

LEITCH LogoMotion II™ Control Bus Protocol

Revision 1.0, March 24, 2003

1. Introduction

This document describes the control bus protocol adaptation version compatible with the Leitch LogoMotion II. The purpose of this adaptation is to make the LogoMotion II automation interface look like the LogoMotion, thus allowing to control a new product without having to change anything to the automation vendor drivers. However, because of the very different nature of the LogoMotion II, some commands don't apply or react slightly differently. This protocol will make the link between the existing automation vendor drivers, as well as providing the necessary information to take advantage of the new features of the LogoMotion II. This document is NOT a replacement of the control bus protocol. It only describes the necessary changes in order to talk to the LogoMotion II, some information in this document is the same as the original Control Bus Protocol, but should be considered as a new branch that will follow it's own path throughout revisions and update.

2. Protocol Overview

LogoMotion command packets have the following format:

'~C <command> [<data><data>...] \ <cs>' Command

'~R <command> [<data><data>...] \ <cs>' Reply

Burst Commands are not supported.

The XY bus opening flag ('~') and closing flag ('\') are **reserved** characters, and must not be sent in the <command> or <data> area. The number of bytes between the opening and closing flags must not exceed 60 bytes. The checksum <cs> is required, and is the modulo 256 sum of all characters between the opening and closing flags, **IN**clusive, and is transmitted as two ascii hex characters, MSB first.

In order to transmit the reserved characters as binary data (as opposed to opening and closing flags), a data filter is applied to all data transmitted and received. The percent sign '%' is used as an escape flag and is deleted from the data stream. If the character following '%' is either 'O' or 'C' it is substituted as follows:

'%O' (25h, 4Fh) → '~' (7Eh) Opening Flag

'%C' (25h, 43h) → '\' (5Ch) Closing Flag

Any character immediately following '%' other than 'O' or 'C' will be used as-is, thus:

'%%' (25h, 25h) → '%' (25h) Percent Character

If a '~' or '\' character is received, it is **always** interpreted as the opening or closing flag.

Note: the <cs> is calculated on the **TRANSMITTED** characters (ie: with the two character sequence '%' and 'O', instead of the '~'). The standard convention for this protocol is to transmit all <word> and <dword> parameters with the MSB first.

Physical Protocol Interfaces

The LogoMotion II is commanded through RS422. The configuration parameters for the 422 port are: **38,400 baud, 8 bit, 1 stop bit, no parity.**

Table of Content

Playback Control Group	6
CP Play Item	6
CQ Query Current Item.....	7
CI Item Info Request	7
CX Change Item Playback Parameters	7
RP Playback Status	8
RI Item Info Reply	9
Keyer Control Group	11
CF Keyer Control	11
CJ Change Keyer Mode.....	11
CV Change Visual Keyer Attributes	12
CT Change Transition Attributes	12
CK Query Keyer Status.....	13
RJ Reply Keyer Mode	13
RK Keyer Status	14
Item Create / Delete Commands	15
CC Create Item	15
CD Delete Image	16
Animation Timeline Commands	17
CA Change Animation Timeline Parameters	17
RA Animation Timeline Status	17
CG / Time&Temp Group	19
CO Query CP/CW	19
CW Change CP/CW Attributes	19
RW Reply CP/CW Attributes.....	20
CH Set CW Data Specific	21
RH Reply CW Data Specific	21
Multi Layer Support	23

Protocol Conventions

<byte>: 8 bits, unsigned, transmitted in binary
<int> / <word>: 16 bit, signed / unsigned, transmitted in binary, MSB first
<long> / <dword>: 32 bit, signed / unsigned, transmitted in binary, MSB first
<item_number>: <word> 0..65535 where 0 is reserved for NotAnImage or default/unknown image.

