REVISION	CHANGE	APPROVED	DATE
1	Published release	JTS	18/11/2025

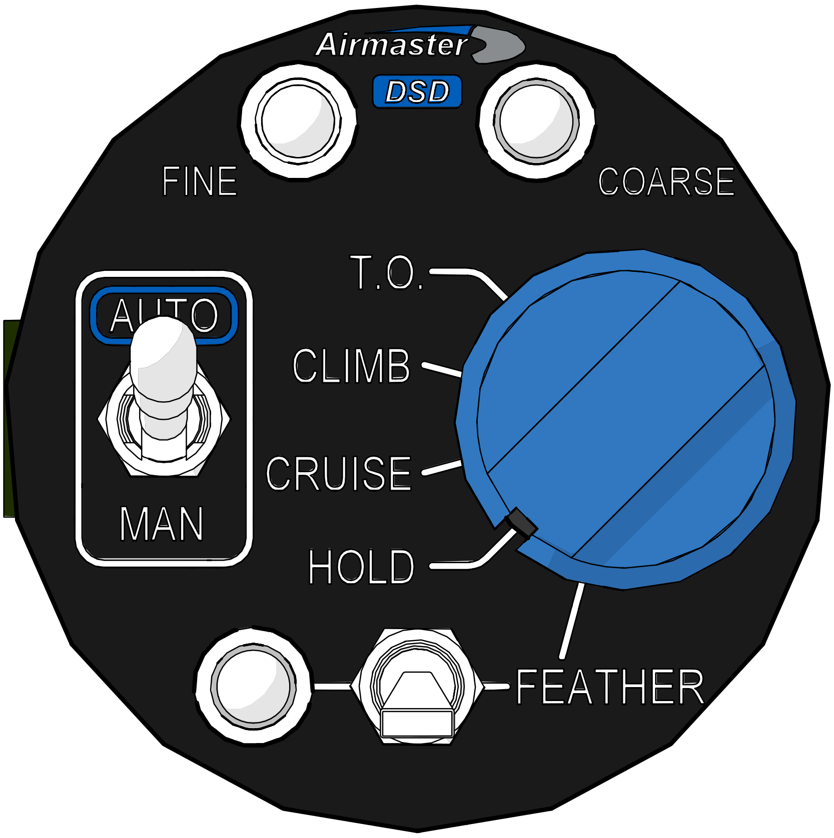
Airmaster Propellers Ltd
20 Haszard Rd, Massey,
Auckland 0614, NZ

Phone: (+64) 9 833 1794
E-mail: support@propellor.com
Web: www.propellor.com

ASI-6-1-3

PROPELLER CAN INTERFACE CONTROL

GUIDE



SUBJECT:	
Operation of Controller	
ASSEMBLY NO:	**APPLICABILITY:**
AC-300x(CAN)	DSD propeller models only (AC300 controllers only)

0. TABLE OF CONTENTS

0.1 Abbreviations

ABBREVIATION	MEANING
CAN	Controller Area Network
DSD	Digital Servo Drive
DLC	Data Length Code
BLDC	Brushless DC

1. TOPIC

1.1 Introduction

This document describes the remote CAN interface capabilities for the Airmaster AC300 control system. The following information is covered:

- Physical requirements and description of the AC300 controller and connectors.
- Electrical requirements for the AC300 control system and CAN interface.
- Details on CAN data layer implementation.
- Details on Airmaster's remote command interface.

This document is applicable to customers who have an AC300 control system, where the controller has a CAN transceiver fitted.

1.2 Control System Architecture

The AC300 control system consists of two hardware units:

- The Airmaster AC300 Controller.
- The Airmaster DSD Hub.

The **AC300 controller** is the main interface to external systems. All CAN commands from external controllers are sent to the AC300 controller. The AC300 controller then commands the hub to adjust propeller pitch based on the received commands. Auxiliary functions, including logging flight information and system status monitoring are also performed by the AC300 controller.

