

# RC Flight Controllers

( 3.6 )

3/2019 - revision 4/2020, 1/2021, 7/2021, 6/2024

[www.dupedup.cz](http://www.dupedup.cz)

I am interested in RC models/drones and like to use **Flight Controllers (FC)**. FC consists of MCU (STM32) and sensors connected via I2C/UART interfaces. The sensor acts as IMU (Inertial Measurement Unit) is 6-axis accelerometer/gyro and a pressure chip sensor acts as a barometer. The IMU sensors are chips MPU6000 or ICM42688 and the barometer BMP280, DPS310 etc. Current, power and temperature sensors are often also embedded. The MCU I/O pins are led out on the board and may be connected via UART to other sensors - GPS modul, compass or pitot tube modules), RC receiver (via Spektrum DSM SRXL2 on 2.4 GHz in my case) or other devices. Some MCU I/O pins are connected to servos via Pulse Width Modulation (PWM) and control flight surfaces for pitch, roll and yaw motion. Some I/O pins are also connected to the regulator (ESC) to control DC brushless motors for an aircraft. The ESC regulator may be controlled by PWM or modern digital protocol e.g. DSHOT.

The base part of the **FC firmware** is a software acts PID regulator and the regulation loop. Inputs are data from sensors and commands from RC receiver. The regulated quantity is the signal to the servos and ESC regulator of motor. This way can be reached the knowledge of 3D position and ability to autoregulate motion of the aircraft. I mostly use open projects as iNAV or Betaflight.

The RC models or drones with the FC are capable to balancing flight, You can use RTH (Return To Home) or Way Points mission capabilities. The great advantage of FC are **Telemetry data** from embedded and connected sensors. Data can be transmitted to a ground station at the pilot. The ground station receive, decode and display flight telemetry data as speed, altitude, tilt, distance etc. via suitable telemetry protocols. I use serial LTM (Light Telemetry Protocol) or MSP (Multiwii Serial Protocol). Telemetry data can be transmitted via radio ISM band 433/868 MHz e.g. LoRa to Ground Station. There can be displayed on a small monitor or computer, save to storage or forward to data network. Bluetooth or WiFi (2.4 GHz) can be used for a short distance.

Also is an option to mix telemetry data as OSD with video from FPV (First Person View) analog/digital camera. Final video signal can be transmitted via board video transmitter (VTX e.g. ImmersionRC) and receive and display FPV video on the ground station. The video signal is transmitted often via 5.8 GHz radio band.

## Specifications some used commercial Flight Controllers (FC)

### SpeedyBee F405 WING AP (2024)

- MCU STM32F405,168MHz,1MB Flash
- IMU(Gyro&Accelerometer): ICM-42688-P
- Barometer :SPL006-001
- OSD Chip: AT7456E
- Blackbox: MicroSD Card Slot
- UART 6 sets(USART1, USART2, USART3, UART4, UART5, UART6 (Dedicated for Wireless board Telemetry connection))
- I2C: 1x Used for magnetometer, digital airspeed sensor
- ADC: 4x (VBAT, Current, RSSI, Analog AirSpeed)
- LED: 12x (11+1"LED"pad)
- ELRS/CRSF receiver: Supported,connected to UART1
- SBUS: Built in inverter for SBUS input (UART2-RX)
- LED. 3x LEDs for FC STATUS (Blue, Green) and 3.3V indicator(Red) 1x RGB
- RSSI Supported Named as RS .
- iNAV: SpeedyBeeF405WING(default) ArduPilot: SpeedyBeeF405WING

### SpeedyBee F405 WING PDB board

- Input voltage : range 7~26V (2~6S LiPo)
- Battery Voltage Sensor: Connect to FC board VBAT, 1K:10K (Scale 1100 in iNav, BATT\_VOLT\_MULT 11.0 in ArduPilot)
- Battery Current Sensor: 90A continuous, 215A peak Connect to FC board Current (Scale 195 in iNav, 50 A/V in ArduPilot)
- TVS Protective diode: Yes
- FC BEC output Output: 5.2V +/- 0.1V DC Continuous current 2.4 Amps, 3A Peak Designed for FC, Receiver, GPS module, AirSpeed module, Telemetry module, WS2812 LED\_Strip
- VTX BEC output Output: Output 9V +/- 0.1V DC Continuous current 1.8 Amps, 2.3A Peak Voltage adjustable, 9V Default, 12V or 5V via jumper Designed for Analog Video Transmitter,Digital Video Transmitter, Camera.

- Servo BEC output: Output 4.9V +/- 0.1V DC Continuous current 4.5 Amps, 5.5A Peak Voltage adjustable, 4.9V Default, 6V or 7.2V via jumper Designed for Servos.