The following number groups are predefined:

0001h thru 0FFFh: Logo Numbers 1..4095
1000h thru 1FFFh: Character Page (CP) Numbers, 0,1..4095
2000h thru 2FFFh: Character Window (CW) Numbers, 0,1..4095
3000h thru 3FFFh: reserved
4000h thru 7FFFh: NEXT logo/CP/CW, relative to # in 12 LSbits
8000h thru BFFFh: PREVIOUS logo/CP/CW, relative to # in 12 LSbits
C000h thru C0FFh: reserved
C100h thru FBFFh: reserved
FC00h thru FCFFh: reserved
FD00h thru FDFFh: reserved
FE00h thru FEFFh: reserved
FF00h thru FFFFh: reserved

<logo_number>: <word> The current DISPLAYABLE Logo number

Software Revisions: <word> binary numbers interpreted as <15byte_Major8><7byte_Minor0>, thus
Version 1.73 is encoded as <110><7310> or <1h><49h>

Card_ID: <byte> grouped as: [7group_code4 3unique#0] where [group_code] is: 0 = reserved 1 =

2 = Playback Card	3 = CG / Time&Temp
4 = Capture Card	5 = LGI
6 = Keyer Card	7 = All Keyers with same [unique#]
8 = Remote Panel	9 =
A = Genlock Card	B =
C = Processing Card	D =
E = Computer/Automation	F = ALL CARDS

and [unique#] is a unique identifier on any interconnected system

Examples are: 2602MP, 2602MG = Playback Cards

1302FK = Keyer Card
2602VC = Capture Card
1302VBR = Processing Card
1302TT = CG / Time&Temp Card

To access the four different layers of the product, the unique numbers will have to be mapped to the wished layer.

Size: <word_Size><word_Hpixels><word_Vlines> where:
<word_Size> = <byte_placement><byte_sizefactor>
<byte_placement> is bit encoded as: 7654TBLR where T=top, B=bottom, L=left, R=right
bit 7: 0=composite, 1= component
bit 6: 0=525, 1=625
bit 5: 0=8 bit data, 1=10 bit data
bit 4: 0=IQ axis, 1=UV, if bit 7=0
0=YcRcB, 1=RGB, if bit 7=1

<byte_sizeofactor> is size denominator (1/2/4/9...) or 0 to IGNORE <word_Size> (ie: custom size in Hpixels/Vlines)

Position: <int_Hpos><int_Vpos> where the absolute origin (0,0) is TOP LEFT Hpos increments by 1, and indicates groups of :

4 pixels for NTSC (4x14.3MHz=280ns), range 0..383 and

2 pixels for D1/PAL (2x13.5MHz=148ns), range 0..360 Vpos increments by 1, and indicates lines on a FIELD basis. where:

Line 0 is the first ACTIVE video line (NTSC line 21, PAL line 26)

The upper 4 bits of the upper byte of both <int_Hpos> and <int_Vpos> are reserved. There is no current definition, however each application MUST mask off the 4 bits to obtain the correct horizontal and vertical position.

Motion_CTL: <byte_motion_flags><word_next_image> where:

<byte_motion_flags>: 76543210:

bit 7: 0 = still 1 = motion

bits 654: 000 = run once and stop

001 = run once and cue next_image

010 = run once and loop to next_image

bit 3: 0 = no fade out 1 = do fade out

<word_next_image> = item_number of next item to cue/loop to, or 0 for cue/loop to self

TimeCode: <dword> arranged as: <byte_H, byte_M, byte_S, byte_F> represents

Hour, Minute, Second, Frame of standard timecode and must be normalized to:

H=0..23, M=0..59, S=0..59, F=0..29 (no support for drop-frame or 24fps)

TC of 0xFFFFFFFF is considered INVALID TC and should be displayed appropriately

DateCode: <dword> arranged as: <byte_Day, byte_Y, byte_M, byte_D> represents

Day: 0=Monday, ..., 6=Sunday

Year, Month, Date and must be normalized to:

Y=96++, M=1..12, D=1..31 (automatic leap year adjustment)

DC of 0xFFFFFFFF is considered INVALID DC and should be displayed appropriately

Character_Windows: <6byte> encoded as four 12bit entities for cw1..cw4 as follows:

[0] = cw1, bits 7..0

[1] = cw1, bits 11..8 in LSn, cw2, bits 3..0 in MSn

[2] = cw2, bits 11..4

[3] = cw3, bits 7..0

[4] = cw3 bits 11..8 in LSn, cw4, bits 3..0 in MSn

[5] = cw4, bits 11..4

Character Set ID: <byte> character set to use or 0 for default character set

FFtype <byte> Type of Fixed Function Character Window, as follows:

0x00 reserved

0x01 timer

0x02 clock

0x03 temperature

0x04 date

Playback Control Group

CP Play Item
CQ Query Current Image
CI Image Info Request
CX Change Image Playback Parameters
RP Playback Status
RI Item Info Reply

CP Play Item

Format: CP <Card_ID> <keyer_ID> <item_number> <keyer_cmd> <flags>

Packet Size: 8 Bytes

Data Structure:

playback_ID: <byte> Card ID of the module to control.

keyer_ID: <byte> Card ID of the keyer to control.