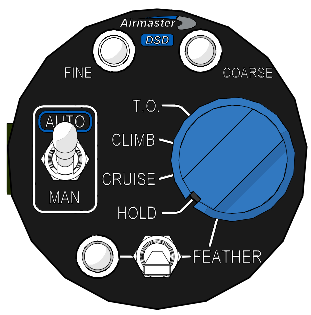
The **DSD hub** actuates the BLDC pitch change motor and monitors blade angle position through a relative encoder system. The DSD hub also ensures that the system stays between the fine and coarse pitch limit stops (normal flight pitch range) and checks hub temperature.

External controllers, such as UAV flight control systems or fly-by-wire controllers, can connect to the AC300 controller through the provided CAN interface. The AC300 controller then communicates to the DSD Hub through the sensor-brush block interface.

1.3 Usage Instructions

To use the Airmaster AC300 controller in remote mode:

- Set controller to **AUTO/HOLD**.



2. System Specifications

2.1 Physical Specifications

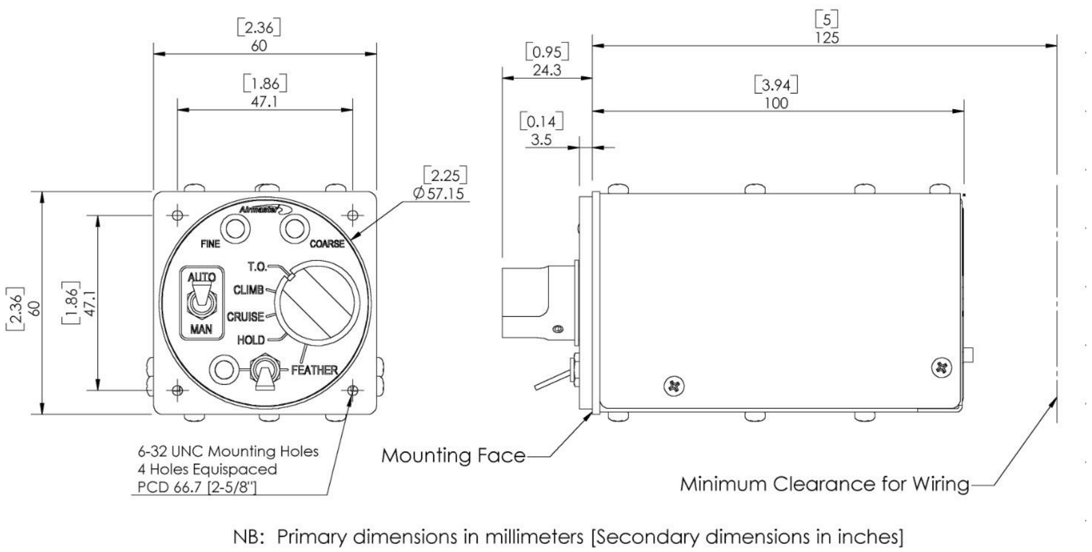
2.1.1 Power Supply

- Stable 12 / 24V input recommended (depending on pitch change motor specification).
- 30V input maximum.
- 5A current source capability required.
- 6A fuse recommended.

2.1.2 Control Circuit Diagram

- DSD control system circuit diagram (PDF) available upon request from Airmaster.

2.1.3 AC300 Controller Dimensions



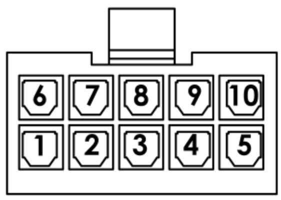
2.2 CAN Physical Layer

2.2.1 Connector

- 10-way latched connector (Molex mini fit pins). Molex P/N: 39012105.
- Connects to controller port [CN5].

> ⓘ *Note*
> *Airmaster can supply the required loom with one end wired to connect to controller port CN5.*

- Pin Layout:
  - Pin 2: CANH (White).
  - Pin 9: CANL (White / Orange).



