

General Description

The Mercury-H7 is a flight controller (FC) designed for small, manually-controlled quadcopters and fixed-wing aircraft. It can be mounted into frames that include 30.5×30.5mm square mounting patterns that fit M3 (3mm diameter) screws respectively. It includes a STMH743 microcontroller, and is compatible with ArduPilot, Betaflight, Px4, and iNav.

Mercury is designed to be connected to motors via a DSHOT-capable Electronic Speed Controller (ESC), and a digital video transmitter (eg DJI, HDZero, Walksnail etc). It can also be connected to an external Global Navigation Satellite System (GNSS) such as GPS, a magnetometer, and other sensors. The FC provides JST SH and GH connectors for this purpose. Additionally, there are labeled solder pads on the bottom of the flight controller than may be used in lieu of these connectors. Sections below provide details on connecting to these systems.

Mercury is DroneCAN capable, and plug and play with DroneCAN systems; it includes an onboard transceiver capable for CAN-FD frames at up to 5Mbps.

Mercury is designed to be used with digital video systems only: It provides no analog on-screen-display functionality.

Mercury includes a full ExpressLRS (ELRS) control receiver radio, with a dedicated microcontroller that runs official ELRS firmware. This can be used for control data only, telemetry only, or control data and telemetry combined.

Important: **Do not use Mercury with a battery voltage higher than 26V.** This means, no more than a 6S (3.7v LiPo) battery. In order to be used with DJI or HDZero Vtx systems, use no smaller than a 3S battery. This means the battery cell count must be between 3S and 6S.

Specifications

- **Dimensions:** 37.5×37.5×8 (height) mm
- **Weight:** 8 grams
- **Mounting holes:** M3 (3mm diameter). 30.5mm × 30.5mm, square pattern
- **Power:** 6V– 26V (2-6S LiPo. Digital Vtxes require 7V, so use a 3-6S LiPo)
- **MCU:** STM32H743. 170Mhz Cortex-M7, 2Mbyte internal flash, 768kb SRAM
- **Inertial Measurement Unit (IMU):** TDK ICM-42688p, with dedicated crystal
- **Barometer (pressure altimeter):** Infineon DPS-310

- **Storage:** 64Mbit “black box” flash: W25Q64
- **Power and flashing:** USB-C (DFU), and SWD
- **ESC pin header:** JST SH, 1mm pitch, 8 pins (Betaflight layout)
- **Digital Vtx / cam pin header:** JST SH, 1mm pitch, 6 pins, (DJI layout)
- **Free UART connections:** 4. (UART7 and UART2 are dedicated to ELRS Rx and digital OSD respectively; UART1, UART3, UART4, and UART8 are available)
- **External I²C buses:** 1
- **Control radio receiver:** Semtech SX1280. (Compatible with ELRS transmitters)
- **Power pad voltages:** VBAT, 9V, 5V, 3.3V
- **Current capacity:** 3A on the 9V and 5V output lines combined. No more than 2A on the 5A line. 500MA on the 3.3V line.

Connecting to an ESC

When used for quadcopters, Mercury is designed to work with a “4-in-1” Electronic Speed Controller (ESC). This ESC is responsible for supplying power to the aircraft via a battery connected to it, and for controlling signals sent to brushless motors, also connected directly to the ESC. This ESC should use the DSHOT protocol to receive power commands from the FC, and send measured RPM back to the FC along the same line.

When used for fixed-wing aircraft, the ESC is likely to only need connections to 1 or 2 motors, so a single-motor ESC may be sufficient. A DSHOT-capable ESC is recommended, but PWM ESCs work as well. PWM ESCs are sufficient for the update rates required for the thrust-providing propellers used on fixed-wing aircraft, but lack the signal robustness and error-checking functionality of DSHOT.

To connect to an ESC, use either the 8-pin Betaflight standard ESC connect (JST SH: 1mm pitch) on the bottom of the FC, or the solder pads labeled “ESC” on the top. This header and these pads contain the following connections. Note that these are in the same order as Diatone FCs. They are in the *reverse order* of Diatone ESC connections. The colors labeled below apply to the included cables.

It may help cable management to twist the ESC connector cable a few times.

- **Ground.** This is connected to the battery’s negative (black) terminal, and provides a common ground voltage reference for the entire aircraft. Passed-through by the ESC.

- **Battery** in (+Bat). This is connected to the battery’s positive (hot) terminal, and is nominally 7.4 to 22.2V, depending on battery cell count. This, stepped down to various lower voltages (mainly 3.3V) provides power to all systems on the aircraft connected to the FC.