SpeedyBee F405 WING Wireless board

- Wireless Configuration (long press BOOT button for 6 seconds to switch modes):
- BLE mode – (green) connect to Speedybee APP
- Wi-Fi mode – (white) connect to QGroundControl APP, Speedybee APP, MissionPlanner, etc.;
- Classic Bluetooth SPP mode – (blue) connect to QGroundControl APP, MissionPlanner
- LED strip controller: (short press BOOT button to switch effects, long press 3 seconds to switch modes):
- 4x WS2812 LED strip connectors, adjustable colors and flashing modes)
- Max 5.2V 1.3A, supports around 68pcs 5050 WS2812 LED beads
- On-board battery level indicator: 4x RGB indicator LED for battery level display by number of lights

#### **MATEK F405 WING (iNAV MATEKF405SE) (2019)**

- MCU: STM32 F405 (Cortex™-M4 core, 168 MHz, 192 KB RAM, 1 MB FLASH)
- IMU: MPU6000 accelerometer/gyro (SPI)
- Baro: BMP280 (I2C)
- OSD: iNAV OSD w/ AT7456E chip
- Blackbox: MicroSD card slot (SD/SDHC)
- VCP & 6x UARTs ((e.g. for GPS modul , RC receiver, Telemetry modul, etc.)
- 9xPWM (2x Motors, 7x Servos outputs)
- 2x I2C
- 3x LEDs for FC STATUS (Blue, Red) and 3.3V indicator (Red)
- Built in inverter for SBUS input (UART2-RX)
- PPM/UART Shared: UART2-RX
- SoftSerial on TX2 pad
- Battery Voltage Sensor: 1:10 (Scale 1100)
- WS2812 Led Strip : Yes
- Beeper : Yes
- RSSI: Yes
- Mounting: 30.5 x 30.5mm, Ø4mm with Grommets Ø3mm , Dimensions: 56 x 36 x 13 mm Weight: 25g

#### **MATEK H743 WLite (iNAV MATEKH743)**

- CPU: STM32H743VIH6, 480MHz, 2MB Flash , IMU: ICM42688-P (SPI1), Baro: DPS310 (I2C2)
- OSD: AT7456E (SPI2)
- Blackbox: MicroSD card slot (SDIO)
- 7x Uarts (1,2,3,4,6,7,8) with built-in inversion
- 13x PWM outputs
- 2x I2C, 1x CAN
- 6x ADC (VBAT, Current, RSSI, Analog AirSpeed, VB2, CU2 ) , 1x JST-GH\_4pin connector for I2C2, 1x JST-GH\_4pin connector for CAN
- 1x JST-SH\_6pin connector for external USB and buzzer
- Dual Camera Inputs switch
- 9V(12V) for VTX power switch
- 6.8~30V DC IN (2~6S LiPo)
- High-precision Current Sense 220A Range
- BEC 5V 2A for FC, BEC 9V 2A for camera/VTX, 12V option, BEC Vx 8A cont. 10A Peak for servos, 5V, 6V or 7.2V option
- LDO 3.3V 200mA
- iNAV Target: MATEKH743
- 4V5: 4.4~4.8V, Max.500mA, 5V: onboard BEC 5V 2A cont. Max.3A, 9V: 9V output, 9V will increase to 12V if "12V"
- Vbat: Battery voltage
- onboard battery voltage sense: iNAV scale 2100
- Curr: Current signal (0~3.3V), onboard current sense: iNAV scale 150, CU2: for external current sensor, 0~3.3V

#### **AIRBOT OMNIBUS F4V6 (iNAV OMNIBUSF4V6) – for WING suitable for telemetry or FPV (2019)**

- MCU: STM32 F405
- BARO: BMP280
- IMU: MPU6000 SPI
- 6PWM output
- 5x UART (e.g. for GPS modul , RC receiver, Telemetry modul, etc.)
- 5V1A BEC on board(3-6s) 8V BEC Camera
- Flash, Only 36x36mm, holes 30.5x30.5mm

#### **AIRBOT OMNIBUS F4Pro(V3) (iNAV OMNIBUSF4PRO) or (Betaflight OMNIBUSF4SD) (2019)**

- MCU: STM32 F405 , BARO: BMP280, IMU: MPU6000 SPI
- 6PWM output
- 3x UART (e.g. for GPS modul , RC receiver and Telemetry modul)

- On-Board OSD (FC over SPI bus)
- MicroSD Blackbox
- 5v3A SBEC
- Built-in Current Sensor
- On-Board Video Filter 5V to VTX and Camera