The keyer ID has no effect.

item_number: <word> The item number to play.

keyer_cmd: <76543210>

bit 7: 1 = unconditional cut off

bit 6: 1 = unconditional fade off

bit 5: 1 = unconditional cut on

bit 4: 1 = unconditional fade on

bit 3: 1 = toggle cut

bit 2: 1 = toggle fade

flags: <76543210>

If <item_number> defines a Logo, then

bit 7: 1 = reload image if already current, 0 = don't reload if current

bit 6: 1 = run immediately, 0 = hold on 1st frame

bit 5: 1 = respect <keyer_cmd> above 0 = ignore <keyer_cmd>

bit 4: 1 = respect stored motion flags 0 = ignore motion flags

This bit has no effect, motion flags are always respected.

Description:

Plays an item and (optionally) triggers a keyer operation.

The <keyer_cmd> is described in the 'CF' Keyer Control command description.

Reply:

RP Playback Status,

RK Keyer Status

CQ Query Current Item

Format: CQ <Card_ID>

Packet Size: 3 Bytes

Data Structure:

Card_ID: <byte> Card ID of the module to query.

Description:

Returns the playback status of the addressed card.

Reply:

RP Playback Status

CI Item Info Request

Format: CI <Card_ID> <item_number_start> <item_number_end>

Packet Size: 7 Bytes

Data Structure:

Card_ID: <byte> The Card_ID of the module being addressed.

item_number_start: <word> The beginning item number for this request.

item_number_end: <word> The ending item number for this request.

Description:

Returns information on all items stored within the item_number range specified (inclusive). If <start> is greater than <end> the command is ignored.

Reply:

RI Item Info if the item exists, otherwise nothing, for each item within the range.

CX Change Item Playback Parameters

Format: CX <Card_ID> <item_number> <Hpos> <Vpos> <flags>

Packet Size: 10 Bytes

Data Structure:

Card_ID: <byte> Card ID of the module being addressed.

item_number: <word> The item number to change

Hpos: <int> Horizontal playback position, as defined in Conventions

Vpos: <int> Vertical playback position, as defined in Conventions

flags: <76543210>

bit 7: 1 = save this position 0 = temporary change

This bit is ignored, all changes are permanent.

bit 6: 1 = absolute position, 0 = relative to current pos

Description:

Changes the playback position of the currently playing item.

Reply:

RP Playback Status

RP Playback Status

Format: RP <Card_ID> <item_number> <contents>

If <item_number> defines a Logo, then <contents> become:

<logo_number> <num_frames> <Hpos> <Vpos> <Size> <Hpixels> <Vlines> <image_info>

If <item_number> defines a CharacterPage, then <contents> become:

<Character_Windows> <Hpos> <Vpos> <CharSetID> <spare>

If <item_number> defines a CharacterWindow, then <contents> become:

<FFtype> <BGno> <CWattributes><Hpos> <Vpos> <CharSetID> <spare>

Packet Size: 20 Bytes

Data Structure:

Common:

Card_ID: <byte> Card ID of the module returning status.

item_number: <word> The current image number, CharacterPage, or CharacterWindow.

Logos:

logo_number: <word> The current DISPLAYABLE Logo number.

num_frames: <word> Number of frames duration, for motion video

Hpos: <int> Horizontal playback position, as defined in Conventions

Vpos: <int> Vertical playback position, as defined in Conventions

Size: <word> Image size, as defined in Conventions

Hpixels: <word>

Vlines: <word>

image_info: <76543210>

bit 7: 1 = motion video, 0 = still

bit 6: 1 = 625 line, 0 = 525 line

bit 5: 1 = T&T valid, 0 = No T&T

bit 1..0: 0 = cued

1 = playing once

2 = playing looping

3 = stopped at end

CharacterPage:

