AIRENCE

USB HID CONTROL PROTOCOL

12-4-2013 Revision v0.5

Contents

Change log

V0.1:
- initial release.

V0.2:
- AIRENCE_SWITCH_CHANGE_EVENT augmented with USB Channel controls (faderstart, ON, CUE switches).

V0.3:
- AIRENCE_LED_ON_WRITE renamed to AIRENCE_LED_WRITE.
- AIRENCE_LED_OFF_WRITE command is obsolete, instead use AIRENCE_LED_WRITE.
- AIRENCE_LED_ON_EVENT renamed to AIRENCE_LED_EVENT
- AIRENCE_LED_OFF_EVENT command is obsolete, instead AIRENCE_LED_EVENT is used.
- commandID 0x02 AIRENCE_LED_ON renamed to AIRENCE_LED
- commandID 0x04 AIRENCE_LED_OFF is obsolete.
- wrap around functionality of encoder value added.

V0.4:
- non-stop switch added to AIRENCE_SWITCH_CHANGE_EVENT

V0.5:
- bug fix: AIRENCE_SWITCH_CHANGE_EVENT *size* field corrected from 6 to 8.
- AIRENCE_SWITCH_CHANGE_REQUEST added
- AIRENCE_SWITCH_CHANGE_RESPONSE added
- commandID 0x04, AIRENCE_LED_ALL added
- AIRENCE_LED_ALL_WRITE added
- AIRENCE_LED_ALL_EVENT added

1. Introduction

This document describes the protocol which is used to communicate with the control section and the USB Channel controls of the Airence USB mixing console. For the communication between the PC (host) and Airence (device) the USB generic HID (Human Interface Device) class protocol is used. HID devices are identified by their PID and VID. For the Airence the ***PID=0x2402***, and the ***VID=0x03EB***. On top of the HID protocol, a custom message based protocol provides the commands to send and receive data which will be described in more detail in this document.

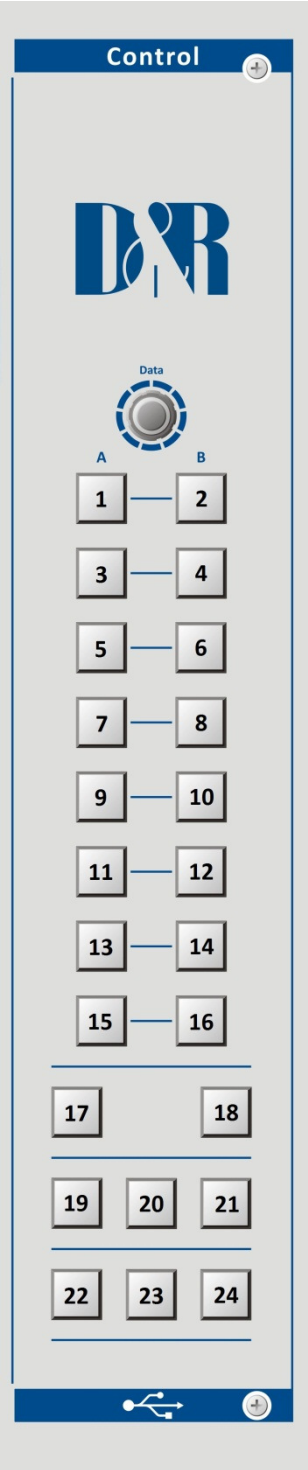


Figure 1. Airence control section

1.1 Control module

Switches

The control section of the Airence USB contains 24 switches which all have free assignable functionality. The switches are numbered from 1 to 24 as can be seen in figure 1. This numbering will be used in software to determine which switch is pressed/released.

Each switch contains a label which easily can be changed to customer needs. In such a way the control section can be customized with meaningful labels to control any software application on the PC.

LEDs

Behind each switch there is a multicolor LED to indicate any action or event. The LEDs can illuminate red, green, and yellow. With this feature one can indicate many different actions behind one switch which allows multifunctionality of the switches.

Furthermore the LEDs behind the switch are also numbered with the same numbering as the switches. For flexibility reasons the LEDs are not assigned to the switch they are placed behind. Instead they can be assigned/controlled freely. For example, switch 12 controls LED 2.