- **Motors 1-4.** This is used to send motor power commands via DSHOT from the FC to the ESC, and for the ESC to send RPM measurements to the FC.
- **ESC telemetry.** This provides data from the ESC to the flight controller over a UART channel. It is connected to the UART3 Rx line on Mercury. It typically provides ESC temperature, and RPM data. The RPM data is typically at a lower update rate than that provided by DSHOT. To use in firmware (eg Betaflight), configure your firmware to use ESC telmetry on UART3. This connection is optional.
- **ESC current.** This outputs a voltage proportional to the total current consumed by the ESC at a given time. The Betaflight config has this scaler value included by default; it is provided here for reference. This connection is optional.

Connecting a digital video transmission system

Mercury is intend to be used with a digital video transmissions system, such as DJI, HDZero, and WalkSnail. These connect to Mercury using a 6-pin JST SH (1mm pitch) pin header or 6 solder pads, both on the top of the FC. These have the following connections. Note that these are the same as used by the DJI O3; it is plug and play with that system.

- **Battery in (+VBat).** Note that Mercury doesn't have a dedicated voltage regulator for the video transmitter; it passes the battery voltage through, after passing through a TVS diode that limits voltage spikes; no-more-than 26 volts ever passes through this pad. Note that at least 7V of the battery must be available to support DJI and HDZero systems.
- **Ground** (to power the camera and transmitter),
- **UART 2 Rx and Tx** connections (To receive information to populate the On-Screen Display (OSD)).
- **A second ground connection**, for use with DJI radio controllers. Optional.
- **A UART 1 Rx HDL data connection**, for use with DJI radio controllers. Optional. If using this line, don't connect anything to the *Rx1* solder pad in the UART section of the FC.

To connect to a Vtx/cam, use either the 6-pin ESC header (JST SH: 1mm pitch) on the top of the FC, or the pads labeled "Vtx/Cam" on the top. The *HDL* and its adjacent ground connection are optional: These are for use with DJI radio controllers.

Mercury comes with 2 Btx connection wire assemblies: One for use with DJI digital (eg DJI O3), and one that has bare wires, for soldering to pads.

Connecting to an external radio receiver

Mercury is compatible with ELRS transmitters using no external hardware.

If you wish to use an external radio receiver, connect the receiver to Mercury using a groundwire soldered to a ground pad, a power wire connected to a power +5V pad, and a UART Tx/Rx pair. (Only Rx is required for normal operation).

Configuring your radio transmitter

The Betaflight firmware that comes with your flight controller is automatically configured to work with the onboard ELRS receiver. Updating using the Betaflight *Configurator* also automatically configures the receiver, so no special action is required.

This is what you should see in the *Configurator* receiver section:

The image shows a screenshot of the Betaflight Configurator's Receiver section. It features three main panels: 'Receiver', 'Telemetry', and 'RSSI (Signal Strength)'. In the 'Receiver' panel, 'Receiver Mode' is set to 'Serial (via UART)'. A yellow warning box contains the text: 'The UART for the receiver must be set to 'Serial Rx' (in the Ports tab)' and 'Select the correct data format from the drop-down, below:'. Below this, 'Serial Receiver Provider' is set to 'CRSF'. The 'Telemetry' panel shows 'TELEMETRY' output is enabled with a toggle switch. The 'RSSI (Signal Strength)' panel shows 'RSSI_ADC' input is disabled with a toggle switch.

In ArduPilot, this will be configured by default as well. The following settings should be configured using *Mission Planner*, or *QgroundControl*:

```
SERIAL5_PROTOCOL 23 # RCIN
RC_OPTIONS 8192 # Set 420k baud for ELRS
RC2_REVERSED 1 # Prevent inverse pitch controls
RSSI_TYPE 3 # RC provided RSSI
ARMING_RUDDER 0 # Disable rudder arming
```

Using the onboard radio for telemetry

By default, the onboard radio is configured for control data only. It can also be configured for telemetry. For example, this can be used to pass arbitrary data between the aircraft and a ground station, using Mavlink messages. If you wish to only use telemetry on this radio, enable Airport on the radio's WiFi menu (Connect to the ExpressLRS Rx WiFi access point, after leaving the flight controller plugged in for one minute without a bound transmitter; the LED will flash rapidly). Set the Baud rate to 9,600kbps.

If you wish to use the onboard radio for both control data and telemetry, you must use the (In beta) ELRS MavLink functionality. To do so, configure Ardupilot as follows:

```
SERIAL5_PROTOCOL 2 # Mavlink
SERIAL5_BAUD 460
SERIAL5_OPTIONS 0 # Remove ELRS 420k baud
RSSI_TYPE 5 # Mavlink
```

In the GCS, set baud to 460kpbs.