Character_Windows: <6byte> The character window numbers making up the character page

Hpos: <int> Horizontal playback position, as defined in Conventions

Vpos: <int> Vertical playback position, as defined in Conventions

CharSetID: <byte> Character Set ID, as defined in Conventions

spare: <4byte> reserved

CharacterWindow:

FFtype <byte> Type of fixed function CW, as defined in Conventions

BGno <byte> Background number or 0 for no background

CWattributes: <word> CW attributes, as defined in Conventions

Hpos: <int> Horizontal playback position, as defined in Conventions

Vpos: <int> Vertical playback position, as defined in Conventions

CharSetID: <byte> Character Set ID, as defined in Conventions

spare: <6byte> reserved

Description:

Returns the settings and operational state of a module.

Reply:

None.

RI Item Info Reply

Format: RI <Card_ID> <item_number> <contents>

If <item_number> defines a Logo, then <contents> become:

<logo_number> <image_info> <num_frames> <Hpos> <Vpos> <Size> <Hpixels> <Vlines >

If <item_number> defines a CharacterPage, then <contents> become:

<character_windows> <Hpos> <Vpos> <CharSetID> <spare>

If <item_number> defines a CharacterWindow, then <contents> become:

<FFtype> <BGno> <CWattributes> <Hpos> <Vpos> <CharSetID> <spare>

Packet Size: 20 Bytes

Data Structure:

Common:

Card_ID: <byte> Card ID of the module returning info.

item_number: <word> The item number.

Logos:

logo_number: <word> The current DISPLAYABLE Logo number.

image_info: <76543210>

bit 7: 1 = motion video, 0 = still

bit 6: 1 = 625 line, 0 = 525 line

bit 5: 1 = fill + key, 0 = video fill only, no linear key

bit 4: 1 = logo deleted, 0 = logo exists or was created OK

bit 3: 1 = delete/create FAILED

bit 2: 1 = WRITE_PROTECT failure

bit 1: 1 = OUT_OF_MEMORY failure

bit 0: 1 = T&T valid, 0 = T&T not valid

num_frames: <word> Number of frames duration, for motion video

Hpos: <int> Horizontal playback position, as defined in Conventions

Vpos: <int> Vertical playback position, as defined in Conventions

Size: <word> Image size, as defined in Conventions

Hpixels: <word>

Vlines: <word>

Note: num_frames, Hpos, Vpos, Size, Hpixels, Vlines are IGNORED if image_info indicates DELETED

CharacterPage:

character_windows: <6byte> The character window numbers making up the character page

Hpos: <int> Horizontal playback position, as defined in Conventions

Vpos: <int> Vertical playback position, as defined in Conventions

CharSetID: <byte> Character Set ID, as defined in Conventions

spare: <4byte> reserved

CharacterWindow:

FFtype <byte> Type of fixed function CW, as defined in Conventions

BGno <byte> Background number or 0 for no background

CWattributes: <word> CWattribute, as defined in Conventions

Hpos: <int> Horizontal playback position, as defined in Conventions

Vpos: <int> Vertical playback position, as defined in Conventions
CharSetID: <byte> Character Set ID, as defined in Conventions

image_info: <76543210>
bit 7: 1 = motion video, 0 = still
bit 6: 1 = 625 line, 0 = 525 line
bit 5: 1 = fill + key 0 = video fill only, no linear key
bit 4: 1 = logo deleted 0 = logo exists or was created OK
bit 3: 1 = delete/create FAILED
bit 2: 1 = WRITE_PROTECT failure
bit 1: 1 = OUT_OF_MEMORY failure
bit 0:
spare: <5byte> reserved

Description:

Returns all attributes of an item, whether playing or not.

Reply:

None.

Keyer Control Group

CF Keyer Control
CJ Change Keyer Mode
CV Change Visual Keyer Attributes
CT Change Transition Attributes
CK Query Keyer Status
RJ Reply Keyer Mode
RK Keyer Status

CF Keyer Control

Format: CF <Card_ID> <keyer_cmd>

Packet Size: 4 Bytes

Data Structure:

Card_ID: <byte> Card ID of the keyer to control.

keyer_cmd: <76543210>

bit 7: 1 = unconditional cut off
bit 6: 1 = unconditional fade off
bit 5: 1 = unconditional cut on
bit 4: 1 = unconditional fade on
bit 3: 1 = toggle cut
bit 2: 1 = toggle fade

Description:

The <cmd> byte will be tested from MSB to LSB, and the first TRUE bit encountered will be acted upon. Should multiple bits be set, the lower order ones will be ignored. If no bits are set, no action is taken and this is NOT considered an error.