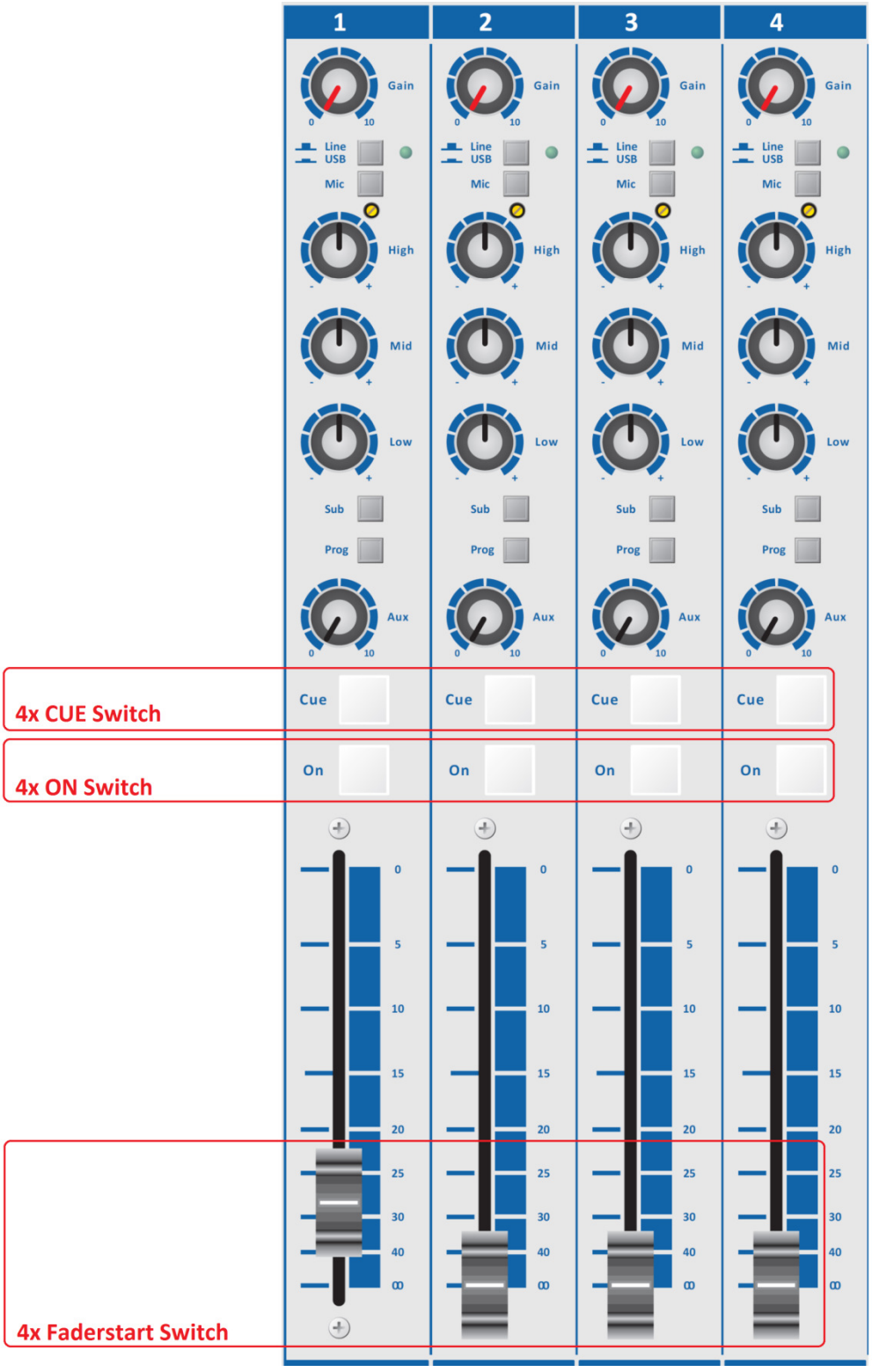
The LEDs can have the following states: ON, OFF, BLINKING.