**Holybro Kakute F7 mini V3 + Betaflight** – for WING suitable for telemetry via LTM

- MCU: STM32F722RET6 32-bit processor, 216MHz, 256Kbytes RAM, 512Kbytes Flash, IMU: MPU6000 (SPI)
- Barometer: BMP280
- USB VCP Driver (all UARTs usable simultaneously; USB does not take up a UART)
- 5 hardware UARTS (UART1,2,3,4,6)
- Supports serial receivers (SBUS, iBus, Spektrum, Crossfire) only. PPM and PWM receivers are not supported.
- Onboard 16Mbytes for Blackbox logging
- 5V Power Out: 1.0A max, 3.3V Power Out: 0.2A max
- Dimensions: 30x30mm
- Mounting Holes: Standard 20mm square to center of holes (M4)
- Weight: 4g

**Holybro Kakute F4 V2 + Betaflight** – WING suitable for telemetry via LTM

**SucceX-E F4 V2.1 + Betaflight** - suitable for telemetry via LTM

**Connected Components**

- **RC Receiver:** SPEKTRUM Serial Receiver 2,4 GHz SRXL (4649T) or SRXL2 (SPM4650), 3.3-8.4 V, 1.4g
- **GPS Modul:** u-blox NEO 6M or Beitian BN280, BN220, BN880
- **Camera FPV :** RunCam Robin, FOXEER Arrow Mini Pro, PAL, 5- 40V, 75 mA/5V, 8.7 g
- **Video FPV Transmitter** 5,8 GHz (VTX): Immersion RC TRAMP HV, 1mW to > 600mW, 2-4S Lipo, 4W / max. 600mW, SMA, 4g
- **Video FPV Receiver** 5.8 GHz: RC700D (diversity), TFT LCD, 2 x SMA, TF Card, AV IO, 1.3 Sonny CCD
- **Telemetry:** HC-06 Bluetooth, HC-12 433 MHz, E32-TTL-100 LoRa 433 MHz, TTGO ESP32 LoRa/LoRaWAN 866 MHz
- **ESC:** DYS Aria 35A 32bit + firmware BL\_Heli32 (v.32\_64), (SBEC (switched BEC) FOXY UBEC 5V/3A)
- **RC Transmitter:** SPEKTRUM 2,4 GHz 8G2
- **Antena VTX** TrueRC Core SMA, 80mm, RHCP, 1.9 dBi, ř.5-6 GHz

Kdysi nejdříve jsem použil levnou desku FC **AIRBOT OMNIBUS F4V6**. Desky tohoto typu jsou zejména určeny pro řízení koptér než křídlových (FIXED WING) modelů se servy. Deska je napájena z 3-6S LiPo a obsahuje MCU STM32, BEC pro napájení periférií 5V, čip barometru, čip gyra a akcelerometru (IMU), čip OSD a senzory napětí a proudu. Z MCU je vyvedena šestice seriových UART (Rx/Tx) a šestice PWM výstupů pro řízení serv a regulátoru (ESC) motorů.

Do FLASH paměti FC resp. MCU lze přes USB (po přepnutí MCU do DFU modu tiskem tlačítka BOOT při zapnutí napájení) „nahrát“ letový firmware. Desky FC mají USB bootloader a do OS počítačů je možno doplnit USB driver pro STM32. Firmware je možné si vyvinout nebo (spíše) použít volný a komunitou vyvíjený software. Takovým je např. **BETAFLIGHT** <https://betaflight.com>, **INAV** <https://github.com/iNavFlight> nebo **Ardupilot**. Jsou pro většinu komerčních FC desek ke stažení v binárním kódu a lze je do příslušného FC přes USB „flashovat“ a následně nastavovat přes přehledné grafické konfiguratory nebo příkazovou řádku CLI v počítači (s OS Windows/Linux). Volné firmware mají výborně zpracovaný popis. Pro přenos konfiguračních dat mezi firmware a GUI konfiguratory se používá seriový protokol MSP.

K jednomu ze seriových portů (UART) FC jsem připojil – tzn. Tx do Rx a naopak + napájení 5V, nejdříve **GPS modul** (ublox NEO-6M, později Betian BN-220) a k video vstupu analogovou **FPV kamerku** RunCam (později Foxeer). K video výstupu FC připojuji **video vysílač VTX** (ImmersionRC Tramp HV) v pásmu 5.8 GHz. Parametry FC je v konfiguratoru možné resp. nutné nastavit dle potřeby. Jde zejména o parametry seriových portů, letové mody a lze i jednoduše editovat rozložení telemetrických hodnot vložených do videesignálu na obrazovce a fonty znaků

Obraz z letové kamery zobrazují na **PAL monitoru 720x576** se zabudovaným přijímačem (Skyzone SKY 700D 5.8GHz 32CH Diversity 7" FPV). Je nutno nastavit stejný frekvenční kanál v pásmu 5.8 GHz jako na vysíláči pomocí přepínačů. Nastavuji CH1 v A rozsahu (5705 MHz)

Frekvenční kanál v pásmu 5.8 GHz lze nastavit na vysíláči VTX pomocí LED a tlačítek. Lepší je připojit MENU vstup Trump IRC na Tx nějakého UART FC a nastavovat radio kanály softwarově pomocí příkazů firmware iNAV/BF. Po připojení RC lze tak i vyvolat OSD setup MENU Trump a další nastavení VTX provádět takto. Kouzelné vyvolání OSD menu je konkrétním nastavením PWM ovladačů RC - výškovka plně potlačit, plyn uprostřed a směrovka plně vlevo.