Radio LED status indications

The radio uses a red LED that blinks in different patterns to indicated status. The meanings can be found at the official ELRs documentation here: <https://www.expresslrs.org/3.0/quick-start/led-status/>

Status radio (red) LED:

LED Indication	Status
Slow blink 500ms on/off	Waiting for connection from transmitter
Fast blinking 25ms on/off	WiFi mode enabled
Medium speed blink 100ms on/off	Radio chip not detected
Double blink then pause	Binding mode enabled
Triple blink then pause	Connected to transmitter but mismatched model-match configuration
Solid on	Connected to a transmitter, or bootloader mode enabled

These rates and protocols are selected on the transmitter, in its ELRS settings. (Eg *SYS* button, *Tools* tab, *ExpressLRS* LUA, on Radiomaster controllers.) *Note* that FLRC provides lower latency, while LoRa provides longer range. Déjà Vu sends the same packet multiple times at 1kHz using FLRC, which provides resistance to interference.)

To use F1000, you may need to increase your radio controller’s baud rate from its default. For example, on Radiomaster controllers, this is adjustable by pressing the *SYS* button, then navigating to the *Hardware* tab. Setting the rate above 400kHz may be required.

Binding to a transmitter

Reference the ExpressLRS documentation here for details on binding a transmitter: <https://www.expresslrs.org/3.0/quick-start/binding/>

By default, the receiver is set up in manual bind mode. After power is applied, the status LED will blink orange, indicating it is ready to bind. To bind from this mode, activate the *Bind* feature using your transmitter's ELRS LUA script.

For automatic binding, set up a binding phrase shared by the transmitter and receiver. This can be set, for either, by allowing the transmitter and receiver each to go into WiFi mode. Using a computer or phone with WiFi capability, connect to the *ExpressLRS RX* or *ExpressLRS TX* WiFi networks. The password is *expresslrs*.

Your web browser should load a configuration page automatically upon opening, similar to WiFi login pages at cafes. Once the page loads (10.0.0.1 is the address), set a binding phrase using the web interface on that page. This should be the first field on the page that loads, labeled *Binding Phrase*. Click the *SAVE & REBOOT* button towards the bottom of that page.

Binding Phrase
i<3elrs

UID **Flashed** (Auto updated by changing the bind-phrase above)
211,1,163,201,243,224

WiFi "auto on" interval (s)
60

UART baud
420000

Invert TX pin
 Lock on first connection

SAVE & REBOOT

Note that on the single-radio Rx, you can tell it's in WiFi mode by the LED strobing rapidly. On the dual-radio Rx, the LED will pulse yellow and green to indicate it's in WiFi mode. In either case, this occurs after being powered on for 1 minute without binding to a transmitter.

Make sure the bind phrases set on the receiver and transmitter match. The bind phrase should be kept private.

Using with Betaflight

Mercury supports Betaflight, version 4.4 and higher. It is compatible with quadcopters, but not fixed-wing aircraft. To flash with BetaFlight, the first step is to start your FC in DFU mode. To do this, hold the button on the FC down, while applying, or resetting power. (eg plugging in the USB cable).

- Connect the FC to a PC that has the Betaflight Configurator installed.
- Using the tab at the top, select *Flash firmware*.
- Select *AnyLeaf Mercury H7* flight controller from the list.
- At the bottom of the screen, select *Load online*.
- Select *Flash firmware*. Wait for the flashing to complete.

Note that this configuration automatically configures the ELRS radio, and the digital display OSD on UART2. To connect to an external radio receiver, configure manually using the Ports and Radio tabs.

The default Betaflight config configures Mercury G7 to use an 8kHz PID loop, and DSHOT 600 (600kHz). If your ESC is compatible, we recommend you enable bidirectional DSHOT. The config sets up "Bitbanged" DSHOT, instead of Burst DMA. Burst DMA appears, as of the latest Betaflight, to be incompatible with bidirectional DSHOT.