- Pin 5/10: GND (White / Blue).

> ⓘ *Note*
> *The pin map shown above is oriented with the female pin end coming out of the page.*

2.2.2 CAN Electrical Specifications

- +/- 14V maximum allowable input to CANH and CANL lines.
- 120 Ω termination optional.

2.3 CAN Data Layer

2.3.1 Interface Specifications

- CAN 2.0A Standard frame format.
- No CAN remote frames.
- 125 / 250 / 500 / 1000 kbits/s speed.
- Maximum of four Airmaster controllers can be connected on the same bus.

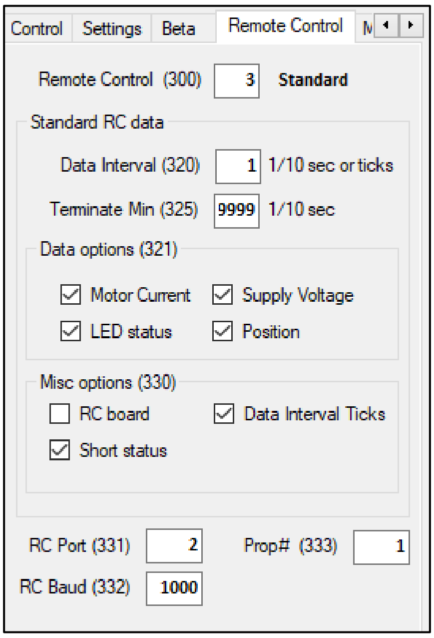
2.3.2 CAN Configuration

The enabling of the AC300 remote control CAN system, along with all other settings and parameters programmed into the controller, is configured using Airmaster's proprietary software: **AC300User.exe** (Airmaster User Program).

This software is run on a Windows PC. Connection to the controller is through the proprietary Airmaster RS232 USB-serial cable (P/N: A0117).

An example of the AC300 controller's remote-control parameters is shown opposite (as displayed in the Airmaster user program). A description of these parameters is provided in the next section.

> ⓘ *Note*
> *For more information on adjusting controller parameters, refer to procedure **ASI-7-2-1**.*



2.3.3 Remote Control Parameters

PAR NAME	PAR NO.	VALUE	DESCRIPTION
Remote Control	(300)	0	Remote control off.
		3	Standard remote control.
		5	Serial to CAN converter.
Data Interval	(320)	0	Do not send any data line updates.
		1 – 100	Send a data line with the minimum period (DATA_INTERVAL_VALUE) * 100ms
		1	Send a data line with the minimum period (DATA_INTERVAL_VALUE) * 20ms. This only applies when “Data Interval Ticks” is selected under Misc Options (330).
Data Options	(321)	0 – 15	Controls what data is sent during a data update. This value is represented as a bitwise flag, where: 0b000 (i.e. ‘1’): Represents motor current only 0b0011 (i.e. ‘3’): Represents motor current & supply voltage, and so on.
		1 (LSB)	Motor current [mA]
		1	Supply Voltage [V] (as read by the controller). (Hardware version AC300_BL1 of control board reads max 12V, even if 24V supply).
		1	LED status (see section 4.5.3).
		1 (MSB)	Position: the current reported angle of DSD system.
Terminate Min	(325)	0 – 9999	Blocks command termination messages (RS_T) unless the command timeout is greater than (TERMINATE_MIN_VALUE) * 100ms.
Misc Options	(330)	Miscellaneous options for CAN control. This value is represented as a bitwise flag.	
		1 (LSB)	*RC Boards:* 1: If a remote-control board is being used. 0: If a standard controller is being used
		1	*Data Intervals Ticks:* 1: Set data line min. update period [units of “ticks” / 20ms]. 0: Set data line min. update period [units of 100ms]. See description for “Data Interval” Par (320).
		1 (MSB)	*Short Status:* 1: Remove duplicate status data (only relevant when “Data Interval Ticks” = 1). 0: Preserve default message format.
RC Port	(331)	0	No auxiliary port is used.
		2	Auxiliary CAN port is used.
RC Baud	(332)	125, 250, 500, 1000	Value of baud rate if using auxiliary CAN port (RC Port = 2). To get actual baud rate, multiply value by 1000 (e.g. Par 332 = 125 means baud rate = 125000, or 125*1000).

