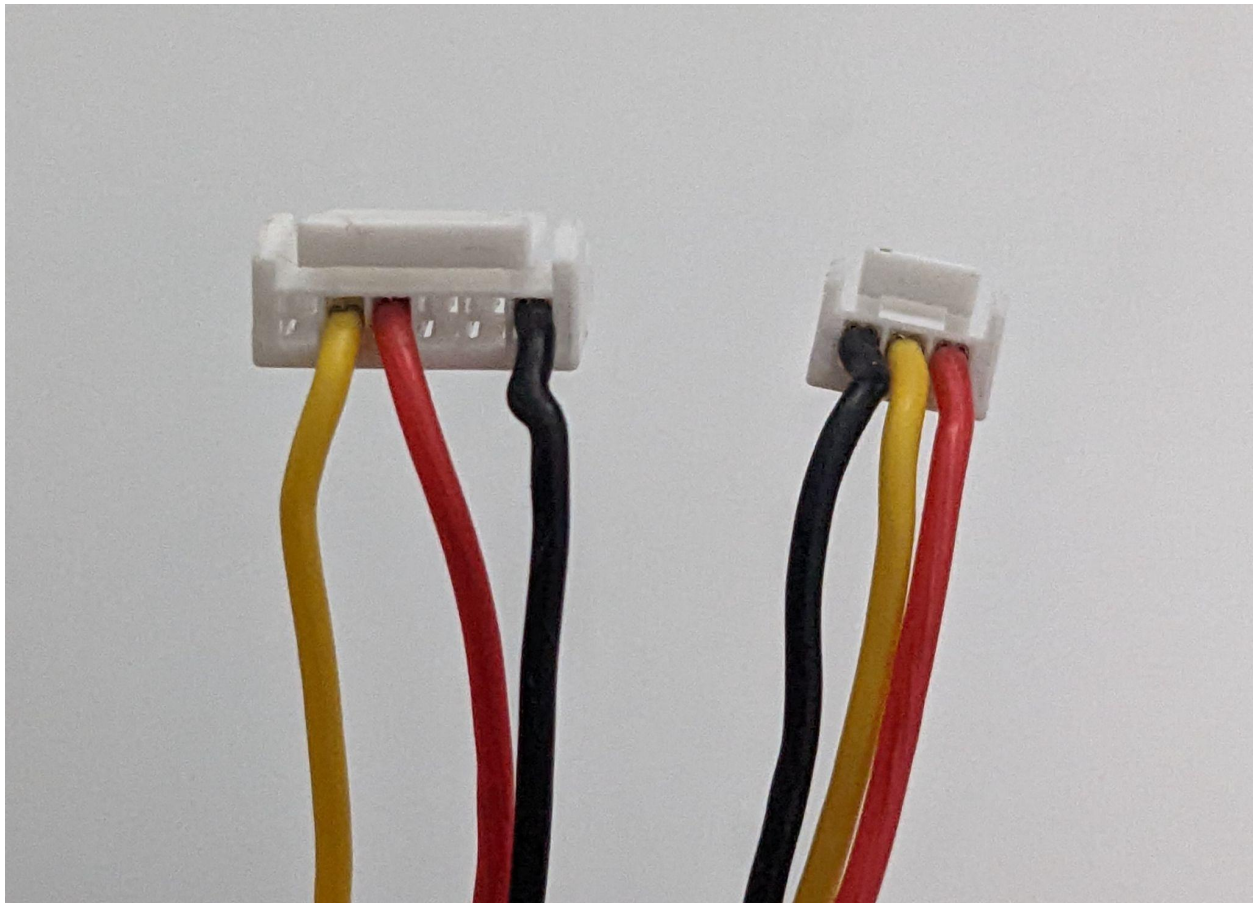


RDS2 Integration Quickstart

Physical Connection

Connect the 3-pin "Drone" port on the winch board to a spare telemetry port (telem1 or telem2) on your flight controller. If your unit did not ship with a cord, see the photo below for how to make it:



Flight Controller Parameters:

Configure the port you used as a MAVLink2 port, and ensure the baudrate matches the winch (which by default uses 9600). Parameters can be changed by connecting the vehicle to QGC or Mission Planner.

Ardupilot

[See here for details](#), but you should just need to change two parameters. For example, if using the telem1 port at baud 9600:

- SERIAL1_PROTOCOL to MAVLink2 (int value 2)
- SERIAL1_BAUD to 9600 (int value 9).

PX4

[See here for details](#). The interface should be configured to enable forwarding, with MODE to CUSTOM. As an example, to use mavlink interface #2 with the TELEM2 port at baud 9600:

- Set MAV_2_CONFIG to TELEM2
- Reboot the drone (so the additional configuration parameters appear)
- Set MAV_2_FORWARD to 1 (to enable forwarding between groundstation/pc and RDS2)
- Set MAV_2_MODE to CUSTOM (to ensure no MAVLink messages are sent by the FC)
- Set SER_TELEM2_BAUD to 9600 (to match the baudrate set in the RDS2 parameters)

MAVLink Interface

**If you are using A2ZQGC, all of the below is handled by the software. This section is for users implementing their own control on a ground station or companion computer.

