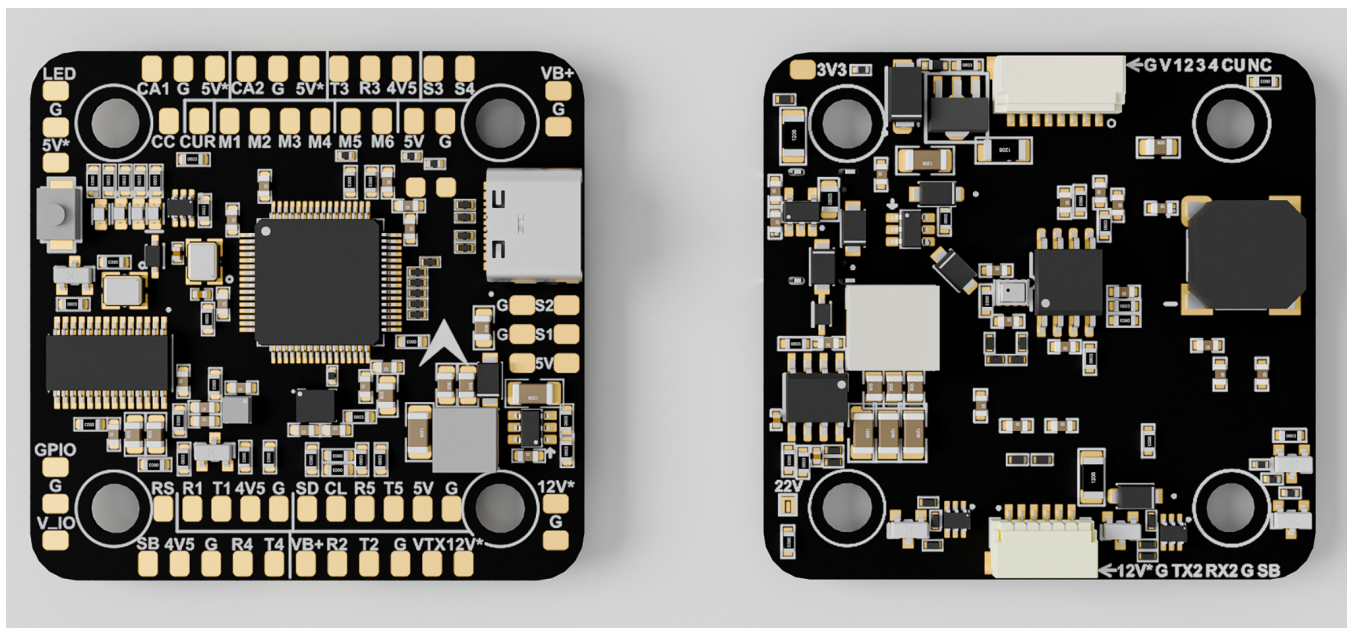


Інструкція з використання польотного контролера версії Skypulse F405 V3

1. Технічні характеристики	2
2. Підключення 3-х приймачів	4
3. Підключення 2-х VTX / Перемикання каналів	7
4. Одночасна робота цифрового та аналогового OSD	9
5. Вимикання живлення камер та VTX	10
6. 8 PINIO режимів	11
7. Покращені режими failsafe	12
8. Перемикання між режимами failsafe в польоті	14
9. Режим сну (swarm) + Beacon off	14
10. Відображення стану PINIO на OSD	16
11. Запобіжник та ініціація підриву	17
12. Відображення статусу AUX	19
13. Інвертування + налаштування 5 сервоприводів	20
14. Автоматичне перемикання потужності VTX	21
15. Маскування OSD через Camera frame	22
16. Покращений ALTHOLD	23
17. Телеметрія на DSHOT600 та 8KГц PID Loop	24
18. Підтримка гекса- та октакоптерів	25
19. Різні корисні налаштування	27
20. Логування температури регулятора	28
21. Налаштування серв та relay Ardupilot	29
22. Переключення каналів VTX в Ardupilot	30
23. Запис логів Ardupilot для F405	32
24. Калібрування нуля струму	33
25. Файли прошивок	34
26. Питання-відповіді (FAQ)	35
27. Changelog	36
28. Немає потрібної функції?	37

1. Технічні характеристики



Напруга і струм

від 4S до 8S, 80A (на канал)

- для 8S бажано використовувати загальну ємність конденсатора не менше 3000uF

Кількість виходів

- 5 UARTів
- 11 серво виходів
- 1 додатковий GPIO/PINIO (USER3 режим у BF)
- інші серво виходи також можуть бути PINIO*
- Camera Control

Датчики

- Барометр DPS310
- Компас LIS3MDL (опціонально)
- Гіроскоп ICM42688P або LSM6DSV16X

Додаткові можливості

- Інтегрований буюер
- Можливість підключити дві камери (camera switch)
- Підтримка [DSHOT600 + 8кГц Pid loop](#)

Флеш пам'ять

16 Мб

BEC 5V

3A

BEC 12V/22V

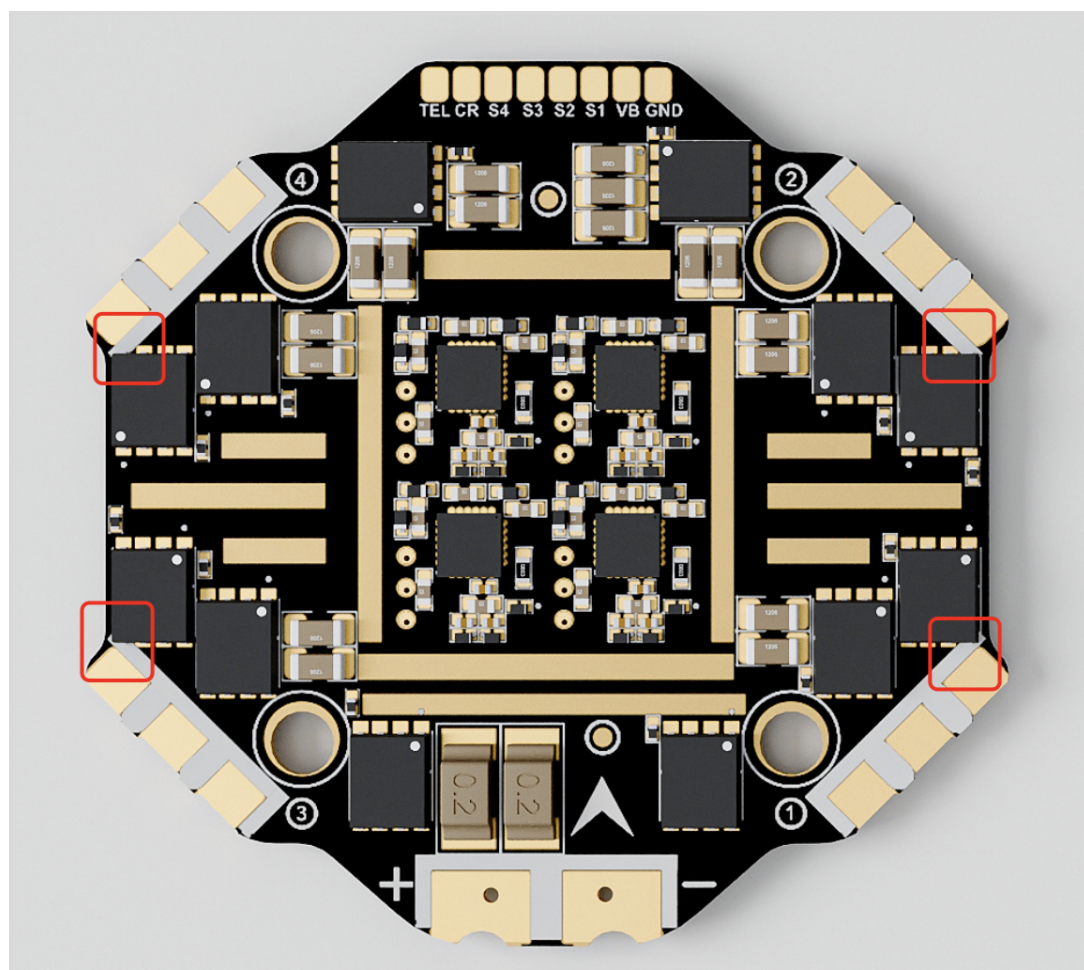
2A

(перемичкою можна збільшити напругу до 22V, 44Вт)

SBUS

Інвертований сигнал під'єднаний до R4 (4-й UART)

Увага! Ви можете з'єднувати ці ділянки припоєм (пад і мосфет можуть бути закорочені, це покращує протікання струму):



2. Підключення 3-х приймачів

Польотний контролер підтримує можливість роботи з **трьома** будь-якими приймачами (ELRS, Crossfire, SBUS і т.п.) одночасно.