Airmaster Propellers Ltd
20 Haszard Rd, Massey,
Auckland 0614, NZ
Phone: (+64) 9 833 1794
E-mail: support@propellor.com
Web: www.propellor.com

3. CAN TX FORMAT

3.1 Overview

The following section describes the data transmitted from the AC300 controller (TX interface). This includes command format and function.

The AC300 controller TX interface can read and respond to a new command every 20 milliseconds. However, if a command is currently executing, the AC300 controller will not respond to the new command until the previous command is completed.

The AC300 controller CAN TX interface gives full control of the propeller to the primary flight controller.

3.2 CAN TX Command Quick Reference Table

CAN ID	COMMAND NAME	DLC	DIRECTION	SUMMARY
501	Drive Fine	3	Tx	Drives propeller towards fine stop.
502	Drive Coarse	3	Tx	Drives propeller towards coarse stop.
503	Drive In Reverse	3	Tx	Drives propeller towards reverse stop.
504	Drive Out Reverse	3	Tx	Drives propeller out of reverse towards fine stop.
505	Stop Command	1	Tx	Stops any currently running command.
510	Set RPM	3	Tx	Tells propeller to govern to specific RPM.
520	Set Position	3	Tx	Drives propeller to position in normal zone.
521	Set Reverse Position	3	Tx	Drives propeller to position in reverse zone.
530	Feather	2	Tx	Drives propeller to feather position stop.
531	Unfeather	2	Tx	Drives propeller out of feather position stop.
540	Request Diagnostic	1	Tx	Requests diagnostic packet from propeller.
541	Request Status Data	2	Tx	Requests status data packet from propeller.
550	Reset Over-Current	2	Tx	Allows control after over-current alarm trigger.

3.3 CAN Command ID 501- Drive Fine

3.3.1 Summary

Drives propeller ***Prop#*** in the fine pitch direction at speed ***Speed*** until the appointed time ***Timeout*** has elapsed, or the fine stop is reached.

3.3.2 Data Fields

Byte#	Field Name	Description	Values
0	Prop#	ID of propeller to drive fine.	1 - 4
1	Timeout	Time to drive fine, specified in 100ms increments.	1 - 255
2	Speed	PC motor speed, expressed in 1/10 increments of max motor RPM.	1 - 10

3.3.3 Example

- Drive propeller 1 fine at speed 5 for 500ms:

ID	DLC	D0	D1	D2
501	3	1	5	5

- Drive propeller 3 fine at speed 10 until fine stop is reached:

ID	DLC	D0	D1	D2
501	3	3	255	10

3.4 CAN Command ID 502 – Drive Coarse

3.4.1 Summary

Drives propeller ***Prop#*** in the coarse pitch direction at speed ***Speed*** until the appointed time ***Timeout*** has elapsed, or the coarse stop is reached.

3.4.2 Data Fields

Byte#	Field Name	Description	Values
0	Prop#	ID of propeller to drive coarse.	1 - 4
1	Timeout	Time to drive coarse, specified in 100ms increments.	1 - 255
2	Speed	PC motor speed, expressed in 1/10 increments of max motor RPM.	1 - 10

3.4.3 Example

- Drive propeller 1 coarse at speed 5 for 500ms:

ID	DLC	D0	D1	D2
502	3	1	5	5

- Drive propeller 3 coarse at speed 10 until fine stop is reached:

ID	DLC	D0	D1	D2
502	3	3	255	10

3.5 CAN Command ID 503 – Drive in Reverse

3.5.1 Summary

Drives propeller ***Prop#*** into the reverse pitch zone at speed ***Speed*** until the appointed time ***Timeout*** has elapsed, or the reverse pitch stop is reached.