Na RC model jsem připevnil desku FC OMNIBUS F4V6 s GPS modulem, FPV kamerkou RUNCAM a video vysílačem VTX Tramp Immersion. Zkouška letu s touto FPV sestavou se vyvíjela velice pěkně a letový obraz na pozemním monitoru s vloženými OSD textovými údaji telemetrických hodnot z FC je super. Oproti jednoúčelovému OSD (např. Tarot) je více letových údajů a např. umělý horizont.

Pro ovládání letu modelu pomocí FC připojuji **PWM výstupy** k ESC motoru a na vstupy serv křídélek, výškovky a směrovky. V Betaflight jsou na deskách OMNIBUS defaultně konfigurované výstupy mapovány pro mix čtyř ESC motorů coptery, ale lze je konfiguračně „přemapovat“ (příkaz "resources") pro mix čtyř serv a jednoho či dvou ESC motorů (příkazy "smix" a "mmix").

K Tx/Rx jednoho z UART na FC připojuji seriový **RC přijímač** Spektrum 4649T (až RC 15 kanálů). Používá seriový protokol SRXL (nově SRXL2) a firmware iNAV/BF ho umí na většině UARTs automaticky rozpoznat. Přijímač má navíc možnost přenášet zpět do RC vysílače telemetrická data z FC pomocí bi-directional SRXL protokolu. *Aktualizace 2021 – Spektrum již používá pouze SRXL2 4650/4650T - iNAV i Betaflight SRXL2 podporuje.*

Pak jsem pořídil (cca 500 Kč) ještě letový kontroler **AIRBOT Omnibus F4ProV3**. Tato deska je o něco starší než F4V6, ale má přímou podporu v současné verzi iNAV (2.1) i pro mix 4 serv a 2 motorů. Mohu na ní zapojit všechny výstupy, tedy všechny ovládací plochy a dva motory. Tento FC obsahuje i integrovaný proudový snímač dimenzovaný na vysoký odběr. Proud pro ESC jde přes tento snímač a měří se celkový proud z AKU - zobrazují jej v OSD.

Nakonec jsem pořídil dražší (cca 1000 Kč) desku **FC Matek F405 WING** přímo výstupy uzpůsobenou pro modely letadel a nahlál na ni poslední **iNAV** (začínal jsem s verzí 2.1 a 2024 je již verze 7.1 !). Nadále používám pro stabilizaci a navigaci modelů letadel většinou FC desky od výrobců MATEK nebo SpeedyBee s firmware INAV.

**Deska FC musí být v trupu modelu správně orientovaná a pevně usazená.** Orientace lze ve firmware upravit, ale defaultně má šipka na desce RC mířit k přidi. RC přijímač, GPS modul, VTX s Heli anténou pro 5.8 GHz a případně i vysílač Telemetrie 868 MHz (např. LoRa – viz dále) s krátkou anténkou pro toto pásmo mám upevněny většinou na trupu či u kořene křídél. FPV či HD kamera je umístěna vpředu na přidi. Vše většinou stačí upevnit oboustrannou lepicí páskou nebo jen suchým zipem. Přívody od serv, ESC a napájení jsou protažené trupem k FC.

**Důležité je zkalibrovat akcelerometr a srovnat výchylky serv ve správném směru** dle vyrovnávaného pohybu z gyro snímače:

- pohyb ocasu nahoru -> výškovka vyrovnává pohybem nahoru
- pohyb ocasem dolů -> výškovka vyrovnává pohybem dolů
- náklon vpravo dolů -> pravé křídélko vyrovnává pohybem dolů a levé nahoru
- náklon vlevo dolů -> levé křídélko vyrovnává pohybem dolů a pravé nahoru

- otočení přidě vpravo -> směrovka vyrovnává otočením vlevo.
- na pokyn z vysílače do pravého náklonu, pravého točení a potlačení se musí RC signál v FC zvyšovat

Správné směry výchylek lze srovnat kombinací reverzu nastavení serv v nastavení (output) a RC vysílače. Vhodné je pro testy zvýšit zisky PI, aby výchylky byly větší a trvaly déle. Pak ale nezapomenout vrátit se k původním hodnotám PI!