Приймачі можуть працювати паралельно і виводити на OSD інформацію про RSSI і LQ одночасно:

```
set multi_rx_show_all_signals = ON
```

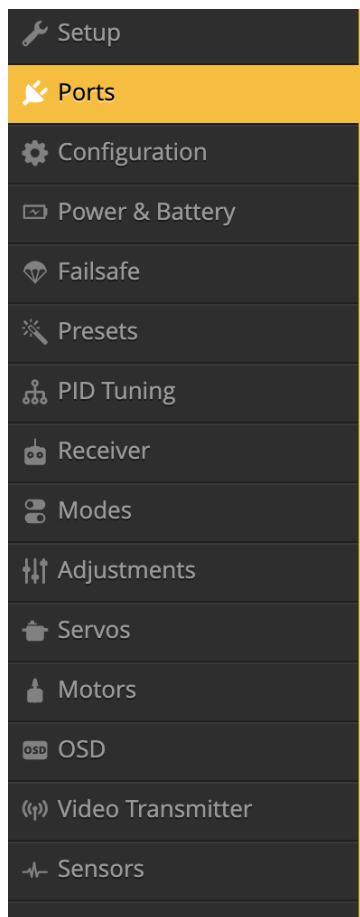
Приймачі перемикаються в залежності від значення LQ та RSSI, змінити рівні перемикачів можна командами (зміни після 05.04.2026):

```
set multi_rx_rssi_switch_threshold = -125
```

```
set multi_rx_link_quality_switch_threshold = 55
```

Якщо ви паяєте медіаконвертор + ELRS, запакуйте медіаконвертор на менший юарт (наприклад UART2, а ELRS приймач на UART4), тоді медіаконвертор буде головний (працюватиме першим).

2.1. Вам потрібно просто припаяти приймачі до вільних UARTів, а потім увімкнути «Serial Rx»:



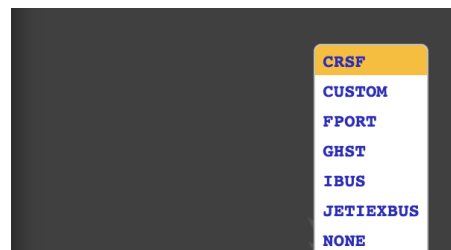
Ports

Note: not all combinations are valid. When the flight controller firmware detects this the se
Note: Do **NOT** disable MSP on the first serial port unless you know what you are doing. You

WARNING: The VTX table has not been set up correctly and without it VTX control will not b

Identifier	Configuration/MSP	Serial Rx	Telemetry Output
USB VCP	☒ 115200 ▾	☐	Disabled ▾ AUTO
UART1	☐ 115200 ▾	☒	Disabled ▾ AUTO
UART2	☐ 115200 ▾	☐	Disabled ▾ AUTO
UART3	☐ 115200 ▾	☐	Disabled ▾ AUTO
UART4	☐ 115200 ▾	☒	Disabled ▾ AUTO
UART5	☐ 115200 ▾	☒	Disabled ▾ AUTO

2.2. Налаштування в меню Receiver призначено для першого увімкненого порту (у наведеному вище прикладі це UART1). Телеметрія також надсилатиметься з цього приймача (так що



ви можете безпечно передавати телеметрію та мати 2 резервних приймача).

2.3. Якщо вам потрібно змінити протокол для додаткових приймачів Rx, ви можете зробити це за допомогою CLI (встановивши: `serialrx_provider2`, `serialrx_provider3`).

2.4. Щоб побачити активні приймачі під час польоту, активуйте параметр `set osd_show_rx_ports = ON`. В такому разі на OSD біля Link Quality (Якість зв'язку) з'явиться додаткова інформація:

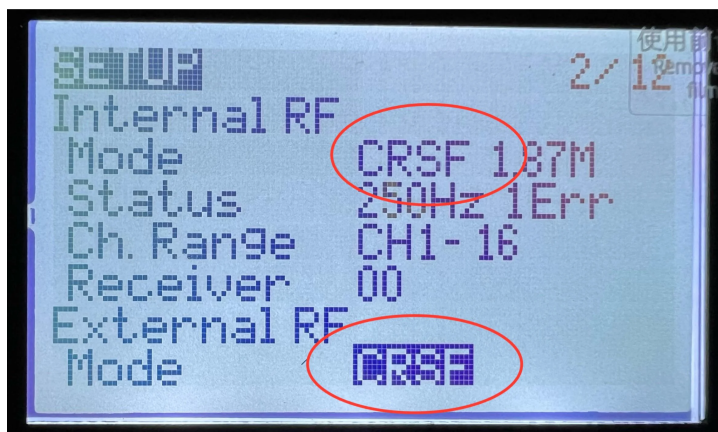
- список працюючих (не зареблених) приймачів позначається цифрами 1, 2, 3:
 - 1 — це перший увімкнений UART у списку портів (у нашому прикладі в пункті 2.1 це UART1)
 - 2 — це наступний порт (у прикладі це UART4) і так далі
- «+» буде біля номера активного приймача, дані LQ на OSD – це цей приймач.
- якщо ви хочете бачити LQ інших приймачів під час польоту, ви можете налаштувати режим «RX SWITCH», щоб змінити активний приймач.

2.5. Активний приймач змінюється автоматично, якщо якість сигналу стає критично низькою.

2.6. Увага! Приймачі можуть працювати паралельно, і виводити інформацію про якість сигналу та RSSI для всіх підключених приймачів, для цього потрібно активувати параметр: `set multi_rx_show_all_signals = on`

Сторона передавача:

Простий тест — взяти Radiomaster із зовнішнім і внутрішнім модулями. Ви можете активувати їх обидва та передати обидва сигнали на приймачі. У цьому випадку вам не потрібно нічого робити з наземною станцією. Для цього відкрийте сторінку налаштування на Radiomaster і переконайтеся, що внутрішній та зовнішній модулі активовані (вибрано CRSF). Такий підхід може стати в нагоді, якщо ви хочете використовувати його на полігоні, або перевірити РЕБ. Ви можете легко підійти до РЕБ і подивитися, на якій відстані ви втрачаєте сигнал (пропав активний приймач у списку LQ, див. п. 2.4), не втрачаючи контролю над дроном (якщо РЕБ не глушить частоту 2,4G).



3. Підключення 2-х VTX / Перемикання каналів

Увага! Знято обмеження на переключення каналів після арму, а також зняти обмеження на частоти.