3.5.2 Data Fields

Byte#	Field Name	Description	Values
0	Prop#	ID of propeller to drive into reverse.	1 - 4
1	Timeout	Time to drive into reverse pitch, specified in 100ms increments.	1 - 255
2	Speed	PC motor speed, expressed in 1/10 increments of max motor RPM.	1 - 10

3.5.3 Example

- Drive propeller 1 into reverse pitch at speed 5 for 500ms:

ID	DLC	D0	D1	D2
503	3	1	5	5

- Drive propeller 3 into reverse pitch at speed 10 until reverse stop is reached:

ID	DLC	D0	D1	D2
503	3	3	255	10

3.6 CAN Command ID 504 – Drive out of Reverse

3.6.1 Summary

Drives propeller ***Prop#*** out of the reverse pitch zone at speed ***Speed*** until the appointed time ***Timeout*** has elapsed, or the fine stop is reached.

3.6.2 Data Fields

Byte#	Field Name	Description	Values
0	Prop#	ID of propeller to drive out of reverse.	1 - 4
1	Timeout	Time to drive out of reverse for, specified in 100ms increments.	1 - 255
2	Speed	PC motor speed, expressed in 1/10 increments of max motor RPM.	1 - 10

3.6.3 Example

- Drive propeller 1 out of reverse at speed 5 for 500ms:

ID	DLC	D0	D1	D2
504	3	1	5	5

3.7 CAN Command ID 505 – Stop Command

3.7.1 Summary

Terminates the current command running on propeller ***Prop#.***

3.7.2 Data Fields

Byte#	Field Name	Description	Values
0	Prop#	ID of propeller to terminate current running command.	1 - 4

3.7.3 Example

- Terminates the current command on propeller 1.

ID	DLC	D0
505	1	1

3.8 CAN Command ID 510 – Set RPM

3.8.1 Summary

Commands propeller ***Prop#*** to govern propeller pitch to maintain a specified target speed ***RPM***.

3.8.2 Data Fields

Byte#	Field Name	Description	Values
0	Prop#	ID of propeller to drive out of reverse.	1 - 4
1 - 2	RPM	Target RPM for the prop to govern to.	

3.8.3 Example

- Command propeller 1 to govern to 5000 RPM:

ID	DLC	D0	D1	D2
510	3	1	19	136

RPM = (19 << 8) + 136 = 5000 RPM.

3.9 CAN Command ID 520 – Set Position

3.9.1 Summary

Commands propeller ***Prop#*** to move to a specified blade pitch ***Angle*** that is within the normal flight pitch zone.

3.9.2 Data Fields

Byte#	Field Name	Description	Values
0	Prop#	ID of propeller to drive out of reverse.	1 - 4
1 - 2	Angle	Blade angle for the propeller to move to (one implied decimal place).	

3.9.3 Example

- Command propeller 1 to move to a blade pitch of 20.5°:

ID	DLC	D0	D1	D2
520	3	1	0	205

Angle = 0 + 205 = 20.5° (1dp).

3.10 CAN Command ID 521 – Set Reverse Position

3.10.1 Summary

- Commands propeller ***Prop#*** to move to a specified blade pitch ***Angle*** that is within the reverse pitch zone.

3.10.2 Data Fields

Byte#	Field Name	Description	Values
0	Prop#	ID of propeller to drive out of reverse pitch zone.	1 - 4
1-2	Angle	Blade angle for the propeller to move to (one implied decimal place). Negative values are represented in 16-bit 2's complement.	

3.10.3 Example

- Command propeller 1 to move to a blade pitch of -10.0°:

ID	DLC	D0	D1	D2
521	3	1	255	156

Angle = (255 << 8) + 156 = -10.0° (16-bit 2's complement. 1dp).

3.11 CAN Command ID 530 – Feather

3.11.1 Summary

- Drives propeller ***Prop#*** to the feather pitch position.