Reply:

RK Keyer Status

CJ Change Keyer Mode

Format: CJ <Card_ID> <mode>

Packet Size: 2 Bytes

Data Structure:

Card_ID: <byte> Card ID of the keyer to control.

keyer_cmd: <76543210> **This command has no effects**

bit 7: 1 = save, 0 = temporary
bit 0: 1 = KEY/FILL, 0 = PGM/PRV

Description:

Output is PGM/PRV when using the internal keyer. If an external keyer is used the output is the FILL and KEY.

Reply:

RJ Change Keyer Mode

CV Change Visual Keyer Attributes

Format: CV <keyer_ID> <item_number> <offset> <gain> <transparency> <flags>

Packet Size: 9 Bytes

Data Structure:

keyer_ID: <byte> Card ID of the keyer to control.

item_number: <word> The item number to change, or 0 for current item

offset: <byte> The new offset setting [0..255].

gain: <byte> The new gain setting [0..255].

transparency: <byte> The new transparency setting [0..255].

flags: <76543210> **These flags have no effects**

bit 7: 1 = save settings permanently with image, 0 = temporary

bit 6: 1 = self key, 0 = linear key

bit 0: 1 = Additive key (3701 only), 0 = Mix Mode key

Description:

Changes the keyer appearance settings of the addressed linear keyer card.

Reply:

RK Keyer Status.

CT Change Transition Attributes

Format: CT <keyer_ID> <item_number> <in_rate> <out_rate> <ramp_amp> <flags>

Packet Size: 9 Bytes

Data Structure:

keyer_ID: <byte> Card ID of the keyer to control.

item_number: <word> The item number to change, or 0 for current item

in_rate: <byte> The new fade-in rate setting in frames [0..255].

out_rate: <byte> The new fade-out rate setting in frames [0..255].

ramp_amp: <byte> The new maximum fade-in ramp amplitude setting [0..255].

flags: <76543210>

bit 7: 1 = save settings permanently with image, 0 = temporary

This bit has no effects, parameters are always saved

Description:

Changes the control characteristics of the addressed linear keyer card.

Reply:

RK Keyer Status

CK Query Keyer Status

Format: CK <keyer_ID> <item_number>

Packet Size: 5 Bytes

Data Structure:

keyer_ID: <byte> Card ID of the keyer to query, or 0 for all keyers to respond.

item_number: <word> The item number to query keyer settings for, or 0 for current settings

Description:

Requests one or all keyers to return its keyer settings.

Reply:

RK Keyer Status.

RJ Reply Keyer Mode

RJ Reply Keyer Mode

Format: RJ <Cmd_ID> <mode>

Packet Size: 2 Bytes

Data Structure:

Card_ID: <byte> Card ID of the keyer to control.

mode: <76543210>

bit 0: 1 = external keyer, 0 = internal

Description:

Returns the what keyer is used (internal or external).

Reply:

None.

RK Keyer Status

Format: RK <keyer_ID> <item_number> <offset> <gain> <transparency>
<in_rate> <out_rate> <ramp_amp> <state>

Packet Size: 12 Bytes

Data Structure:

keyer_ID: <byte> Card ID of the keyer returning status.
item_number: <word> The item number which status is for, or 0 for current settings
offset: <byte> The offset setting [0..255].
gain: <byte> The gain setting [0..255].
transparency: <byte> The transparency setting [0..255].
in_rate: <byte> The fade-in rate setting in frames [0..255].
out_rate: <byte> The fade-out rate setting in frames [0..255].
ramp_amp: <byte> The maximum fade-in ramp amplitude setting [0..255].

state: <76543210>

bit 7: 1 = keyer enabled 0 = keyer disabled
bit 6: 1 = tally on 0 = tally off
bit 5: 1 = bypass relay on 0 = bypass relay off
bit 4: 1 = in transition 0 = transition complete
bit 3: 1 = self key 0 = linear key
bit 0: 1 = Additive key (3701 only) 0 = Mix Mode key

Description:

Returns the settings and operational state of a linear keyer card.