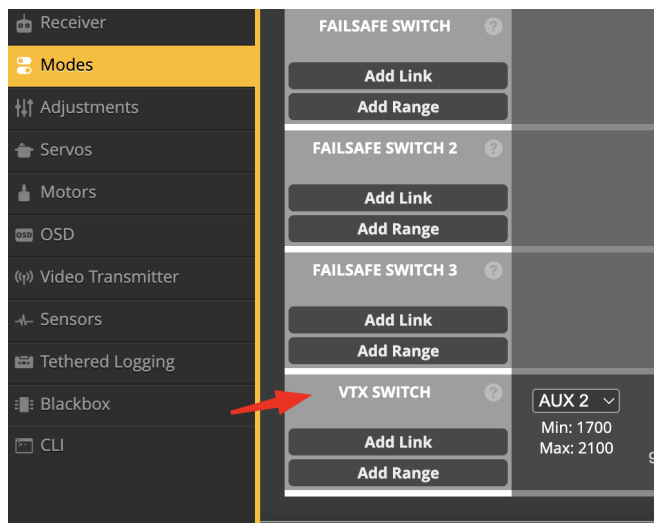
Польотний контролер підтримує одночасне використання двох VTX таблиць. Для активації вам достатньо в меню Порти (Ports) вибрати режим VTX (Tramp або Smartaudio) для двох UART інтерфейсів:

Identifier	Configuration/MSP	Serial Rx	Telemetry Output		Sensor Input		Peripherals	
USB VCP	☒ 115200	☐	Disabled	AUTO	Disabled	AUTO	Disabled	AUTO
UART1	☐ 115200	☐	Disabled	AUTO	Disabled	AUTO	VTX (TBS Smz	AUTO
UART2	☐ 115200	☐	Disabled	AUTO	Disabled	AUTO	VTX (IRC Tramp	AUTO
UART3	☐ 115200	☐	Disabled	AUTO	Disabled	AUTO	Disabled	AUTO
UART4	☐ 115200	☒	Disabled	AUTO	Disabled	AUTO	Disabled	AUTO
UART5	☐ 115200	☐	Disabled	AUTO	Disabled	AUTO	Disabled	AUTO

Робота відео передавачів може бути як паралельною, так і “по-черзі” (у такому разі неактивний VTX переходить у pit mode).

За замовчуванням відеопередачі будуть працювати по-черзі, щоб увімкнути паралельний режим виконайте у CLI наступну команду: `set vtx_switch_parallel = ON`

Щоб перемикатися між відеопередавачами (в режимі “по-черзі”), VTX SWITCH:



- Коли VTX SWITCH не активний – це дія 1-го передавача в списку портів, коли активний – 2-го.
- Переключаючи тумблер на пульті, ви зможете перемикатися між передавачами та встановлювати різні VTX таблиці для кожного з них. При переключенні з одного VTX на інший потрібно перезайти на сторінку "Video Transmitter".
- Ви можете встановити будь-які частоти, а також перемикати канали під час польоту.

- Відео вихід може бути спільний (просто паралельно підключений до двох VTX без додаткових елементів на платі), а управління Tramp/SmartAudio у кожного передавача має бути своє (по окремих UARTax).

4. Одночасна робота цифрового та аналогового OSD

В клі потрібно прописати примусово, що аналогове OSD – основне:

```
set osd_displayport_device = MAX7456
```

І в Портах вмикнути MSP + Display port для UARTа де підключено цифрову камеру.

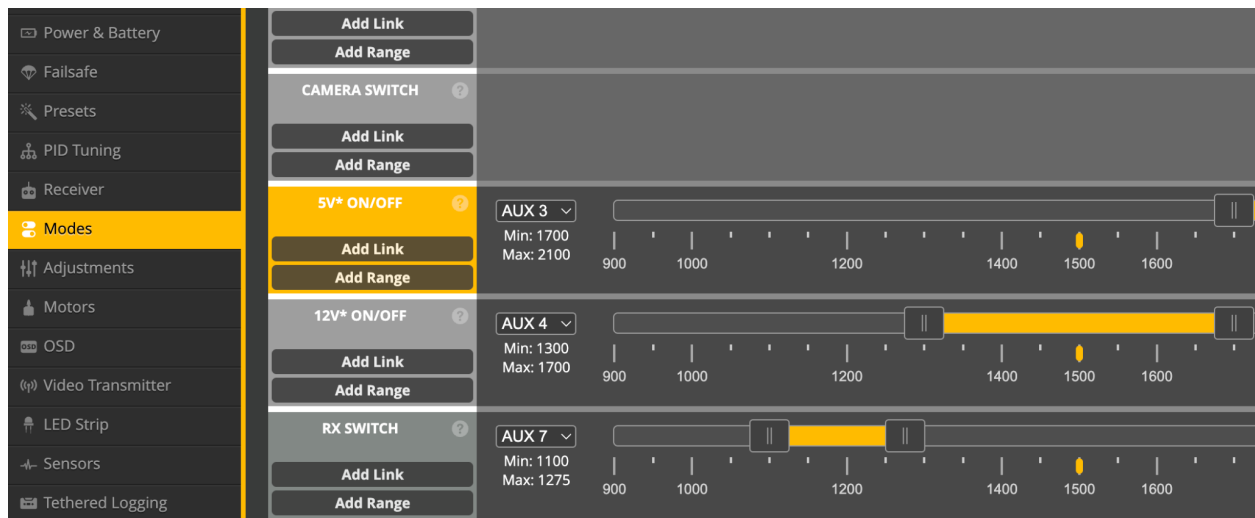
5. Вимикання живлення камер та VTX

Ви можете контролювати живлення на падах 5V* та 12V* без додаткових реле та мосфетів.

Щоб режими працювали, у pinio_box мають бути прописані такі id: 43, 100, 101:

set pinio_box = 255,255,255,43,100,101,255,255 (замість 255 може бути будь-які інші налаштування, див. [розділ налаштування PINIO](#)):

- CAMERA SWITCH (43) — перемикання камер.
- CAMERA 5V POWER (100) — увімкнення/вимкнення живлення позначених як 5V*.
- VTX 12V POWER (101) — увімкнення/вимкнення живлення позначених як 12V*.



6. 8 PINIO режимів

Польотний контролер підтримує налаштування до 8 режимів PINIO (USER1-USER8).

Три режими встановлені за замовчуванням:

```
set pinio_box = 255,255,42,43,100,101,255,255
```

- Керування V_IO (42) — вмикання 5V окремим піном (USER3)
- CAMERA SWITCH (43) — перемикання камер (USER4)
- CAMERA 5V POWER (100) — вимкнення живлення позначених як 5V* (USER5)
- VTX 12V POWER (101) — вимкнення живлення позначених як 12V* (USER6)

Ви можете додати 5 інших PINIO:

- USER1 – 40
- USER2 – 41
- USER7 – 102
- USER8 – 103

Наприклад, щоб активувати замість S1 додатковий PINIO, потрібно виконати наступні команди:

Для режиму USER1:

```
set pinio_box = 40,255,42,43,100,101,255,255
```

```
resource SERVO 1 none
```

```
resource PINIO 1 A15
```

Для режиму USER7:

```
set pinio_box = 255,255,42,43,100,101,102,255
```

```
resource SERVO 1 none
```