3.11.2 Data Fields

Byte#	Field Name	Description	Values
0	Prop#	ID of propeller to drive into feather pitch position.	1 - 4

3.11.3 Example

- Drive propeller 1 into the feather pitch position:

ID	DLC	D0
530	1	1

3.12 CAN Command ID 531 – Unfeather

3.12.1 Summary

Drives propeller ***Prop#*** out of the feather pitch position, returning to normal flight pitch zone.

3.12.2 Data Fields

Byte#	Field Name	Description	Values
0	Prop#	ID of propeller to drive out of the feather pitch position.	1 - 4

3.12.3 Example

- Drive propeller 1 out of the feather pitch position:

ID	DLC	D0
531	1	1

3.13 CAN Command ID 540 – Request Diagnostic Packet

3.13.1 Summary

Requests diagnostic packet (CAN ID 71X) from propeller ***Prop#*** immediately.

3.13.2 Data Fields

Byte#	Field Name	Description	Values
0	Prop#	ID of propeller to receive status data from.	1 - 4
1	Send Data	Boolean value representing whether to continue sending status data.	0, 1

3.13.3 Example

- Request diagnostic packet from propeller 1:

ID	DLC	D0
540	1	1

3.14 CAN Command ID 541 – Request status data

3.14.1 Summary

Requests system response packet ***CAN ID 70X*** from propeller ***Prop#*** immediately.

- If ***Send Data*** = 0: propeller ***Prop#*** will stop sending periodic ('D'-type) system response packets.
- If ***Send Data*** = 1: propeller ***Prop#*** will start/continue sending periodic system response packets.

3.14.2 Data Fields

Byte#	Field Name	Description	Values
0	Prop#	ID of propeller to receive status data from.	1 - 4
1	Send Data	Boolean value representing whether to continue sending status data.	0, 1

3.14.3 Example

- Request system response packet from propeller 1 but tell it to stop sending further periodic system response packets:

ID	DLC	D0	D1
541	2	1	0

3.15 CAN Command ID 550 – Reset Overcurrent

3.15.1 Summary

Resets overcurrent warning on propeller ***Prop#***.

3.15.2 Data Fields

Byte#	Field Name	Description	Values
0	Prop#	ID of propeller to reset over-current alarm for.	1 - 4
1	Reset Val	Must be set to 0.	0

3.15.3 Example

- Reset over-current alarm on propeller 1.

ID	DLC	D0	D1
550	2	1	0

Airmaster Propellers Ltd
20 Haszard Rd, Massey,
Auckland 0614, NZ
Phone: (+64) 9 833 1794
E-mail: support@propellor.com
Web: www.propellor.com

4. CAN RX FORMAT

4.1 Overview

The following section describes the AC300 controller RX interface.

The AC300 controller can send system information back to the main controller under three circumstances:

- When the AC300 controller receives a command.
- When the AC300 controller successfully executes a command.
- At a user-configurable interval.

The information sent in the response packets for users to receive real-time feedback on the current propeller status.

4.2 CAN RX Response Quick Reference Table

CAN ID	Command Name	DLC	Direction	Summary
70X	System Response Packet	8	Rx	Sends system health information (e.g. current mode of operation, status flags, current RPM).
71X	Diagnostic Packet	8	Rx	Sends diagnostic information such as firmware versions and parameter settings.
72X	Data Information Packet	8	Rx	Sends information as specified by data options parameter (parameter no. 321).

4.3 CAN ID 70X – System Response Packet

4.3.1 Summary

This packet sends system health information to the primary controller.

4.3.2 Data Fields

BYTE#	FIELD NAME	DESCRIPTION	VALUES
0	Packet Type	*Packet type in ASCII format:* 0d83/S: Status packet. 0d84/T: Termination packet. 0d68/D: Data packet.	83, 84, 68.
1	Error	Field indicating configuration or command errors.	1 - 7
2	Mode	*Current mode of operation in ASCII format:* 0d83/S: Set RPM mode. 0d68/D: Direct command mode. 0d80/P: Positioning mode. 0d87/W: Waiting command. 0d73/I: Idle.	83, 68, 80, 87, 73.
3	Status Flags	Bitwise field indicating controller status flags.	