Reply:

None.

Item Create / Delete Commands

CC Create Item

CD Delete Image

CC Create Item

Format: CC <Card_ID> <item_number> <contents>

If <item_number> defines a Logo, then <contents> become:
<logo_number> <Size> <Hpixels> <Vlines> <num_frames>

If <item_number> defines a CharacterPage, then <contents> become:
<Character_Windows> <Hpos> <Vpos>

If <item_number> defines a CharacterWindow, then <contents> become:
<FFtype> <BGno> <CWattributes> <CharSetID> <Hpos> <Vpos> <spare>

If <item_number> defines a CharacterSet, then <contents> become:
<TotalCSBytes> <spare> <spare> <spare> <spare> <spare> <spare>

Packet Size: 15 Bytes

Data Structure:

Common:

Card_ID: <byte> Card ID of the module to create on.

item_number: <word> The item number to create.

Logos:

logo_number: <word> The current DISPLAYABLE Logo number, or logo to LINK to.

Size: <word> Image size, as defined in Conventions

Hpixels: <word>

Vlines: <word>

num_frames: <word> Number of frames duration

CharacterPage:

Character_Windows: <6byte> The character window numbers making up the character page

Hpos: <word> Absolute horizontal playback position, as defined in Conventions

Vpos: <word> Absolute vertical playback position, as defined in Conventions

CharacterWindow:

FFtype <byte> Type of fixed function CW, as defined in Conventions

BGno <byte> Background number or 0 for no background

CWattributes: <word> CWattributes, as defined in Conventions

CharSetID: <byte> Character Set ID, as defined in Conventions

Hpos: <word> Absolute horizontal playback position, as defined in Conventions

Vpos: <word> Absolute vertical playback position, as defined in Conventions

spare: <byte> reserved

CharacterSet:

TotalCSBytes: <lword> Total number of bytes in this character set

spare: <6byte> reserved

Description:

Creates the indicated item. If the item already exists, it will **NOT** be deleted, and this command will fail. Existing items must be explicitly deleted with the Delete Item command.

Reply:

RC Capture Status (if storage compaction is in progress)

RI Image Info Reply

CD Delete Image

Format: CD <playback_ID> <item_number>

Packet Size: 5 Bytes

Data Structure:

playback_ID: <byte> Card ID of the playback card to control.

item_number: <word> The item number to delete.

Description:

Deletes the indicated image, if it exists, and returns the freed memory to the available pool.

Reply:

RI Image Info Reply

Animation Timeline Commands

CA Change Animation Timeline Parameters
RA Animation Timeline Status

CA Change Animation Timeline Parameters

Format: CA <playback_ID> <item_number> <delay> <pause> <next_image> <motion_flags>
Packet Size: 16 Bytes

Data Structure:

playback_ID: <byte> Card ID of the playback card to control.
item_number: <word> The item number to change
delay: <dword> Timecode of animation delay, at start of animation timeline
pause: <dword> Timecode of animation pause, at end of animation timeline
next_image <word> the image number of the next logo in the timeline
motion_flags: <76543210>
bit 7: 1 = motion logo 0 = still logo
bit 6,5: motion sequence command:
00 = run, then hold at end of logo (hold at last frame)
01 = run, then cue next_image (hold at first frame)
10 = run, then play next_image (no pause)
11 = reserved
bit 4: 1 = CUT ON before running
bit 3: 1 = FADE ON before running
bit 2: 1 = CUT OFF after running
bit 1: 1 = FADE OFF after running:
bit 0: 1 = save parameters permanently 0 = temporary

Description:

Changes the motion timeline of the selected logo.