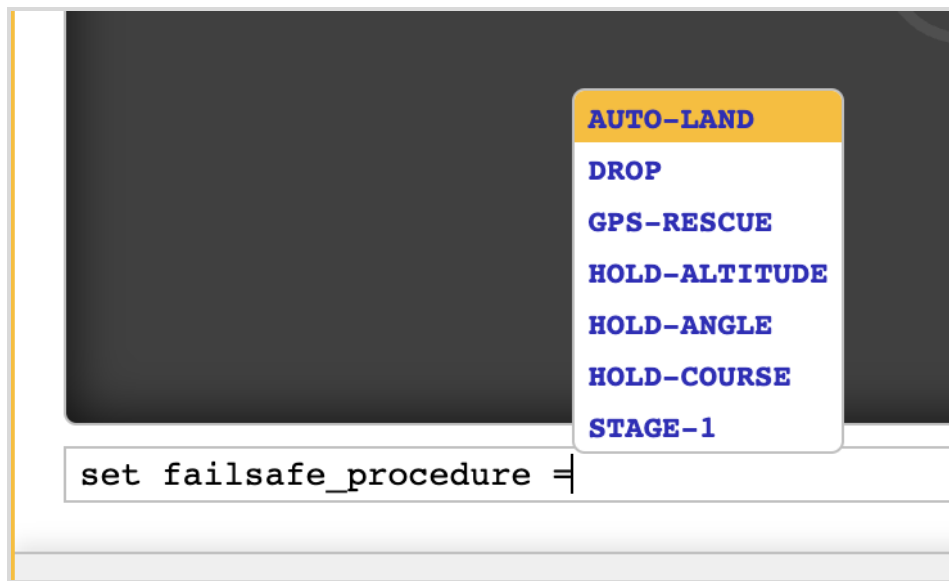
```
resource PINIO 7 A15
```

7. Покращені режими failsafe

Важливо: Перемикання режимів для Stage 2 в конфігураторі вимкнене, ви можете змінити режимі тільки через cli: `set failsafe_procedure =`

Подовжений failsafe працює за принципом прошивки Milbeta, потрібно вказати параметр `mb_delay_factor` який буде множитись на значення з конфігуратора (це час коли дрон отримує команди вказані у Stage1)

Ви також можете налаштувати додаткові режими (і перемикатися між ними, [див. розд. 8](#)):



- **HOLD_ANGLE**

Цей режим корисний в момент доведення дрона до цілі. Дрон тримає кут нахилу (Pitch) і швидкість (Throttle), які були встановлені на момент втрати сигналу.

- **HOLD_COURSE**

Цей режим корисний для проходження зони дії перешкод. Дрон тримає висоту і курс, які були на момент втрати сигналу. Ви можете налаштувати параметри:

- Щоб задати висоту відносно точки старту, на яку дрон перейде і буде тримати до відновлення сигналу (значення в метрах, значення "0" – висота на момент втрати сигналу):

```
set failsafe_hold_altitude = 100
```

- Щоб задати кут нахилу (Pitch), який буде використовуватись під час дії режиму (значення в градусах, 0 – кут на момент втрати сигналу):

```
set failsafe_course_pitch = 10
```

- **HOLD_ALTITUDE**

Дрон перейде в режим ALT HOLD. Може тримати поточну висоту на момент втрати сигналу або перейти на задану висоту. Налаштування:

- Щоб встановити висоту (значення в метрах, 0 – висота на момент втрати сигналу):
`set failsafe_hold_altitude = 100`
 - Або щоб виставити висоту на яку має підняти дрон (наприклад якщо підняти на висоту 20 метрів від поточної дрона):
`set failsafe_climb_altitude = 20`
- Щоб активувати режим обертання по осі Yaw (для пошуку сигналу):
`set failsafe_hold_altitude_rotate = ON`
- Дрон може перемикнутися на режим посадки (Landing) автоматично через вказаний таймаут:
`set failsafe_hold_time = 60`
- Ви можете вказати швидкість зниження (також працює для режиму AUTO_LAND):
`set failsafe_descent_rate = 50` (50 сантиметрів за секунду)
- Є стандартні налаштування, важливі для цього режиму:
 - `failsafe_landing_time` – через скільки секунд зробити дізарм після початку лендінгу (поставити побільше, щоб дрон встиг сісти), збільшити час можна через `mb_landing_factor`
 - `landing_disarm_threshold` – значення для акселерометра по G, щоб бетафлайт визначив удар об землю і задізармив двигуни сам, при повільному спусканні можливо потрібно буде поставити менше значення

● STAGE-1

Активує налаштування, які виставлені в Конфігураторі в розділі Stage 1.

● Повернення без GPS

Режим в розробці, стежте за оновленнями.

Важливо: Протестуйте потрібні вам режими, щоб розуміти які можуть бути особливості. Якщо у вас виникли питання або потрібні додаткові режими, звертайтеся до розробника.

Важливо: Щоб вибрані режими працювали коректно, вам потрібно встановити значення Failsafe Switch в налаштуваннях в Stage 2:

valid pulses.

Failsafe Switch

Stage 2 Failsafe Switch Action



Stage 2 - Settings

1,5 Period of time in Stage 1 failsafe after signal loss [seconds]



10,0 Failsafe Throttle Low Delay [seconds]



Stage 2 - Failsafe Procedure

☐ Drop



8. Перемикання між режимами failsafe в польоті

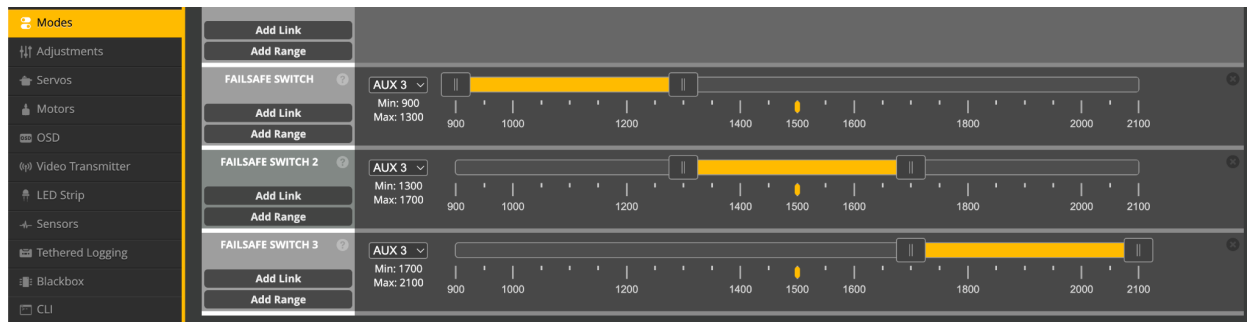
- Щоб подолати перешкоди, активуйте режим **HOLD_COURSE** – дрон підтримуватиме напрямок руху.
- Щоб дрон долетів до цілі, використовуйте режим **HOLD_ANGLE** – зберігається кут нахилу та швидкість.
- Якщо ви втратили відеозв'язок, підніміть дрон на безпечну висоту за допомогою режиму **HOLD_ALTITUDE**.

Для цього:

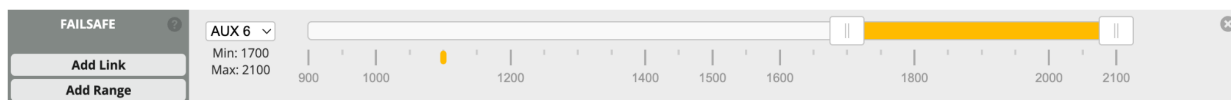
1. Встановіть параметри для кожного режиму, щоб мати можливість перемикатися між ними:



2. В розділі "Режими" призначте потрібний канал AUX для активації функцій:



3. Ви можете активувати будь-який встановлений режим вручну з пульта керування. Для цього налаштуйте окремий канал AUX для режиму FAILSAFE:



Щоб увімкнути відображення активного режиму failsafe на екрані OSD, використовуйте команду:

```
set osd_show_failsafe_type = ON
```

9. Режим сну (swarm) + Beacon off

Увага! Beacon та інші особливості роботи дрона в тривалому очікуванні (30+ хв) виправлені:

але якщо вам потрібно цілеспрямовано вимкнути всі BEACON на двигуни, можете використати цей параметр:

```
set mb_beacon_off = ON
```

Цей режим дозволяє завчасно розставити увімкнені дрони в режимі сну. Споживання енергії при цьому становить не більше 100 мА, тому що автоматично без додаткових засобів вимикаються всі основні споживачі енергії – VTX, камера, OSD чіп та інша периферія яка запаряна на 5V* та 12V*, які позначені зірочками. Це дозволяє дронам довго (навіть кілька днів) залишатися в готовності до активації.