4	RPM/Angle High	The 8 MSB of the target propeller RPM in set RPM mode, otherwise the 8 MSB of the current blade pitch.
5	RPM/Angle Low	The 8 LSB of the target propeller RPM in set RPM mode, otherwise the 8 LSB of the current blade pitch.
6	Current RPM High	The 8 MSB of the current propeller RPM.
7	Current RPM Low	The 8 LSB of the current propeller RPM.

4.3.3 Behaviour

Packet Types

- **'S'/Status** packets are to acknowledge a received command.
- **'T'/Termination** packets are used when command execution has finished.
- **'D'/Data** packets are sent at an interval specified by the user in the controller parameters.

Error

Error Number	Description
1	Bad command.
2	Invalid value for command.
3	Invalid mode. Controller not in AUTO->REMOTE mode.
4	Reversing command but not reversing hub.
5	Not in reverse but reverse only command.
6	Second numeric value does not match first.
7	RPM greater than Beta->Max Engage RPM.

Mode

- **'S'** for set RPM mode. This is when the controller is governing the blade pitch to match a set RPM.
- **'D'** for direct command mode. This is when the controller is executing a direct command.
- **'P'** for positioning mode.
- **'W'** for waiting command mode.
- **'I'** for idle mode.

Status Flags

Bit	Name	Description
0	Fine Stop	High while fine stop is active.
1	Coarse Stop	High while coarse stop is active.
2	Feather / Reverse Stop	High when feather/reverse stop is active.
3	Reverse Stop	High when reverse stop is active (in feather and reverse system).
4	Open Circuit Error	An open circuit error has occurred.
5	Over Current Error	An overcurrent error has occurred.

6	Slipring Error	A slipring voltage error has occurred.
7	Calibration Error	The system has lost calibration.

Set RPM/Angle
If mode = S (set RPM mode), this field shows the target RPM. Otherwise, it shows the current blade pitch.

Current RPM
This field shows the current propeller RPM.

4.3.4 Example

- Receive a packet from Prop# 1, which is waiting for a command at the coarse stop, with a current blade pitch of 35.5°:

ID	DLC	D0	D1	D2	D3	D4	D5	D6	D7
701	8	68	0	87	0	35	5	0	0

  - **ID:** 70**1**, where the 1 indicates Prop# 1.
  - **D0:** 0d68, ASCII encoding for 'D' which is a data packet.
  - **D1:** 0, indicating no errors.
  - **D2:** 0d87, ASCII encoding for 'W' which is wait command mode.
  - **D3:** Bit 1 is set, indicating coarse stop is active.
  - **D4:** Whole part of blade pitch (35 degrees).
  - **D5:** Fractional part of bade pitch (0.5 degrees).
  - **D6 - D7:** No current RPM.

- Receive a packet from Prop# 2, which is governing to a target RPM of 5000 from a current RPM of 5400:

ID	DLC	D0	D1	D2	D3	D4	D5	D6	D7
702	8	83	0	83	0	19	136	21	24

  - **ID:** 70**2**, where the 2 indicates Prop# 2.
  - **D0:** 0d83, ASCII encoding for 'S' which is a status packet.
  - **D1:** 0, indicating no errors.
  - **D2:** 0d83, ASCII encoding for 'S' which is set RPM mode.
  - **D3:** No bits are set, indicating no stops are active.
  - **D4 - D5:** Target RPM is **(**19 << 8) + 136 = 5000 RPM.
  - **D6 - D7:** Current RPM is (21 << 8) + 24 = 5400 RPM.

- Receive a short status packet from Prop#2:

ID	DLC	D0
702	1	83

- **ID:** 70**2**, where the 2 indicates Prop# 2.
- **D0:** 0d83, ASCII encoding for ‘S’ which is a data packet.