Here's an example of the *Ports* tab in Betaflight, showing how to configure with an OSD, the onboard ELRS Rx, and GPS. Note that Mercury will set up UART7 (ELRS Rx), and UART2 (OSD) by default as shown. If you use a GPS, make sure to set the *Sensor Input* column to *GPS / AUTO*, and depicted below:

Identifier	Configuration/MSP	Serial Rx	Telemetry Output	Sensor Input	Peripherals
USB VCP	<input checked="" type="checkbox"/> 115200 ▾	<input type="checkbox"/>	Disabled ▾ AUTO ▾	Disabled ▾ AUTO ▾	Disabled ▾ AUTO ▾
UART1	<input type="checkbox"/> 115200 ▾	<input type="checkbox"/>	Disabled ▾ AUTO ▾	Disabled ▾ AUTO ▾	Disabled ▾ AUTO ▾
UART2	<input checked="" type="checkbox"/> 115200 ▾	<input type="checkbox"/>	Disabled ▾ AUTO ▾	Disabled ▾ AUTO ▾	VTX (MSP + D ▾ AUTO ▾
UART3	<input type="checkbox"/> 115200 ▾	<input type="checkbox"/>	Disabled ▾ AUTO ▾	Disabled ▾ AUTO ▾	Disabled ▾ AUTO ▾
UART4	<input type="checkbox"/> 115200 ▾	<input type="checkbox"/>	Disabled ▾ AUTO ▾	Disabled ▾ AUTO ▾	Disabled ▾ AUTO ▾
UART7	<input type="checkbox"/> 115200 ▾	<input checked="" type="checkbox"/>	Disabled ▾ AUTO ▾	Disabled ▾ AUTO ▾	Disabled ▾ AUTO ▾
UART8	<input type="checkbox"/> 115200 ▾	<input type="checkbox"/>	Disabled ▾ AUTO ▾	Disabled ▾ AUTO ▾	Disabled ▾ AUTO ▾

Set the Peripherals column to *VTX(MSP + Displayport)*

The image below is from Betaflight's *Receiver* tab. Note that the protocol is configured to be Serial, using the CRSF provider. Telemetry is enabled, and the RSSI channel is set to AUX 11. AUX 11 corresponds to Link Quality. This is done due to a limitation on some DJI OSDs, where only 1 receiver statistic can be shown. (As configured on the OSD tab as "RSSI value"). If you wish to show RSSI instead, set this to AUX 12.

Receiver

Serial (via UART) ▼

Receiver Mode

- The UART for the receiver must be set to 'Serial Rx' (in the Ports tab)
- Select the correct data format from the drop-down, below:

CRSF ▼

Serial Receiver Provider

Telemetry

TELEMETRY

Telemetry output

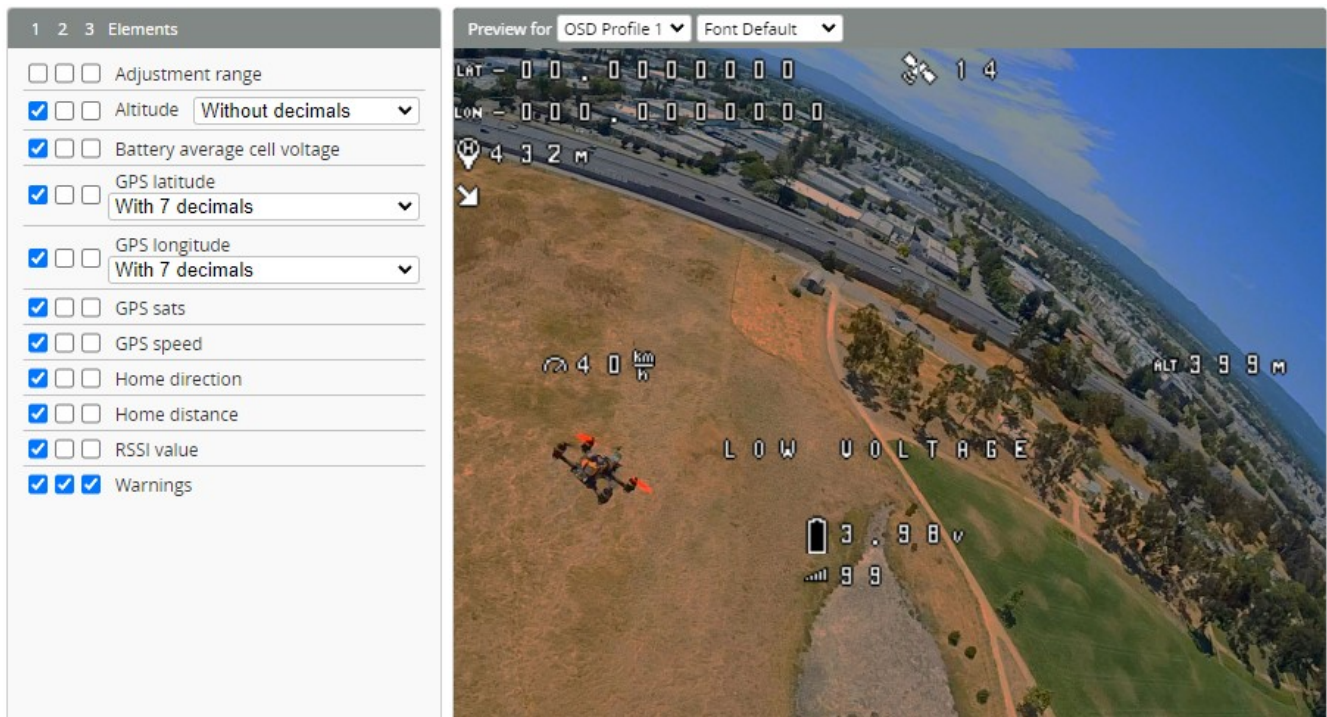
RSSI (Signal Strength)