Щоб активувати режим сну, потрібно прописати ID дрона та сумарну кількість дронів у рої:

```
set swarm_drone_id = 1  
set swarm_total_drones = 10
```

Після налаштування стане доступним режим **DRONE SWITCH**, який перемикає дрони. Рекомендується налаштувати цей режим **на кнопку**:

- **Затискання кнопки на 3 секунди** — обнуляє лічильник на всіх дронах, і вони всі переходять у режим сну
- **Один клік на кнопку** — активує 1-й дрон, він "прокидається" і починає транслювати відео
- **Ще один клік** — поточний дрон засне, а наступний прокинеться

На OSD біля індикатора режиму польоту буде відображатися номер поточного активного дрона.

Увага! Поки дрон "спить", він не буде сприймати команди з пульта, на всі канали будуть активовані значення з Failsafe Stage 1.

Щоб деактивувати режим, потрібно виконати:

```
set swarm_drone_id = 0
```

Для коректної роботи необхідно:

- На всіх приймачах має бути прописана однакова бінд фраза.
- На регуляторах має бути вимкнений режим повідомлення про неактивність (інакше через 10 хв почнуть пищати мотори).
- Налаштування VTX мають бути однаковими на всіх дронах.
- Телеметрія має бути вимкнена (інакше на передавач буде приходити телеметрія одночасно з усіх дронів).

10. Відображення стану PINIO на OSD

PINIO можуть працювати не лише як виходи, але і як входи. Це дозволяє вам відображати на екрані стан зовнішніх пристроїв. Наприклад, якщо у вас є плата ініціації, яка подає сигнал 3.3V (логічну 1) на один зі своїх контактів, коли вона активована, ви можете відобразити цю інформацію прямо на екрані OSD.

З'єднайте контакт зовнішнього пристрою з відповідним PINIO на польотному контролері, наприклад S1:

```
resource SERVO 1 none  
resource PINIO 1 B0
```

Тепер PINIO 1 буде в нас на місці S1, ставимо 2 на відповідному місці:

```
set pinio_config = 2,1,1,1,1,1,1,1  
set pinio_box = 40,255,255,43,100,101,255,255
```

І потім:

```
set osd_show_input_pinios = ON  
set box_user_1_name = UVAGA
```

Потрібні номери пінів, можете перевірити через команду `resource`, наприклад:

B0	S1
B1	S2
A6	S3
A7	S4

11. Запобіжник та ініціація підриву

Увага! Плата ініціації повинна мати свої запобіжники (механічні, по часу, тощо). Ця функція може використовуватись тільки як додаткова функція захисту та інформування по OSD.

Увага! Зняти запобіжник (FUSE) можна коли дрон не заармлено, але дрон має бути готовий до арму (підключена батарея, відключено USB, тощо). Це зроблено для безпеки. Тобто, якщо ви хочете зняти запобіжник з підключеним USB, потрібно поставити галочку на вкладці Motor tests (дозволити арм з підключеним USB).

Для активації функції запобіжника/підриву на польотному контролері необхідно виконати в CLI команду:

```
set fuse_charge_on = on
```

Після активації, мають з'явитися режими **CHARGE** (підрив) та **FUSE** (запобіжник).

Модуль працює в двох режимах PINIO та PWM. Для PINIO режиму використовується два пади, для PWM – один.

Наприклад для PINIO режиму якщо взяти S1, S2 (можна вибрати будь-які інші піни):

```
resource SERVO 1 none  
resource SERVO 2 none
```

```
resource PINIO 1 B00  
resource PINIO 2 B01
```

Тепер:

- **S1** працюватиме як **PINIO 1** – функція **CHARGE** (підрив)
- **S2** працюватиме як **PINIO 2** – функція **FUSE** (запобіжник)

Якщо вам потрібно щоб на плату ініціації подавався PWM сигнал, ви можете активувати цей режим командою:

```
set fuse_by_servo = ON
```

У цьому режимі використовується лише один пад, наприклад якщо S1 (можна вибрати будь-який інший пін):

```
resource SERVO 1 none  
resource PINIO 1 B00
```

Увага! Ресурс PINIO прописуєте незалежно у вас ШИМ чи просто GPIO, система сама перемикне налаштування пада.

- На **PINIO 1** будуть слатися такі сигнали: PWM 1000 – запобіжник не знятий, PWM 1500 – запобіжник знятий, PWM 2000 – підрив.
- Змінити сигнали можна через параметри: **fuse_servo_disabled**, **fuse_servo_fuse**, **fuse_servo_charge**, **fuse_servo_auto_charge** – має бути 1000, якщо плата на підрив при ударі

активується відразу після зняття запобіжника (FUSE), інше значення додасть режим “AUTO CHARGE” щоб пультом активувати підрив при ударі

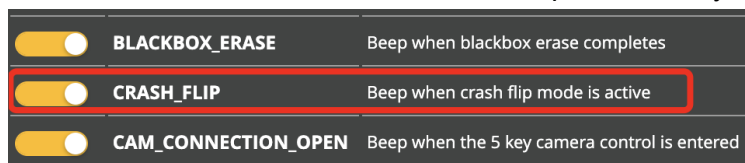
Якщо потрібно щоб плата слала 2 окремі PWM сигнали (окремо FUSE, окремо CHARGE), вам потрібно активувати параметр:

```
set fuse_servo_dual = ON
```

і визначити PINIO 2.

На екрані OSD будуть відображатися відповідні попередження та інформація.

Польотний контроллер буде подавати звукові сигнали, якщо активовано підрив з вимкненим запобіжником, для цього потрібно щоб був активований CRASH_FLIP:



Ви також можете активувати подачу звукових сигналів за X секунд до закінчення таймера для додаткової безпеки і привернення уваги (“0” вимикає подачу звукових сигналів по таймеру):

```
set fuse_beep_before = 20
```

Інші налаштування:

- `fuse_timer` – змінити час таймера.
- `fuse_ready_message` – змінити повідомлення коли запобіжник активовано.
- `fuse_keep_timer` – якщо ON, таймер буде тікати незалежно від перемикача підриву; в іншому випадку, таймер буде скидатися поки не деактивувати перемикач підриву.
- `fuse_signal_on_aux` – послати сигнал на запобіжник одразу після активації aux. Увагу, використовуйте цей параметр тільки щоб синхронізувати плату ініціації якщо в неї є свій таймер.
- `fuse_wait_feedback` – не зніметься запобіжник, поки плата не отримає логічне “1” на PINIO, як налаштувати PINIO на вхід, див. [10. Відображення стану PINIO на OSD](#) (вмикати `osd_show_input_pinios` не обов’язково!).

Увага! Якщо для вашої плати ініціації потрібні додаткові функції, напишіть нам hello@skypulse.ua, і ми швидко адаптуємо прошивку під ваші вимоги.

12. Відображення статусу AUX

`set osd_aux_channel = 3` (номер аукса на який у вас виведено управління платою ініціації)

`set osd_aux_text1 = SAFE`

`set osd_aux_text2 = ACTIVE`

`set osd_aux_text3 = ARMED`

Для наочності, ви можете додати спецсимвол перед словом, наприклад xCFSAFE, де xCF – код символу (потрібно шрифти Milbeta), або один зі стандартних шрифтів:

<https://betaflight.com/docs/development/OSD-Glyphs> (наприклад x3ESAFE).