4.4 CAN ID 71X – Diagnostic Packet

4.4.1 Summary

This packet type is used to send additional controller data to the customer when a CAN ID 70X data packet is sent.

4.4.2 Data Fields

Byte#	Field Name	Description
0 - 1	AC300 Firmware Version	The firmware version of the AC300 controller.
2 - 3	AC300 Ctrl Word	The control word of the AC300 controller.
4 - 5	DSD Hub Firmware Version	The firmware version of the DSD Hub.
6	Reserved	N/A
7	Remote Fields	Remote control data fields requested.

4.4.3 Behaviour

AC300 Firmware Version

- Example: Version 10.120:
  D0=39, D1=136. SW = (39 << 8) + 136 = 10120

AC300 Control Word

Parameter Name	Parameter Number	Value	Description
Remote Control	300	0	Remote control Off.
		3	Standard remote control.
		5	Serial to CAN converter.

DSD Hub Firmware Version

- Example: Version 1.35.
  D0 = 0
  D1 = 135
  Version = 0 + 135 = 1.35 (one implied decimal place).

Remote Fields

The remote fields requested by the ‘data options’ parameter (parameter number 321).

4.5 CAN ID 72X – Data Information Packet

4.5.1 Summary

The AC300 controller uses the ***CAN ID 72X*** packet type to send additional AC300 controller data to the primary controller. It always follows the ***CAN ID 70X*** data packet.

> ⓘ *Note The AC300 parameter settings (configured in the AC300User Program) control the data fields sent.*

4.5.2 Data Fields

Byte#	Field Name	Description
0 - 1	Motor Current	Total current drawn by the BLDC motor.
2 - 3	Voltage	Voltage level at the controller input.
4	Reserved.	
5	LED Status	The system UI output.
6 - 7	Angle	The current blade angle of the system.

4.5.3 Behaviour

Motor Current
The current [mA] drawn by the drive system including BLDC motor and A0130 hub.

Voltage
The voltage level [V] at the input to the control system (1 implied decimal place).

> ⓘ *Note*
> *For power board revisions up to and including BL12, the voltage monitoring circuit caps the reported voltage at ~12.6V.*

LED Status
Indicates current output of the system LED UI.

Status	Description
0	Idle
1	Pitch decreasing
2	Pitch increasing
3	Pitch increasing in feather
4	No speed signal
5	Fine pitch limit
6	Coarse pitch limit
7	Feather pitch limit
8	Driving at fine pitch limit
9	Driving at coarse pitch limit
10	Driving at feather pitch limit
11	Beta mode engagement (step 1)
12	Fine pitch limit override (Beta option)
13	Coarse pitch limit override (Feather option)
14	Over-current while pitch decreasing

15	Over-current while pitch increasing
16	Over-current while pitch increasing in feather
17	Open circuit failure
18	Controller software fault
19	Pitch decreasing in reverse
20	In reverse zone
21	Driving at reverse pitch limit
22	Reverse pitch limit
23	Slipring error while pitch increasing.
24	Slipring error while pitch decreasing.

Angle
Current blade angle of the system. Represented in 16-bit 2's complement (1 implied decimal place).

4.5.4 Example

- Receive a packet from Prop# 1, which is governing to a target RPM of 5000 from a current RPM of 5400:

ID	DLC	D0	D1	D2	D3	D4	D5	D6	D7
721	8	4	26	12	2	0	2	18	4

  - **ID:** 72**1**, where the 1 indicates Prop# 1.
  - **D0-D1:** Current is (4 << 8) + 26 = 1050mA.
  - **D2-D3:** 12.2V, where D2 (12) is the whole part and D3 (2) is the decimal part.
  - **D4:** Reserved.
  - **D5:** 2, for pitch increasing (see table 4.5.3).
  - **D6 - D7:** 18.4, where D6(18) is the whole part and D7(4) is the decimal part.