Doplňuji k FC **pásek s 5 x RGB LED**. Tím dostávám pomocí barev jednotlivých LED signalizaci stavu AKU, GPS, ARM apod. Lze doplnit bzučák na zvukovou signalizaci.

Pro **zasílání telemetrie** připojuji k FC přes další UART radiový modul HC-12, nebo později modul LORA, na 433/866 MHz - viz článek Telemetrie.

Mírný nárazový vítr je ideální čas na **test letových modů** FC iNAV. Pomocí funkce „blackbox“ lze zaznamenávat letové údaje a údaje gyra na interní 16MB flash v FC, po přistání data stáhnout a dekodovat si je např. toolem iNAV Blackbox Explorer. Lze pak zkoumat údaje telemetrie a stavy gyra. Na black box záznamu z dnešního letu lze zobrazit chování modelu s FC v různých letových modech nastavených na AUX přepínačích vysílače „namapovaných“ na jednotlivé RC kanály.

Nejdříve musí mít FC splněny podmínky pro „odjištění“ motoru **ARM**. Zejména funkční HW, nastavený počet GPS satelitů (default 6), nezapnutý NAV mod či nepřipojený konfigurátor. Důvod nemožnosti ARM je možné vylistovat v CLI „status“. Pak lze vzlétnout a to raději v modu **MANUAL**, kdy FC nemá žádnou kontrolu nad řídicími plochami modelu.

Po vypnutí manuálu a nezapnutí jiného modu se použije defaultní **stabilizační mod ACRO (RATE)**, FC přebírá kontrolu nad výchylkami řídicích ploch, sleduje signál z gyra a v případě nechtěných rychlých změn pohybu (např. nárazy větru) se snaží pomocí serv řídicích ploch vrátit model do původní polohy a let stabilizovat. Pak zapínám modu **ANGLE/HORIZON** FC navíc ke gyru bere v úvahu i signál z akcelerometru a snaží se regulací křidélek a výškovky držet stabilní vodorovný přímý let. Náklony letu mohou ovládat i z RC vysílače, ale jen do určité míry (u ANGLE) resp. od určitého prahu náklonů se tato regulace vypne (u HORIZON) pro možnost akrobacie. Výška regulovaná v těchto modech není a směr je ideální ovládat jen směrovkou, kde zůstává RATE.

Je možné používat též **navigačních modů**, ale musí být aktivní ACRO mod a být plně funkční GPS. Při modu **NAV COURSE HOLD** je model stabilizován a snaží se autonomně držet nastavený kurs dle GPS. U modu **NAV CRUISE** (Altitude Hold + Course Hold) se model navíc snaží držet nastavenou výšku dle hodnot z barometru či GPS. Používá se zde regulace otáček motoru pro regulaci příčného náklonu modelu.

Po zapnutí modu **NAV RTH** (Return To Home) model autonomně stoupá do přednastavené výšky (např. 60 m) a míří na startovní pozici zaznamenanou z GPS při zapnutí **ARM** nebo volbou Reset Home Position. Nad startovní pozicí pak krouží (Loiter) dokud se pilot nesmiluje nebo se nevybije AKU. Kroužit („zaparkovat ve vzduchu“) je možné i zapnutím modu **NAV POSHOLD** (Position Hold). Pro autonomní let dle přednastavených GPS bodů lze použít i mod **WP** (Autonomous Waypoint Mission).

Pokud by za letu přijímač na modelu ztratil RC signál, model by přešel do **Failsafe** a pro tento případ mám nastaven přechod do autonomního modu RTH (lze nastavit i přistání **AutoLanding**).

Regulace výstupů FC pro motory a serva je pomocí sw regulátoru **PID**. Regulační veličinou je defaultně střída PWM signálu (1000-2000 mikrosek při 50 Hz) pro ESC motoru a serva řídicích ploch s cílem stabilní pozice v prostoru. Velikost regulační odchylky je ovlivněna vstupy z RC kanálů z přijímače a hodnotami z připojených senzorů - barometru, gyra, akcelerometru a (případně) GPS. Zpětná vazba a velikost regulační odchylky je počítána:

- **proporcionálně (P)** – čím větší odchylka od normálu je, tím více je potřeba regulovat
- **integračně (I)** - počítá se z předchozích hodnot a bere tím v úvahu čas a zmírňuje trvalou odchylku regulace
- **derivačně (D)** - derivuje okamžitých změn a tím potlačuje např. šum regulačního snímače

Úroveň hodnot PID regulace je možno uživatelsky ovlivnit jejich násobkem. Správné nastavení PID je poměrně věda, velké P má za následek rozkmitání výstupů, malé I stálou regulační odchylku a velké D úplně utlumení a dlouhé opoždění regulace. Vše je komplikováno vibracemi z motorů, šumem senzorů, atd.