І активувати Aux value

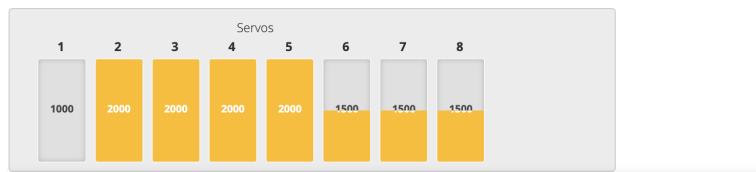
☐	☐	☐	Artificial horizon sidebars
☐	☐	☐	Artificial horizon
☒	☐	☐	Aux value 
☐	☐	☐	Battery average cell voltage
☐	☐	☐	Battery current draw

13. Інвертування + налаштування 5 сервоприводів

Ви можете керувати налаштуваннями 5-ти серв у конфігураторі замість 2-х як у стандартній прошивці:

Name	MIN	MID	MAX	CH1	CH2	CH3	CH4	A1	A2	A3	A4	A5
Servo 1	2000	1500	1000	☐	☐	☐	☐	☐	☐	☒	☐	☐
Servo 2	1000	1500	2000	☐	☐	☐	☐	☐	☐	☒	☐	☐
Servo 3	1000	1500	2000	☐	☐	☐	☐	☐	☐	☒	☐	☐
Servo 4	1000	1500	2000	☐	☐	☐	☐	☐	☐	☒	☐	☐
Servo 5	1000	1500	2000	☐	☐	☐	☐	☐	☐	☒	☐	☐
Servo 6	1000	1500	2000	☐	☐	☐	☐	☐	☐	☐	☐	☐
Servo 7	1000	1500	2000	☐	☐	☐	☐	☐	☐	☐	☐	☐
Servo 8	1000	1500	2000	☐	☐	☐	☐	☐	☐	☐	☐	☐

☐ Enable Live mode



Функція **SERVO_TILT** має бути увімкнена, а **CHANNEL_FORWARDING** вимкнена:



Для інвертування напрямку роботи сервоприводів не потрібно змінювати налаштування прошивки або апаратури. Ви можете просто поміняти місцями значення **min** та **max** в налаштуваннях серв у Betaflight Configurator, що змінить напрямок руху сервоприводу.

Change Direction in TX To Match												
Name	MIN	MID	MAX	CH1	CH2	CH3	CH4	A1	A2	A3	A4	A5
Servo 1	2000	1500	1000	☐	☐	☐	☐	☐	☐	☒	☐	☐
Servo 2	1000	1500	2000	☐	☐	☐	☐	☐	☐	☒	☐	☐

На польотному контролері вже є виходи S1-S4, S5 можна отримати замість LED виходу, для цього в клі потрібно виконати такі команди:

```
resource LED_STRIP 1 none
```

```
resource SERVO 5 B09
```

14. Автоматичне перемикання потужності VTX

Ви можете налаштувати автоматичне перемикання рівнів потужності відеопередавача (VTX) після певного часу з моменту активації дрона (арму). Це допоможе маскувати позицію з якої злітає дрон без потреби вручну регулювати потужність.

Для налаштування використовуйте команду:

```
set vtx_power_switch_timeouts = a,b,c
```

де:

- **a** — затримка (у секундах) перед перемиканням на 2-й рівень потужності після арму
- **b** — затримка до перемикання на 3-й рівень після другого
- **c** — затримка до перемикання на 4-й рівень після третього

Приклад налаштування:

```
set vtx_power_switch_timeouts = 10,20,30
```

В такому разі:

- Після арму VTX працюватиме на 1-му (найменшому) рівні потужності
- Через 10 секунд → автоматично перемикнеться на 2-й рівень потужності
- Через 20 секунд після цього (загалом 30 секунд після арму) → перемикнеться на 3-й рівень
- Через 30 секунд після цього (загалом 60 секунд після арму) → перемикнеться на 4-й рівень

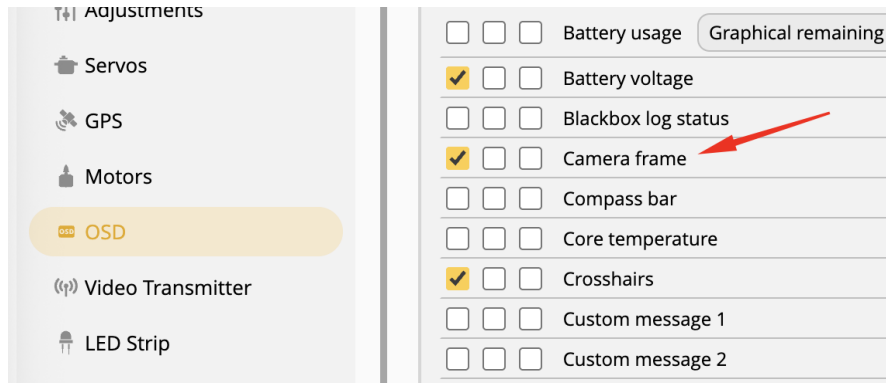
Спрощене налаштування:

```
set vtx_power_switch_timeouts = 10
```

Це налаштує перемикання потужності кожні 10 секунд для всіх рівнів.

15. Маскування OSD через Camera frame

1. Активувати Camera frame на OSD:



2. Встановити потрібні розміри і розташування фрейму:

```
set osd_camera_frame_height = 10
```

```
set osd_camera_frame_width = 10
```

3. Вам також потрібно активувати заповнення екрану прописавши код символу, який буде заповнювати екран, якщо ви використовуєте шрифти Milbeta, це номер 210:

```
set osd_frame_fill_char = 210
```

16. Покращений ALTHOLD

Режим працює автоматично, не потрібно робити додаткові налаштування (якщо `ap_hover_throttle = 0`).

- Фіксування висоти здійснюється по центру газу (як у ardupilot). Відповідно газ вище центру – піднятися вгору, нижче центру – опуститися вниз.
- При увімкненні режиму, якщо не рухати стік газу, то дрон буде залишатися на одному місці.
- Поки стік газу не довести до центру (при першому включенні), дрон не буде реагувати на газ і змінювати висоту.

Щоб вимкнути регулювання висоти стіками (дрон буде завжди тримати висоту де вмикнувся режим ALTHOLD): `alt_hold_deadband = 0`

Щоб вийти з режиму ALTHOLD потрібно поставити тротл приблизно в положення де дрон зависає (щоб різко не злетів угору чи впав униз), або поставити в те саме місце в якому вмикався режим ALTHOLD.

Щоб збільшити (або зменшити) швидкість зміни висоти змініть `alt_hold_climb_rate` (в дециметрах, тобто множиться на 10, отримуємо см, наприклад поставлене значення 30 = 300см)

Увага! Під час увімкнення AltHold режим Angle не активується автоматично. Якщо вам потрібен режим Angle, налаштуйте його в режимах, щоб вмикався разом з ALTHOLD.

17. Телеметрія на DSHOT600 та 8КГц PID Loop

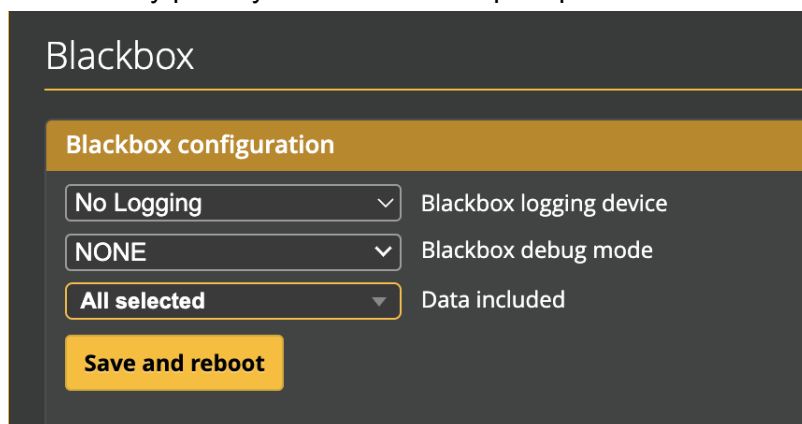
Польотний контролер оптимізовано для швидкісної роботи і ви можете активувати максимальний режим роботи DSHOT600 + 8 КГц (що не доступно для більшості контролерів на базі F405).