Reply:

RA Animation Timeline Status

RA Animation Timeline Status

Format: RA <playback_ID> <item_number> <delay> <pause> <next_image> <motion_flags>
Packet Size: 16 Bytes

Data Structure:

playback_ID: <byte> Card ID of the playback card giving status
item_number: <word> The item number described
delay: <dword> Timecode of animation delay, at start of animation timeline
pause: <dword> Timecode of animation pause, at end of animation timeline
next_image <word> the image number of the next logo in the timeline
motion_flags: <76543210>
bit 7: 1 = motion logo 0 = still logo
bit 6,5: motion sequence command:
00 = run, then hold at end of logo (hold at last frame)
01 = run, then cue next_image (hold at first frame)
10 = run, then play next_image (no pause)
11 = reserved
bit 4: 1 = CUT ON before running

bit 3: 1 = FADE ON before running
bit 2: 1 = CUT OFF after running
bit 1: 1 = FADE OFF after running:
bit 0: 1 = item has Time&Temp settings 0 = item does not have T&T settings

Description:

Indicates the motion timeline of the indicated logo.

Reply:

None

CG / Time&Temp Group

CO Query CP/CW
CW Change CP/CW Attributes
RW Reply CP/CW Attributes
CH Set CW Data Specific
RH Reply CW Data Specific

CO Query CP/CW

Format: CO <Card_ID> <CPnumber> <CWnumber> <FFtype> <flags>

Packet Size: 9 Bytes

Data Structure:

Card_ID: <byte> Card ID of the CG/T&T module to query
CPnumber: <word> CP item number, or 0 for global CWnumber
CWnumber: <word> CW item number within CP (if CPnumber!=0), else global Cwnumber
FFtype: <byte> Type of fixed function CW, as defined in Conventions

flags: <76543210>
bit 7..0: reserved

Description:

This command queries info on a particular CP or CW. Both Attributes and Data Specific info is returned.

Reply:

RW Reply CP/CW Attributes
RH Reply CP/CW Data Specific

CW Change CP/CW Attributes

Format: CW <Card_ID> <CPnumber> <CWnumber> <FFtype> <BGno> <Cwattributes>
<CharSetID> <Hpos> <Vpos> <flags>

Packet Size: 16 Bytes

Data Structure:

Card_ID: <byte> Card ID of the CG/T&T module to control.
CPnumber <word> CP item number, or 0 for global CWnumber change
CWnumber <word> CW item number within CP (if CPnumber!=0), else global CWnumber
FFtype <byte> Type of fixed function CW, as defined in Conventions
BGno <byte> Background number or 0 for no background
CWattributes <word> Attributes to use on specified CW
CharSetID <byte> Character set to use on CW
Hpos <int> Relative horizontal position
Vpos <int> Relative vertical position

flags: <76543210> **These flags have no effect**

bit 7: 1 = save, 0 = TEMP
bit 5: 1 = apply CHARSET, 0 = ignore CHARSET
bit 4: 1 = apply BGNO, 0 = ignore BGNO
bit 3: 1 = absolute position, 0 = relative to current position
bit 2: 1 = apply vpos field, 0 = ignore vpos field
bit 1: 1 = apply hpos field, 0 = ignore hpos field

bit 0: 1 = apply attributes field, 0 = ignore attributes field

Description:

Reply:

RW Reply CP/CW Attributes

RW Reply CP/CW Attributes

Format: RW <Card_ID> <CPnumber> <CWnumber> <FFtype> <BGno> <CWattributes>
<CharSetID> <Hpos> <Vpos> <flags>

Packet Size: 17 Bytes

Data Structure:

Card_ID: <byte> Card ID of the CG/T&T module replying

CPnumber <word> CP item number, or 0 for global CWnumber

CWnumber <word> CW item number within CP (if CPnumber!=0), else global Cwnumber

FFtype <byte> Type of fixed function CW, as defined in Conventions

BGno <byte> Background number or 0 for no background

CWattributes <word> Attributes to use on specified CW

CharSetID <byte> Character set to use on CW

Hpos <int>

Vpos <int>

flags: <76543210> **these flags always return status OK (0x0)**

bit 7: 1 = error in previous CW settings 0 = all is OK

bit 6: 1 = invalid CP number 0 = CP number OK

bit 5: 1 = invalid CW number 0 = CW number OK

bit 4..0: reserved

Description:

This command returns info on a particular CP or CW.