Encoder

On top of the control section there is a rotary encoder located which can be used to adjust parameter values of the controlled application.

1.2 USB Channel Controls

In the Airence there are four USB channels available which can be used to send and recieve a total of 4 stereo pairs to and from the PC. Only on this four USB channels the faderstart, ON and CUE switch signals are transmitted to the PC using the Airence USB HID Control Protocol. These can be monitored by any playout software to trigger a specific function.



2. Message format

Transmitting data through the USB bus using the HID class protocol is done by reports. A report always contains an ID and the data inside the report. In order to succesfully communicate with the Airence control section ***only reportID 0*** may be used. More information about report descriptors can be found on www.usb.org. For now, remember that the Airence only will listen to reports with reportID = 0.

A control message has a fixed size of 8 bytes containing a *SIZE* byte, *COMMAND* byte, and a *PAYLOAD* section of maximal 6 bytes. The *SIZE* byte contains the number of bytes used (max. 8) in the message, including this *SIZE* byte.

byte:	0	1	2	3	4	5	6	7
	SIZE	**COMMAND**	**PAYLOAD**					
	n	cmd	p0	p1	p2	p3	p4	p5

2.1 Command byte

The command byte consists of a 2-bit *TYPE* and a 6-bit *ID* field.

bit:	7	6	5	4	3	2	1	0
	TYPE		**ID**					
	t1	t0	id5	id4	id3	id2	id1	id0

Bit	Symbol	Value	Description
5:0	ID	-	Command Identifier (ID)
7:6	TYPE	00	Write command, from host to device
		01	Request command, from host to device
		10	Response command, reply from device to host on a request
		11	Event, from device to host (i.e. button pressed)

Command ID

A commandID can have a value between 0x01 and 0x3F. The following commandIDs are currently available in the Airence:

Value	Description
0x01	AIRENCE_FIRMWARE_VERSION
0x02	AIRENCE_LED
0x03	AIRENCE_LED_BLINK
0x04	AIRENCE_LED_ALL
0x05	AIRENCE_SWITCH_CHANGE
0x06	AIRENCE_ENCODER_INCREMENT
0x07	AIRENCE_ENCODER_DECREMENT

3. Messages

3.1 Write

A message of the type *WRITE* is transmitted from the PC to Airence. The message is intended to perform an action on the Airence control section (i.e. illuminate a LED). If the action is correctly executed, the Airence will reply with a corresponding event.

AIRENCE_LED_WRITE

byte:	0	1	2	3	4	5	6	7
	SIZE	**COMMAND**	**PAYLOAD**					
	0x04	0x02	lednum	color	-	-	-	-

Byte	Symbol	Value	Description
0	SIZE	0x04	Message size
1	COMMAND	0x02	AIRENCE_LED_WRITE
2	lednum		LED number, 1(0x01) to 24(0x18), 0xFF = ALL LEDs
3	color	0x00	NONE (LED off)
		0x01	RED
		0x02	GREEN
		0x03	YELLOW

AIRENCE_LED_BLINK_WRITE

byte:	0	1	2	3	4	5	6	7
	SIZE	**COMMAND**	**PAYLOAD**					
	0x06	0x03	lednum	color_on	color_off	speed	-	-

Byte	Symbol	Value	Description
0	SIZE	0x06	Message size
1	COMMAND	0x03	AIRENCE_LED_BLINK_WRITE
2	lednum		LED number, 1(0x01) to 24(0x18), 0xFF = ALL LEDs
3	color_on	0x00	NONE (LED off)
		0x01	RED
		0x02	GREEN
		0x03	YELLOW
4	color_off	0x00	NONE (LED off)
		0x01	RED
		0x02	GREEN
		0x03	YELLOW
5	speed	0x00	SLOW
		0x01	NORMAL
		0x02	FAST

AIRENCE_LED_ALL_WRITE

byte:

0	1	2	3	4	5	6	7
SIZE	**COMMAND**	**PAYLOAD**					
0x08	0x04	LD4..1	LD8..5	LD12..9	LD16..13	LD20..17	LD24..21

Byte	Symbol	Value	Description
0	SIZE	0x08	Message size
1	COMMAND	0x04	AIRENCE_LED_ALL_WRITE
7:2	LD[m..n]	-	bit 1:0 LED[n] color 00 = NONE (LED off) | 01 = RED | 10 = GREEN bit 7:6 LED[m] color 11 = YELLOW

3.2 Request

A message of the type *REQUEST* is transmitted form the PC to Airence. The message is intended to read data from the Airence. After the Airence received the request it will respond with a corresponding *RESPONSE* message. Therefore, a *REQUEST* message always forms a pair with a *RESPONSE* message.