Режим DSHOT600 особливо рекомендований для швидкісних FPV-дронів, які потребують високої чутливості та швидкої реакції на команди керування, а також для рам, у яких неможливо повністю прибрати вібрації за допомогою стандартних фільтрів. Використання PID Loop із частотою 8 кГц у поєднанні з DSHOT600 забезпечує максимально точне й стабільне керування, зменшує затримки та покращує загальну якість польоту.

Для активації: спочатку поставити режим DSHOT600, потім виставити частоту Pid loop 8Khz.

Зверніть увагу: польотний контролер побудований на базі мікропроцесора F405, продуктивності якого може бути недостатньо для одночасної роботи режиму DSHOT600 разом з іншими ресурсомісткими функціями вашого дрона. У разі перевищення навантаження, Betaflight не дозволить заармити дрон та виведе повідомлення про помилку «Load».

Під час використання протоколу DSHOT600 рекомендується відключати dshot_edt = OFF а також функцію логування, це допоможе знизити навантаження на процесор та забезпечити стабільнішу роботу польотного контролера:



The screenshot shows the 'Blackbox' configuration menu in Betaflight. It has a dark grey background with a yellow header bar. The title 'Blackbox' is in white. Below the header, 'Blackbox configuration' is written in yellow. There are three dropdown menus: 'No Logging' (with a downward arrow), 'NONE' (with a downward arrow), and 'All selected' (with a downward arrow). To the right of these are labels: 'Blackbox logging device', 'Blackbox debug mode', and 'Data included'. At the bottom left is a yellow button with the text 'Save and reboot'.

Blackbox configuration	
No Logging	Blackbox logging device
NONE	Blackbox debug mode
All selected	Data included

Save and reboot

18. Підтримка гекса- та октакоптерів

Група таймера	Канал	Ardu GPIO	Пін	Призначення
TIM8	CH1	53	C06	Мотор M4
	CH2	50	C07	Мотор M1
	CH3	51	C08	Мотор M2
	CH4	52	C09	Мотор M3
TIM2	CH3	54	B10	Мотор M5
	CH4	55	B11	Мотор M6
TIM3	CH3	56	B00	Серво S1
	CH4	57	B01	Серво S2
	CH1	58	A06	Серво S3 / або TIM13
	CH2	59	A07	Серво S4 / або TIM14
TIM4	CH4	60	B09	LED Strip

Особливості конфігурації:

- **Режим гексакоптера** підтримується з коробки: достатньо активувати тип двигунів "HEX".
- **Режим октакоптера (ОСТО)** також підтримується, але для цього рекомендовано задіяти піни **S1-S2** як мотори (не S3-S4). В цьому випадку:
 - Сервоканали **S1-S2** необхідно перевизначити для моторів:
resource SERVO 1 none
resource SERVO 2 none
resource SERVO 3 none
resource SERVO 4 none
resource MOTOR 7 B00
resource MOTOR 8 B01
resource SERVO 1 A06
resource SERVO 2 A07

Виходи S3-S4 будуть визначені як Servo 1 та Servo 2.

Особливості роботи сервоприводів:

- Сервоканали **S1-S4** можуть бути розподілені аж на **три різні групи таймерів**, що дозволяє підключити 3 окремі групи пристроїв: S1-S2 - одна група, S3 може бути назначений на TIM13 (2-га група), S4 може бути назначений на TIM11 (3-тя група).

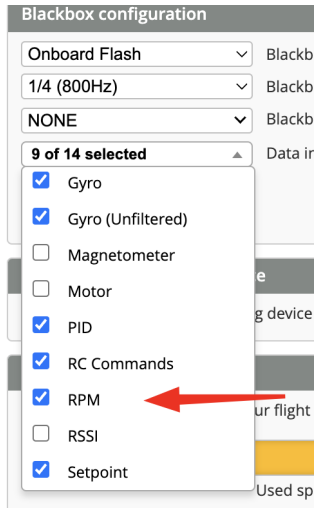
19. Різні корисні налаштування

1. Вимкнення програмних led: `set led_enabled = OFF`
Програмно можна вимкнути тільки два леди, інші два леди системні, горять якщо на дроні є живлення 5V та 3.3V (їх потрібно замазувати, або випаювати, можна замовити версію без ледів).
2. Встановлення точки Home:
`mb_gps_home_lat`
`mb_gps_home_lon`
`mb_gps_home_max_distance`
3. Збільшення каналів телеметрії до 16: `set mavlink_send_rc_channels = ON`
4. Збільшити час приземлення в режимі failsafe: `set mb_landing_factor`
5. Вимкнути всі звуки бузера: `beeper -ALL`

20. Логування температури регулятора

Для цього потрібно увімкнути режим EDT для DSHOT:
set dshot_edt = ON

В режимах blackbox, має бути увімкнено RPM:



Після цього, графіки з температурою ви можете переглядати на сайті
<http://skypulse.ua/pidpulse/>

21. Налаштування серв та relay Ardupilot

21.1) Мотори та серво:

- a. M1-M4 – це SERVO1 - SERVO4
На ці серво виходи налаштовані DMA і повинні використовуватись як мотори або серво виходи для управління керуючими поверхнями літака.
- b. M5-M6 – це SERVO5 - SERVO6
Ці серво виходи можуть використовуватись як звичайні серво, мотори для літака.
- c. S1-S4 – це SERVO7-SERVO10
Серво виходи для гімбала, скиду, парашуту.
- d. LED – це SERVO11
Додатковий серво вихід.

Опис таймерів див. в розділі [17. Підтримка гекса- та октакоптерів](#)

22.2) Relay налаштування:

- 1. Вмикнути потрібний relay:
RELAYx_FUNCTION = 1
RCn_OPTION = Relay_x On/Off (1-й relay x це опція 28, 2-й – 34, потім 35, 36, 66, 67)
- 2. Налаштувати відповідний пін:
 - Перемикання між камерами CAM1/CAM2:
RELAY1_PIN = 81
 - Вмикання/вимикання 5V*:
RELAY2_PIN = 82
 - Вмикання/вимикання 12V*:
RELAY3_PIN = 83
 - Пад GPIO:
RELAY4_PIN = 84
 - Пад CC (в ардупайлоті налаштований як GPIO):
RELAY5_PIN = 85
- Якщо потрібно переназначити серво вихід як relay:
SERVO_n_FUNCTION = -1
І тоді для RELAYx_PIN прописуєте 50 + (n-1), де n - номер серво виходу, наприклад:
SERVO1 - це PIN 50, SERVO2 відповідно 51 і т.д.

22. Переключення каналів VTX в Ardupilot

Канали

Щоб перемикання каналів працювало, потрібно проставити канал який за це відповідатиме через `VTX_FREQ_CH = X`. Далі виставити потрібні частоти в параметрах `VTX_FREQ1`, `VTX_FREQ2`, ... `VTX_FREQ8`. Якщо вам потрібно перемикатися між 3 каналами, прописуєте тільки 1, 2, 3 інші будуть ігноруватися.

Якщо для вашого VTX вказано FACTORY в налаштуваннях `vtxtable`:

```
vtxtable band 1 BOSCAM_A A FACTORY 611 613 615 617 619 621 623 625
```

тоді вам потрібно прописувати не частоти, а бенд і канал, наприклад: "15" – бенд 1, канал 5; 26 – бенд 2, канал 6 тощо.

Потужність

Вмикнути перемикання потужності потрібно або стандартним методом `RCx_OPTION = 94` (VTX Power) або через опцію `VTX_POWER_CH = X`.