Pro modely letadel se tu používá spíše **PIFF**, kde derivační složka je potlačena a zisk složky FF (**Feed Forward**) udává nakolik bude regulátor brát vážně i ruční zásahy do řízení.

Koeficienty PID a další hodnoty pro regulaci např. maximální možná rychlost otáčení modelu kolem jednotlivých os (**rates** v deg/sec) či potlačení PID hodnot při vyšším plynu (**TPA**) se nastavují v konfigurátoru.

PID zprvu nastavuji cca:

- zisky (gain) P-15, I-12, FF-60, D-0,
- rate roll 150, pitch 100, yaw 90 deg/sec.

V případě kmitů PID je nutno zmenšit P, v případě stálé odchylky směrů (driftování díky stálé regulační odchylce) pomalu zvětšovat I. Firmware \*\*iNAV obsahuje možnost zapnutí Autotunnig při letu, ale nakonec je lepší si s nastavením zisků PIFF a dalších hodnot pohrát při pokusných letech.

## Digitální ESC protokol DSHOT

Pro řízení motoru přes ESC na křídlových modelech stačí klasická analogová PWM modulace, ale pro řízení motoru pomocí FC je už PWM "zastaralé" pro koptery. Nové ESC protokoly, např. digitální DSHOT, jsou zesynchronizovány s regulační smyčkou PID a motory rychleji reagují na změny.

**DSHOT** je digitální protokol mezi FC a ESC. Délka jednotlivých pulsů v cyklu definuje bit, tedy 0 nebo 1. V jednom cyklu má 11 bitů (2000 úrovní) na řízení throttle a jeden bit jako požadavek na seriové slovo telemetrie přes separátní seriový kabel zpět do FC a 4 bitové CRC. U DSHOT150 je délka cyklu cca 0,1 ms, je 200x rychlejší reakce pro řízení throttle než 20 ms u běžného PWM. Navíc digitální ESC s podporou DSHOT umí posílat telemetrii (otáčky).

Požívuji si **ESC DYS Aria 35A**. Ten obsahuje 32b CPU ARM s [firmware BLHeli\\_32](#).

Nastavení či update firmware BLHeli v ESC lze provést pomocí konfigurátoru BLHeliSuite\_32 a tzv. „Passthrough“ - ESC se připojí k motoru i k FC, spustí se iNAV nebo Betaflight (bez Connect k seriovému portu) a v konfigurátor BLHeliSuite se nastaví v menu „bootloader Betaflight“ pro seriový port který se ukazuje v iNav/BF. Konfigurace/update firmware v ESC pak probíhá přes USB na FC.

Nové verze iNAV by měly podporovat **ESC Telemetrii**, tedy posílání telemetrie z datového vývodu ESC (např. můj nový regulátor DYS ARIA s 32b MCU s firmware BLHeli\_32 ho má označen TE) na volný UART na FC. Zkouším DSHOT zatím s Betaflight na F4V6 Omnibus a s iNAV + MATEk405). Nastavuji ESC DSHOT 120 a propojuji UART4 s datovým výstupem "TE" na desce ESC. Daří se pak zobrazit v OSD např. otáčky motoru (RPM), teplotu desky ESC a napětí.

Laboruji ještě s výstupem na destičce ESC DYS s označením "CU" (analogový CURRENT) a jeho připojením na analogový vstup pro proudový snímač na desce F4V6.. Snímat proud mimo snímač v FC se mi nedaří.

Nové verze firmware pro FC podporují nyní i **Bidirect Telemetry DSHOT**. Pro funkční „bidirect“ je nutné flashnout ESC DYS BLHeli\_32 na poslední verzi (nyní 32.7). Datové pakety telemetrie jsou pak obsaženy přímo v odpovědích DSHOT regulačního protokolu a není nutný datový propoj z ESC na UART desky FC. BF umí data z paketů DSHOT načíst a online nasnímané hodnoty otáček, napětí a proudu z ESC zobrazit, např. v OSD. Navíc nasnímané hodnoty může používat firmware jako vstupní hodnoty pro dynamické nastavení softwarových frekvenčních filtrů (RPM filtrů).

Dynamické filtry analyzují gyrosignál a ořezávají šumy resp. vyšší frekvence signálu (LPF-LowPassFilter, např. od 90Hz) nebo vyřezávají některé (NotchFilter) frekvence signálu (200-300Hz). Tím umožňují zvýšení odstupe užitečného signálu gyra od šumu z vibrací motorů/konstrukce trupu a tím zpřesnění PID regulace. Filtr RPM umožní dynamicky stanovit filtrovanou frekvenci dle aktuálních otáček motoru a tím pravděpodobných vibrací přenášených na gyro.