AIRENCE_FIRMWARE_VERSION_REQUEST

byte:

0	1	2	3	4	5	6	7
SIZE	**COMMAND**	**PAYLOAD**					
0x02	0x41	-	-	-	-	-	-

Byte	Symbol	Value	Description
0	SIZE	0x02	Message size
1	COMMAND	0x41	AIRENCE_FIRMWARE_VERSION_REQUEST

AIRENCE_SWITCH_CHANGE_REQUEST

byte:

0	1	2	3	4	5	6	7
SIZE	**COMMAND**	**PAYLOAD**					
0x02	0x45	-	-	-	-	-	-

Byte	Symbol	Value	Description
0	SIZE	0x02	Message size
1	COMMAND	0x45	AIRENCE_SWITCH_CHANGE_REQUEST

3.3 Response

A message of the type *RESPONSE* is transmitted from the Airence to the PC when prior a request message was received.

AIRENCE_FIRMWARE_VERSION_RESPONSE

byte:	0	1	2	3	4	5	6	7
	SIZE	**COMMAND**	**PAYLOAD**					
	0x04	0x81	major	minor	-	-	-	-

Byte	Symbol	Value	Description
0	SIZE	0x04	Message size
1	COMMAND	0x81	AIRENCE_FIRMWARE_VERSION_RESPONSE
2	major		Firmware major revision
3	minor		Firmware minor revision

AIRENCE_SWITCH_CHANGE_RESPONSE

byte:	0	1	2	3	4	5	6	7
	SIZE	**COMMAND**	**PAYLOAD**					
	0x08	0x85	sw_8_1	sw_16_9	sw_24_17	sw_enc_non	usb_2_1	usb_4_3

Byte	Symbol	Value	Description
0	SIZE	0x06	Message size
1	COMMAND	0xC5	AIRENCE_SWITCH_CHANGE_RESPONSE
4:2	sw_m_n	-	bit 0: switch n (1=pressed, 0=released) | bit 7: switch m
5	sw_enc_non	-	bit 0: encoder switch (1=pressed, 0=released) bit 1: non-stop switch (1=pressed, 0=released)
6	usb_2_1	-	bit 0: USB1 faderstart (1=ON, 0=OFF) bit 1: USB1 ON (1=pressed, 0=released) bit 2: USB1 CUE (1=pressed, 0=released) bit 3: USB2 faderstart (1=ON, 0=OFF) bit 4: USB2 ON (1=pressed, 0=released) bit 5: USB2 CUE (1=pressed, 0=released)
7	usb_4_3	-	bit 0: USB3 faderstart (1=ON, 0=OFF) bit 1: USB3 ON (1=pressed, 0=released) bit 2: USB3 CUE (1=pressed, 0=released) bit 3: USB4 faderstart (1=ON, 0=OFF) bit 4: USB4 ON (1=pressed, 0=released) bit 5: USB4 CUE (1=pressed, 0=released)

3.4 Events

A message of the type *EVENT* is transmitted from the Airence to the PC. An event will be generated when the state of a switch , encoder, or LED has changed. These events can be used to synchronise states of LEDs and switches between the Airence and the controlled application on the PC.