Reply:

none

CH Set CW Data Specific

Format: CH <Card_ID> <CPnumber> <CWnumber> <FFtype> <flags> <FFstatus> <count>
<data1> ... <dataN>

Packet Size: 11 + COUNT Bytes

Data Structure:

Card_ID: <byte> Card ID of the CG/T&T module to control

CPnumber <word> CP item number, or 0 for global CWnumber

CWnumber <word> CW item number within CP (if CPnumber!=0), else global Cwnumber

FFtype <byte> Type of fixed function CW this specific data belongs to

flags: <76543210>

bit 7..0: reserved

FFstatus <byte> reserved

count: <byte> number of bytes to follow (0..255)

data1..N <byte> data specific to the CW

Description:

This command sets specific data for a CharacterWindow. The data is FFtype specific, and is defined below

Reply:

RH Reply CW Data Specific

RH Reply CW Data Specific

Format: RH <Card_ID> <CPnumber> <CWnumber> <FFtype> <flags> <FFstatus> <count>
<data1> ...
<dataN>

Packet Size: 11 + COUNT Bytes

Data Structure:

Card_ID: <byte> Card ID of the CG/T&T module replying

CPnumber <word> CP item number, or 0 for global CWnumber

CWnumber <word> CW item number within CP (if CPnumber!=0), else global Cwnumber

FFtype <byte> Type of fixed function CW this specific data belongs to

Flags: <76543210>

bit 7: 1 = error in previous CH settings 0 = all is OK

bit 6: 1 = invalid CP number 0 = CP number OK

bit 5: 1 = invalid CW number 0 = CW number OK

bit 4..0: reserved

FFstatus: <byte> status pertaining to this CW

count: <byte> number of bytes to follow (0..255)

data1..N <byte> data specific to the CW

Description:

This reply returns info on specific data for a CharacterWindow. The data is FFtype specific, and is defined below.

Reply:

none

Data Specific Entities

FFtype Count Data1..N

FFclock 4 <TimeCode> <lword> time of day, as per Conventions

FFdate 4 <DateCode> <lword> date, as per Conventions

FFtemp 3 <degrees> <int> signed temperature * 10 (ie: +25.3 degrees = 253)
<byte> Scale, 0 = Celcius, 1 = Farenheit

FFtimer 4 <TimeCode> <lword> timer preset, as per Conventions

FFstatus <76543210>

FFclock bit 7: 1 = RTC clock failure 0 = RTC operating OK
bit 6: 1 = TOD timecode unavailable 0 = TOD timecode OK
bit 5..0: reserved

FFdate bit 7: 1 = RTC clock failure 0 = RTC operating OK
bit 6: 1 = Date timecode unavailable 0 = Date timecode OK
bit 5..0: reserved

FFtemp bit 7: 1 = temp probe failure 0 = temp probe operating OK
bit 6: 1 = manual TEMP 0 = probe TEMP
bit 5..0: reserved

FFtimer bit 7..0: reserved

Multi Layer Support

In order to be able to use multiple layers, the unique numbers of the card have to be mapped to the wished layer. In order to do that, the file "LeitchAutomationSetup.txt" in the "Eng" folder of the MGI 3901 Flash Card has to be configured. Here is a default view of that file:

UniqueNbLayer1:	1
UniqueNbLayer2:	2
UniqueNbLayer3:	3
UniqueNbLayer4:	4
TNTUniqueNb1TimeLayer:	2
TNTUniqueNb1TempLayer:	3
TNTUniqueNb2TimeLayer:	3
TNTUniqueNb2TempLayer:	4
TNTUniqueNb3TimeLayer:	1
TNTUniqueNb3TempLayer:	2
TNTUniqueNb4TimeLayer:	2
TNTUniqueNb4TempLayer:	3
TempLogo:	1
TimeLogo:	2
LegacyEnabled:	1
LeitchProtocol:	1

Parameters Description:

UniqueNbLayerX: x

In this case the layer 1 will be accessed by using unique number 1, layer 2 with unique number 2 and so on. Using the same principle, any layer can be mapped to the wished unique number.

TNTUniqueNbxTimeLayer: x

TNTUniqueNbxTempLayer: x

Defines what layer will be used for time and temperature

TempLogo: x

TimeLogo: x

Defines wish logo will be used when using the T&T commands

Legacy Enabled: x

1 = indicate a use with previous automation software or panel without alteration.

0 = enables some new behaviour that wouldn't be supported by previous automation vendors or panels

LeitchProtocol: x

0 = disabled

1 = enabled