## Telemetrie

Na pozemní stanici (**GCS - ground control station**) bych rád on-line přijímat telemetrická data z modelu - tedy GPS, rychlost, výšku, kurs, palubní napětí/proud a náklony. Zároveň chci mít možnost online uploadovat GPS souřadnice z pozemní stanice do FC (Way Points) na modelu pro lety Flight Mission.

Jako telemetrický protokol lze u FC s firmware Betaflight/iNAV použít MultiWii Serial Protocol (MSP), LTM (Lightweight Telemetry) nebo MAVLink. Seriový telemetry protokol **MSP** je „tahací“ a vyžaduje vyšší datovou rychlost (nad 9600 bps). Aplikace v pozemní stanici si musí o paket dat nejdříve říci.

**LTM** je „tlačící“, v přednastavené četnosti některá telemetrická data automaticky posílá a je vhodnější pro nižší přenosové rychlosti (500 – 9600 bps). LTM v zasílaných paketech poskytuje méně telemetrických dat, ale pro zobrazení telemetrie z modelu to dostačuje. V iNAV lze nastavit (ltm\_update\_rate) četnost zasílání LTM dat od 105 B/sec do 302 B/sec tak, aby korespondovala s používanou přenosovou rychlostí (pro 9600 mám 303 B/s). Kvůli radio duty cycle lze jít i níže.

Pro příjem telemetrie a přípravu/upload way points na pozemní stanici lze použít např. free [mwptools](#) na LINUX nebo aplikaci [Mission Planner](#) na Android. Ve Windows lze použít iNAV Configurator. Lze si vyrobit vlastní aplikační projekt (viz dále).

Pro radiový přenos dat z model použiji nejdříve Bluetooth moduly (HC-06), pak zkusím radio moduly pro radiový přenos v bezlicenčním pásmu 433 MHz (např. HC-12) a cílově použiji protokol LoRa (LongRange) např. s moduly E45-TTL-100. LoRa používá efektivnější modulaci CSS (Chirp Spread Spectrum) než HC-12 (modulace PSK) a zvládne tak větší dosah. Moduly lze získat v „Arduino shopech“ za cenu v řádu desítek/stovek Kč.

**Bluetooth moduly HC-06** je vhodné nastavit na max. UART rychlost 115200 bps (defaultně má HC-06 nastaveno 9600 bps). K tomu stačí modul USB/UART převodníku (např. CP2102) seriově propojit s BT modulem HC-06 (Vcc->5V, GND->GND, Rx->Tx, Tx->Rx), použít USB PC či notebooku, nainstalovat do OS driver pro virtuální COM port (VCP driver) a z nějakého seriového terminálu (např. Termit) přes COM port (9600/8N1) do modulu poslat konfigurační AT příkazy. HC-06 má nový firmware (3.0) a často dokumentované AT příkazy vrací chyby. Nová verze má AT příkazy:

- pro změnu rychlosti AT+UART=115200,0,0
- pro změnu jména AT+NAME:namestring
- nastavení pinu "AT+PSWD:"pinstring" (default 1234)

V iNAV lze na volném UART FC nastavit zasílání telemetrie **protokolem MSP** s 115200 bps. Lze použít současně MSP/LTM na 9600 bps. iNAV použije nejdříve MSP a po odjištění motorů (ARM) se změní na LTM.

Pak postačí připojit BT modul HC-06 (5V, GND, Tx->Rx, Rx->Tx) k UART na desce FC. Pro RX vstup modulu by se měl připojit přes **převodník logických úrovní 5/3.3V** nebo odporový dělič z 1k/2k - logika modulu používá 3.3V. To platí u většiny „arduino“ radio modulů.

Ted' přijde instalace [mwptools](#) pro příjem telemetrie do notebooku. Aplikace je pro Linux, což do Windows instaluji VMWARE Workstation Pro 15 (2024 - 17.5.2) a do něj virtuální Ubuntu 18.04 (2024 - 24.04). U starších VW je třeba zakázat "Device/Credential Guard" . Po instalaci Ubuntu do virtuálu, doplnění potřebných balíčků, poslední verze zdrojových kódů mwptools z Githubu a provedení build je funkční mwptools na světě. Ve Windows 11 lze použít i Windows Subsystem for Linux (WSL2) a spustit tool serial2UDP a ve mwp se připojit na sériová data z BT pomocí UDP. To se mi nejevilo jako stabilní.