AIRENCE_LED_EVENT

byte:	0	1	2	3	4	5	6	7
	SIZE	**COMMAND**	**PAYLOAD**					
	0x04	0xC2	lednum	color	-	-	-	-

Byte	Symbol	Value	Description
0	SIZE	0x04	Message size
1	COMMAND	0xC2	AIRENCE_LED_EVENT
2	lednum		LED number, 1(0x01) to 24(0x18), 0xFF = ALL LEDs
3	color	0x00	NONE (LED off)
		0x01	RED
		0x02	GREEN
		0x03	YELLOW

AIRENCE_LED_BLINK_EVENT

byte:	0	1	2	3	4	5	6	7
	SIZE	**COMMAND**	**PAYLOAD**					
	0x06	0xC3	lednum	color_on	color_off	speed	-	-

Byte	Symbol	Value	Description
0	SIZE	0x06	Message size
1	COMMAND	0xC3	AIRENCE_LED_BLINK_EVENT
2	lednum		LED number, 1(0x01) to 24(0x18), 0xFF = ALL LEDs
3	color_on	0x00	NONE (LED off)
		0x01	RED
		0x02	GREEN
		0x03	YELLOW
4	color_off	0x00	NONE (LED off)
		0x01	RED
		0x02	GREEN
		0x03	YELLOW
5	speed	0x00	SLOW
		0x01	NORMAL
		0x02	FAST

AIRENCE_LED_ALL_EVENT

byte:

0	1	2	3	4	5	6	7
SIZE	**COMMAND**	**PAYLOAD**					
0x08	0xC4	LD4..1	LD8..5	LD12..9	LD16..13	LD20..17	LD24..21

Byte	Symbol	Value	Description
0	SIZE	0x08	Message size
1	COMMAND	0x04	AIRENCE_LED_ALL_EVENT
7:2	LD[m..n]	-	bit 1:0 LED[n] color 00 = NONE (LED off) | 01 = RED | 10 = GREEN bit 7:6 LED[m] color 11 = YELLOW

AIRENCE_SWITCH_CHANGE_EVENT

byte:

0	1	2	3	4	5	6	7
SIZE	**COMMAND**	**PAYLOAD**					
0x08	0xC5	sw_8_1	sw_16_9	sw_24_17	sw_enc_non	usb_2_1	usb_4_3

Byte	Symbol	Value	Description
0	SIZE	0x06	Message size
1	COMMAND	0xC5	AIRENCE_SWITCH_CHANGE_EVENT
4:2	sw_m_n	-	bit 0: switch n (1=pressed, 0=released) | bit 7: switch m
5	sw_enc_non	-	bit 0: encoder switch (1=pressed, 0=released) bit 1: non-stop switch (1=pressed, 0=released)
6	usb_2_1	-	bit 0: USB1 faderstart (1=ON, 0=OFF) bit 1: USB1 ON (1=pressed, 0=released) bit 2: USB1 CUE (1=pressed, 0=released) bit 3: USB2 faderstart (1=ON, 0=OFF) bit 4: USB2 ON (1=pressed, 0=released) bit 5: USB2 CUE (1=pressed, 0=released)
7	usb_4_3	-	bit 0: USB3 faderstart (1=ON, 0=OFF) bit 1: USB3 ON (1=pressed, 0=released) bit 2: USB3 CUE (1=pressed, 0=released) bit 3: USB4 faderstart (1=ON, 0=OFF) bit 4: USB4 ON (1=pressed, 0=released) bit 5: USB4 CUE (1=pressed, 0=released)

AIRENCE_ENCODER_INCREMENT_EVENT

byte:	0	1	2	3	4	5	6	7
	SIZE	**COMMAND**	**PAYLOAD**					
	0x03	0xC6	abs_value	-	-	-	-	-

Byte	Symbol	Value	Description
0	SIZE	0x03	Message size
1	COMMAND	0xC6	AIRENCE_ENCODER_INCREMENT_EVENT
2	abs_value		Absolute encoder value [0 - 255] (wrap around)

AIRENCE_ENCODER_DECREMENT_EVENT

byte:	0	1	2	3	4	5	6	7
	SIZE	**COMMAND**	**PAYLOAD**					
	0x03	0xC7	abs_value	-	-	-	-	-

Byte	Symbol	Value	Description
0	SIZE	0x03	Message size
1	COMMAND	0xC7	AIRENCE_ENCODER_DECREMENT_EVENT
2	abs_value		Absolute encoder value [0 - 255] (wrap around)