`VTX_POWER_CH` додано, якщо вам потрібно наприклад на канал повісити додатковий функціонал. Нобто ви можете перемикати потужність і контролювати наприклад Relay через один і той самий AUX.

І за аналогією прописуєте `VTX_POWER1`, `VTX_POWER2`, ... `VTX_POWER6`.

Значення прописуються в тому форматі, як приймає VTX, або мВт: 25, 500, 1000, або в dBm: 14, 27, 30, або 1, 2, 3, 4 – ті значення, які зазначені у параметрі `powervalues`.

Можна прописати 2 чи 3 значення між якими буде перемикатися потужність (не обов'язково всі 4 використовувати).

Увага!

На деяких передавачах `VTX_POWER1` повинен бути виставлено в 0 (спробуйте, якщо не працює звичайний режим).

Увага!

Серіал порт повинен бути коретно налаштований:

```
SERIAL2_OPTIONS = 4 (HalfDuplex)
```

Для Tramp:

```
SERIAL2_BAUD = 9 (9600)
```

```
SERIAL2_PROTOCOL = 44
```

Для Smart Audio:

SERIAL2_BAUD = 4 (4800)

SERIAL2_PROTOCOL = 37

23. Запис логів Ardupilot для F405

Прошивка Skypulse для Ardupilot модифікована таким чином, щоб можна було записувати логи для підів (використовуйте skypulse.ua/pidpulse для аналізу) і шумів.

Потрібно встановити такі параметри:

LOG_BITMASK = 0 (або 4096 якщо записувати одночасно з підами)

INS_LOG_BAT_MASK = 1

INS_LOG_BAT_OPT = 4

Якщо вам потрібні логи з розділу PIDR і PIDP використовуйте:

LOG_BITMASK = 4096

Увага!

Обов'язково вимкніть:

INS_RAW_LOG_OPT = 0

LOG_REPLAY = 0

24. Калібрування нуля струму

Стандартні коефіцієнти для betafight `ibata_scale`:

- 4in1 80A: 100
- GORO 120A: 50
- SHIVA 6in1 120A: 50
- 6in1 80A: 50
- Одинарний 120A: 66

Для ardupilot `BATT_AMP_PERVLT`:

- 4in1 80A: 100
- GORO 120A: 200
- SHIVA 6in1 120A: 200
- 6in1 80A: 200
- Одинарний 120A: 150

Залежно від налаштувань польотного контролера та підключених до нього інших ADC-пристроїв, покази струму можуть мати невелике зміщення. Тобто навіть коли двигуни не запущені, OSD може показувати споживання струму, наприклад 1–2 А.

Якщо з вимкненими двигунами у вас відображається струм більше ніж 1А, потрібно налаштувати offset.

У Betaflight offset задається в міліамперах.

Наприклад, якщо без навантаження показує 2А, можна прописати: `-200` (це змістить покази приблизно до 0А).

У ArduPilot параметр `BATT_AMP_OFFSET` задається у вольтах. Найпростіший спосіб – виміряти напругу на піні CUR польотного контролера при вимкнених двигунах і прописати це значення в `BATT_AMP_OFFSET` (на відміну від Betaflight, в ArduPilot значення зазвичай вказується без мінуса: 0.3, 1.5 тощо).

Якщо вам потрібне точне вимірювання струму, потрібно відкалібрувати множник струму. Для цього потрібен зовнішній ватметр, який показує реальне споживання. Порядок дій:

- 1) Підключіть ватметр.
- 2) Дайте навантаження, наприклад 9 А за показами ватметра.
- 3) Подивіться, який струм показує OSD.
- 4) Розрахуйте новий множник за формулою:

новий_множник = старий_множник × (струм_на_OSD / струм_на_ватметрі)

Наприклад, якщо ватметр показує 9А, а OSD показує 7А, потрібно змінити множник відповідно до цієї формули: новий_множник = $100 * (7 / 9) = 78$

Щоб зменшити стрибки показників струму, можна збільшити фільтр: `ibat_lpf_period`

25. Файли прошивок

Betaflight (останнє оновлення 01.06.2026)

F405:

http://skypulse.ua/firmware/betaflight/betaflight_4.6.0_STM32F405_SKYPULSEF4V3.hex

H743:

https://skypulse.ua/firmware/betaflight/betaflight_4.6.0_STM32H743_SKYPULSEH7V2.hex

Ardupilot (останнє оновлення 05.05.2026)

F405:

Літак: https://skypulse.ua/firmware/ardupilot/f4v3/arduplane_with_bl.hex

Коптер: https://skypulse.ua/firmware/ardupilot/f4v3/arducopter_with_bl.hex

Ровер: https://skypulse.ua/firmware/ardupilot/f4v3/ardurover_with_bl.hex

H743:

Літак: https://skypulse.ua/firmware/ardupilot/h7v2/arduplane_with_bl.hex

Коптер: https://skypulse.ua/firmware/ardupilot/h7v2/arducopter_with_bl.hex

Inav (бета):

http://skypulse.ua/firmware/inav/inav_8.0.1_SKYPULSEF4V3.hex

26. Питання-відповіді (FAQ)

- Дрон на старті стрибає і перевертається, ніби змінився порядок моторів

Якщо ви у `betaflyght` вмикаєте дрон з режимом `Air mode`, і він довго стоїть на землі заведений (ви не злітаєте), тоді накопичується інтегральна складова підів і дрон може стати некерований при злеті.

Ви можете зробити ваші піди менш агресивними (зменшити множник на вкладці `PID` тюнінг). Також, якщо у вас увімкнений постійний `airmode`, його можна вимкнути і вмикати тумблером за потреби в польоті.

- Не заходить в меню `betaflyght`

Для цього потрібно активувати функцію `CMS` через `cli`:
`feature CMS`
`save`

27. Changelog

2026-03-25

Виправлено переключення між двома приймачами, коли один з приймачів медіаконвертор, інший - ELRS.

Додано маскування OSD.

Додано можливість встановлена кастомної швидкості для UART CRSF.

2026-02-23

Покращено логіку активації каналів для Double-VTX.

Нормалізовано максимум діапазонів VTX-таблиці до 16.

2026-02-01

Покращено поведінку попереджувального біпера Fuse (приглушується в стані armed).

Розширено PWM-режим Fuse: додано опцію dual-servo (fuse_servo_dual) і пов'язані покращення керування.

Додана підтримка одночасного виведення RSSI/LQ на OSD від кількох ELRS приймачів (закрита бета).

2026-01-20

Додано підтримку кастомного blink-повідомлення в OSD (blink_msg) з читанням/записом через MSP.

PWM-значення сервопривода для Fuse стали налаштовуваними (fuse_servo_disabled / fuse_servo_fuse / fuse_servo_charge).

2026-01-08

Додано підтримку Skypulse H743 з двома гіроскопами.

2025-12-18

Додано налаштовувані координати GPS Home та опційний ліміт максимальної дистанції.

Логіка діапазону сервоприводів тепер враховує середню точку (midpoint).

Дрібне виправлення скидання стану Fuse.

2025-12-06

Суттєво розширено функцію Fuse/Charge: більше параметрів таймера, повідомлень і керування; PWM/servo-режим (fuse_by_servo).

Додано можливість вимикати статусні LED (led_enabled).

Скориговано стандартну поведінку біпера, щоб було менше зайвих сигналів (низький заряд за замовчуванням приглушено).

2025-10-10

Виправлено логування температури ESC для конфігурацій з одним ESC.

Виправлено зависання OSD після тривалого сну (35+ хвилин) і покращено пробудження.

28. Немає потрібної функції?

Напишіть нам на hello@skypulse.ua. Ми можемо швидко адаптувати нашу прошивку під ваші потреби (як для Betaflight, так і для Ardupilot).