**Bluetooth** modulem HC-06 je potřeba spárovat s Windows s BT (ve Win 11 je třeba změnit „rozšířené“ vyhledávání zařízení). Je vhodné v konfiguraci BT device nastavit **COM porty (COM5/6)** pro příjem/zasílání dat z BT do z/do seriového portu. Pak by mělo stačit ve vmware pro VM s Ubuntu doplnit virtualizaci seriových portů, ve virtuálním Linux zkontrolovat jejich připojení (sudo setserial -g /dev/ttyS[0123]), případně nastavit práva na jejich použití (sudo chmod 666 /dev/ttyS0) a připojit mwp např. k ttyS0. V grafice mwptools jsou pak pěkně zobrazená telemetrická data ze snímačů připojených k FC na modelu, je tu umělý horizont, mapa a trasa letu i hlasový výstup. Je možné online do FC poslat až 60 WP bodů pro Flight Mission. Aplikaci mwptools si lze přizpůsobit pomocí klíčů v Gsettings (schema je org.mwptools.planner) v Linuxu.

Bluetooth má dosah do 10 m, propojením přes **radio 433 MHz** pomocí **modulů HC-12** a malou anténkou pro tato pásma dosahnu cca 500 m. Vysílací modul HC-12 připojuji přímo k seriovému portu na desce FC (UART2). V iNAV na tomto UART nastavuji telemetrie protokol na **LTM / 9600 bps** . Přijímací HC-12 modul propojuji přes UART 9600 bps s osvědčeným Bluetooth modulem HC-06 (pomocí AT nastavuji zpět 9600 bps). HC-12/HC-6 napájím step down/LDO modulem 3,3V z 2-3 článkového LiPo akumulátorku. Přes takto vytvořenou „gateway“ přijímám telemetrii LTM (resp. nejdříve MSP a následně po ARM na modelu přes LTM) přes Bluetooth do mwptools.

Při dobré konstelaci a nízké datové rychlosti je dosah na 433 MHz až 500 m. Parametry lze u HC-12 modulů nastavit pomocí AT příkazů. Výkon lze nastavit až na 100 mW (v ČR je povolených jen 15 mW !! v pásmu 433 MHz a duty cycle 1%). Rychlost UART na modulu použiji defaultních 9600 bps (lze přenastavit pomocí AT+Bxxx), což je dle manuálu rychlost 15000 bps „in air“.

Nejnověji používám radiové propojení **radio LoRa 868 MHz a LoRa modulů** s anténkou pro tato pásma a s dosahem i více než 1 km. Ideální je LoRa modul **E32-TTL-100**, který má seriový interface. Lze použít i ESP32 s LoRa (TTGO) nebo podobné moduly. Vysílací radiový LoRa modul je na modelu připojen přes UART k FC. Použije se protokol LTM 9600 bps. Přijímač je tvořen obdobnou gateway jako s HC-12, tedy propojením radiového LoRa modulu s BT modulem. Připojením BT modulu do OS je pak možno LTM data zobrazovat v mwptool nebo si do Windows naprogramovat vlastní aplikaci chytající data z BT nebo COM portu a graficky je pak zobrazovat. Na testy zobrazení LTM paketů je možné použít seriový terminál.

Vhodným „Arduino compatible“ MCU a doprogramováním firmware lze vytvořit vlastní projekty „krabicové“ pozemní stanice s display bez nutnosti použít na zobrazení dat mobil nebo notebook.

### **Jednoduchá Ground Station pro příjem telemetrie s MCU a LoRa modulem**

Jako příklad „krabicového“ řešení jsem si (2021) vytvořil malou přijímací stanici, která je tvořena **MCU ESP32 VROOM32D** (2024 - STM32F103C8 BluePiLL ARM 32).

LoRa modul je MCU připojen přes UART. K MCU jsem dále připojil TFT display (SPI interface) a SD čtečku (SPI interface). Viz schema. Vše je napájeno 3.3V z napěťově stabilizačních LDO a 3S LiPo akumulátoru. Do MCU jsem přes Arduino IDE naprogramoval firmware, který rozděljuje přijatý LTM datastream z LoRa na seriový port MCU na jednotlivé hodnoty telemetrie, zobrazuje je na display a zapisuje na SD kartu do binárního souboru. Pro čtení souboru jsem si v C do Windows GUI doplnil jednoduchou aplikaci, která ze zaznamenaných letových dat může dělat různé statistiky.

Pokud je v místě signál LoRaWAN, je možné posílat telemetrie data i na bránu sítě LoRaWAN. V tomto případě v menší frekvenci resp. v menším datovém toku a množství - např. jen občasná GPS data. Z řídicího serveru LoRaWAN lze pak data přes API pomocí callback a http POST data přeposílat na REST webovou aplikaci na síťovém serveru a zde je zaznamenávat jako "remote on-line blackbox".