RSSI_ADC

Analog RSSI input

Channel Map	RSSI Channel
AETR1234 ▼	AUX 11 ▼

An example Betaflight OSD config with DJI Vista or Air Unit is shown below. This is an example of useful information to display given its limited set of OSD features. If using DJI O3 or HDZero, you may have be able to select more options. (Many options you select will simply fail to show on the goggles when using Vista or Air Unit). Unlike the settings above, this *is not* set up by default; you must configure the OSD items manually.



CAN support

This flight controller supports DroneCAN peripherals, with up to 5Mbps connections. Note that Betaflight and iNav do not support CAN; ArduPilot and PX4 do. It includes a 120Ω resistor bridging the CAN high and low lines. It uses a standard DroneCAN 4-pin JST-GH port. Common CAN peripherals included integrated GPS + magnetometers, motors, servos, and sensors.

Power delivery

Mercury H7 has 9V, 5V and 3.3V power lines exposed. The 9V line, and total combined power output is limited to 3A. 5V line supports 2A of current. (This includes connected CAN peripherals) The 5V and 3.3V lines may be used with external hardware, like GPS and time-of-flight sensors. Note that the current capacity on the 5v line isn't high enough for most servos; if you wish to use this FC with more than one servo (eg fixed-wing aircraft), an external regulator (BEC) is required.

The IMU's power supply is isolated from other circuitry on the FC using a power filtering circuit that eliminates high-frequency oscillations.

Pin Mappings

Mercury uses the following pin assignments:

Connection	MCU Port and pin	MCU peripheral	Notes
Motor 1	PC6	Timer 3, Ch1	
Motor 2	PC7	Timer 3, Ch2	
Motor 3	PC8	Timer 3, Ch3	
Motor 4	PC9	Timer 3, Ch4	
Motor 5	PE5	Timer 15, Ch1	
Motor 6	PE6	Timer 15, Ch1	
Motor 7	PE13	Timer 1, Ch3	
Motor 8	PE14	Timer 1, Ch4	
UART1 Tx	PB6		
UART1 Rx	PB7		Also exposed as DJI HDL, and the GPS connector
UART2 Tx	PA2		Used for OSD
UART2 Rx	PA3		Used for OSD
UART3 Tx	PD8		
UART3 Rx	PD9		Also exposed as ESC telemetry
UART4 Tx	PC10		

UART4 Rx	PC11		
UART7 Tx	PB4		Used for the onboard ELRS receiver; no pad
UART7 Rx	PB3		Used for the onboard ELRS receiver; no pad
UART8 Tx	PE1		
UART8 Rx	PE0		
I2C1 SDA	PB9		
I2C1 SCL	PA15		
I2C2 SDA	PA8		Used by baro-altimeter; no pad
I2C2 SCL	PA9		Used by baro-altimeter; no pad
Buzzer	PE9		This is a PWM signal alternating between ground and 3.3V
Batt voltage ADC	PA4	ADC1 or 2, Ch18	From a voltage divider, with scaling factor 11
Current ADC	PB2	ADC1, Ch16	
CAN1 Tx	PD1		Connected to a transceiver
CAN1 Rx	PD0		Connected to a transceiver

The Vtx/cam pad and header pin labeled "HDL" is the same connection as the UART 1 Rx pad. The ESC telemetry pad and header pin is the same connection as the UART 3 Rx pad. If using the HDL connection, don't use UART1 Rx for anything else; if using ESC telemetry connection, don't use UART 3 Rx for anything else.

Note that there are no exposed SPI pads; both SPI lines are used by FC peripherals (IMU and flash storage).

Support

If you have any questions, or support requests, contact us by email: anyleaf@anyleaf.org.

© 2024 AnyLeaf LLC