4. Test application

4.1 SimpleHIDWrite

For the integration of the Airence control protocol in your own application it can be very useful to have an application which provides simple write routines for HID devices in development stage. In this chapter the program SimpleHIDWrite will be used to show how to turn on a LED on the control section of the Airence. The used program can be downloaded from:

http://www.lvr.com/hidpage.htm/

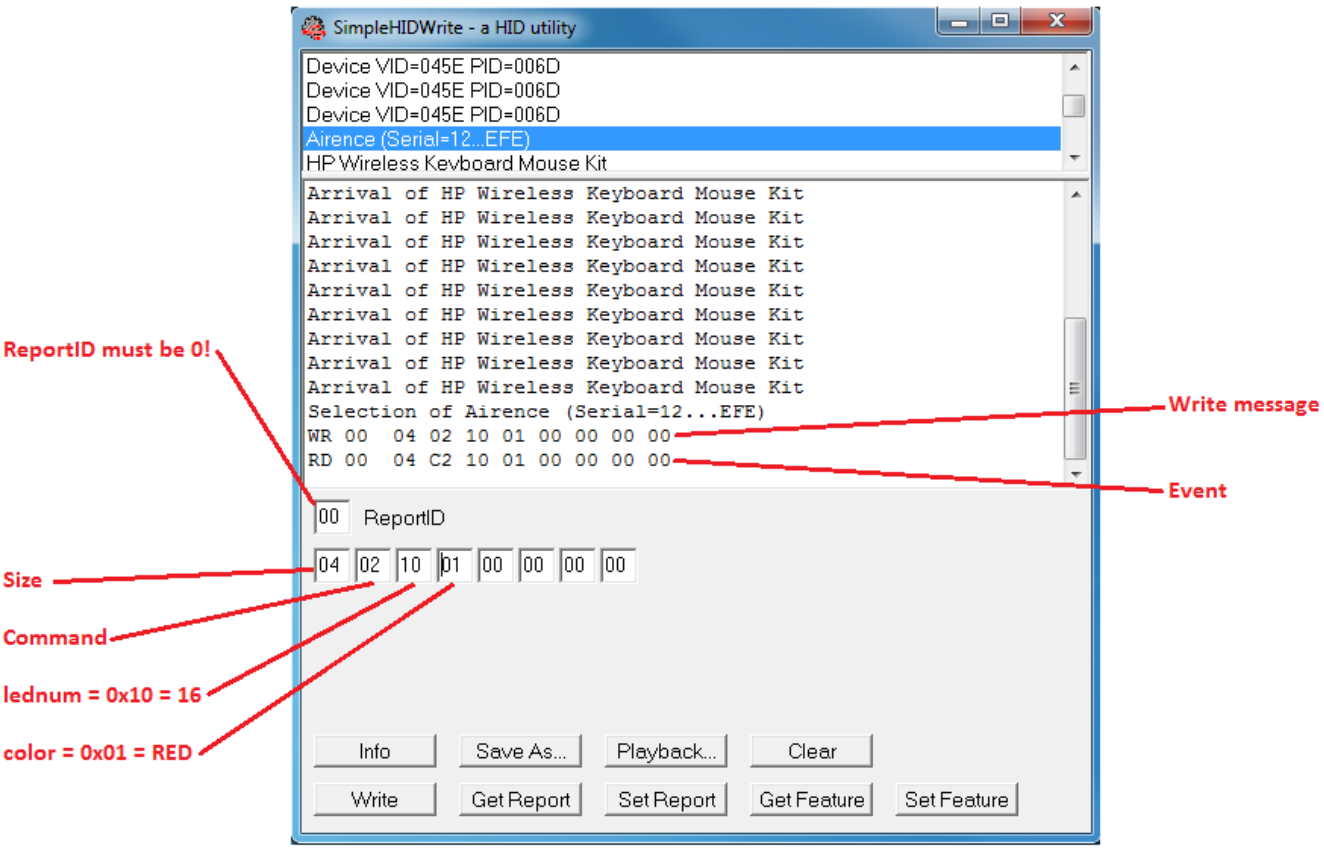


Figure 2. Generate a write message