Once the RDS2 is connected and parameters configured, your GCS should start getting WINCH_STATUS messages and HEARTBEAT messages from, by default, System 1 (usually the drone), Component 7 (to identify the winch). If you send MAV_CMD_DO_WINCH messages to this system/component (this is important, check those IDs!), you can control the RDS2.

Parsing RDS2 Status

The message streamed from the RDS2 is of type [WINCH_STATUS](#). The specific interpretation of the fields differs slightly from the MAVLink spec. Additionally, a single high-level RDS2 state indicator is packed into the status field. See the table below:

Field Name	Type	Units	Standard Description	RDS2 notes
time_usec	uint64_t	us	Timestamp (syncd to UNIX time or since system boot).	As described
line_length	float	m	Length of line released. NaN if unknown	As described
speed	float	m/s	Speed line is being released or retracted. Positive values if being released, negative values if being retracted, NaN if unknown	As described, but may have a scale factor (ie not directly in m/s)

tension	float	kg	Tension on the line. NaN if unknown	Tension is not directly measured. This field reports the coil current of the drive motor.
voltage	float	V	Voltage of the battery supplying the winch. NaN if unknown	As described
current	float	A	Current draw from the winch. NaN if unknown	This reflects the current draw of the control board only, not including winch motor.
temperature	int16_t	degC	Temperature of the motor. INT16_MAX if unknown	This reflects the temperature of the clutch servomotor
status	uint32_t		Status flags as encoded in WINCH_STATUS_FLAGS	Not all fields are used. Additionally a high-level status enum is packed into the last 8 bits of this field. See table below.

The 'status' field does contain most of the flags in the spec, but we suggest only using the RDS2_STATUS enum packed into the last 8 bits of the field for your primary status display as they are more useful for understanding the state of the system. The code below will extract this state:

```
mavlink_msg_winch_status_decode(&message, &winch_status);

int rds2_state = winch_status.status >> 24;
```

This table shows the interpretation of this state enum:

Int Value	State Name	Description
0	ATTACHED	Package or hook is secured at the top with no line out.
1	IN_AIR	Package or hook is hanging, spool locked, in the air with some line out.
2	ON_GROUND	Package or hook is on the ground, spool locked, with line out.
3	FREEFALL	UNUSED FOR RDS2
4	BRAKING	UNUSED FOR RDS2
5	REELING_DOWN	Payload is being actively lowered.
6	REELING_UP	Payload is being actively hauled upward.
7	FREEWHEEL	The spool is unlocked and unpowered, for loading or emergency release.
8	UNUSED	UNUSED
9	DEV	Unit is reading/writing logs to SD

In addition, the RDS2 will stream [DISTANCE_SENSOR](#) measurements from its downward-facing LIDAR unit.

Commanding the RDS2

Since the RDS2 handles low-level functions independently, it does not use most of the parameters in the MAV_CMD_DO_WINCH. The only parameter to specify is the parameter 2, "Action".

Param (:Label)	Description
1: Instance	<i>Unused (component ID is used to differentiate multiple systems)</i>
2: Action	<i>Action to perform. **The RDS2 does NOT use the stock WINCH_ACTION enum, see the RDS2_ACTION below.</i>
3: Length	<i>Unused (winch automatically detects ground)</i>
4: Rate	<i>Unused (winch accelerates and decelerates automatically)</i>
5	Empty.
6	Empty.
7	Empty.

RDS2_ACTION:

Value	Command name	Description	Allowable States
0	UNUSED	N/A	
1	DELIVER	Convenience function: Performs a reeldown to the ground, followed automatically by a reelup. Should only be allowed when state is ATTACHED.	ATTACHED
2	LOCK	Locks the spool, either to stop a REELUP/REELDOWN/DELIVER , or to regain hold of the package after a FREEWHEEL.	REELING_DOWN REELING_UP FREEWHEEL
3	FREEWHEEL	Disengages the clutch, allowing the spool to spin freely. Used in an emergency to release the tether and fly away.	ATTACHED IN_AIR ON_GROUND
4	REELUP	Reels the package or hook up until either it reaches the top or is manually stopped with a LOCK.	IN_AIR ON_GROUND
5	REELDOWN	Reels the package or hook down until either it reaches the ground or is manually stopped with a LOCK.	ATTACHED IN_AIR ON_GROUND

User Interface Suggestions

To implement a control panel for the RDS2, we suggest a panel of buttons, one for each required RDS2_ACTION, and a readout panel to display information from the WINCH_STATUS and DISTANCE_SENSOR messages.

For the main flight view, we suggest including RDS2_STATE, line_length, speed, and current_distance (from DISTANCE_SENSOR). The other fields may be placed in a separate detail menu.

Additionally, we suggest only enabling each button if the RDS2 is in an allowable RDS2_STATE per the RDS2_ACTION table, showing the button as disabled otherwise.